

IBM Smart Analytics System

Understand IBM Smart Analytics System configuration

Learn how to administer IBM Smart Analytics System

Integrate with existing IT systems



Whei-Jen Chen
Rafael Aiello
Silvio Luiz Correia Ferrari
Zohar Nissare-Houssen



International Technical Support Organization

IBM Smart Analytics System

February 2011

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (February 2011)

This edition applies to IBM Smart Analytics System 5600, 7600, and 7700.

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team who wrote this book	ix
Acknowledgements	x
Now you can become a published author, too!	xi
Comments welcome	xii
Stay connected to IBM Redbooks	xii
Chapter 1. IBM Smart Analytics System	1
1.1 Overview	2
1.1.1 Architecture	4
1.2 IBM Smart Analytics System portfolio	7
1.2.1 IBM Smart Analytics System 1050 and 2050	8
1.2.2 IBM Smart Analytics System 5600	8
1.2.3 IBM Smart Analytics System 7600 and 7700	10
1.2.4 IBM Smart Analytics System 9600	13
1.2.5 IBM Smart Analytics System family summary	13
1.3 IBM training	14
1.3.1 IBM Professional Certification	15
1.3.2 Information Management Software Services	15
1.3.3 IBM Software Accelerated Value Program	17
1.3.4 Protect your software investment: Ensure that you renew your Software Subscription and Support	17
Chapter 2. Installation and configuration	19
2.1 Planning	20
2.1.1 Smart Analytics System customer worksheet	20
2.1.2 Floor diagram and specification review	25
2.2 Installation of IBM Smart Analytics System	26
2.2.1 Installation at the IBM Customer Solution Center	26
2.2.2 Installation at the customer site	27
2.3 Documentation and support for the IBM Smart Analytics System	28
Chapter 3. High availability	31
3.1 High availability on IBM Smart Analytics System	32
3.2 IBM Tivoli System Automation for Multiplatforms on IBM Smart Analytics System	33

3.3	High availability overview for the core warehouse servers	34
3.4	Managing high availability resources for the core warehouse.	39
3.4.1	Monitoring the high availability resources for the core warehouse . .	41
3.4.2	Starting and stopping resources with the high availability management toolkit	43
3.4.3	Starting and stopping resources with Tivoli SA MP commands	45
3.4.4	Manual node failover for maintenance	46
3.4.5	Manual node failback	48
3.5	High availability for the warehouse application module.	51
3.5.1	Starting and stopping high availability resources for warehouse application servers	56
3.5.2	Manual failover warehouse application node	56
3.5.3	Manual failback of the warehouse application node	57
3.6	High availability for the business intelligence module	59
3.6.1	Starting and stopping high availability resources for BI module	65
3.6.2	Manual failover BI type 1 node to BI type 2 node	69
3.6.3	Manual failover BI type 2 node to BI type 1 node	69
3.6.4	Manual failback BI type 1 node	70
3.6.5	Manual failback BI type 2 node	71
	Chapter 4. Maintenance	75
4.1	Managing DB2 message logs	76
4.1.1	The db2dback shell script	76
4.1.2	db2support -archive	76
4.1.3	The db2diag utility	77
4.2	Changing the date and time	80
4.3	IBM Smart Analytics System upgrades.	82
4.3.1	IBM Smart Analytics System software and firmware stacks	82
4.3.2	The Dynamic System Analysis tool.	83
4.3.3	IBM Smart Analytics System Control Console	85
4.4	IBM HealthCheck Service	86
4.5	IBM Smart Analytics System installation report.	87
4.6	IBM Smart Analytics System backup and recovery.	88
4.6.1	Operating system backup and recovery	89
4.6.2	Database backup and recovery	97
	Chapter 5. Monitoring tools	105
5.1	Cluster and operating system monitoring	106
5.1.1	AIX and Linux	106
5.1.2	IBM Systems Director	106
5.2	DB2 monitoring	110
5.2.1	DB2 monitoring utilities	111
5.2.2	DB2 Performance Expert for Linux, UNIX, and Windows	117

5.3 Storage monitoring	122
5.3.1 IBM Remote Support Manager	122
5.3.2 Internal disks	124
5.3.3 SAN switches	124
5.4 Network monitoring	126
Chapter 6. Performance troubleshooting	127
6.1 Global versus local server performance troubleshooting	128
6.1.1 Running performance troubleshooting commands	129
6.1.2 Formatting the command output	135
6.2 Performance troubleshooting at the operating system level	137
6.2.1 CPU, run queue, and load average monitoring	137
6.2.2 Disk I/O and block queue	142
6.2.3 Memory usage	148
6.2.4 Network	150
6.3 DB2 Performance troubleshooting	152
6.3.1 CPU consumption	155
6.3.2 I/O usage	169
6.3.3 DB2 memory usage	186
6.3.4 DB2 network usage	193
6.4 Common scenario: Data skew	195
6.4.1 Operating system monitoring	196
6.4.2 DB2 monitoring	196
Chapter 7. Advanced configuration and tuning	203
7.1 Configuration parameters	204
7.1.1 Operating system and kernel parameters	204
7.1.2 DB2 configuration	217
7.1.3 DB2 buffer pool and table spaces	236
7.2 DB2 workload manager	243
7.2.1 Working with DB2 workload manager	244
7.2.2 Configuring a DB2 workload manager for an IBM Smart Analytics System	247
7.2.3 DB2 workload manager resources	273
7.3 Capacity planning	274
7.3.1 Identifying resource requirements	275
7.3.2 Increasing capacity of existing systems	276
7.3.3 Adding additional modules	278
Appendix A. Smart Analytics global performance monitoring scripts ..	281
Appendix B. Scripts for DB2 workload manager configuration	299
B.1 Creating MARTS tables	300
B.2 Untuned DB2 workload manager configuration	303

B.3 Tuned DB2 workload manager configuration	315
Related publications	319
IBM Redbooks publications	319
Online resources	319
Help from IBM	320
Index	321

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	InfoSphere™	Solid®
Alphablox®	Intelligent Miner®	System p®
Balanced Warehouse®	Netcool®	System x®
Cognos®	NetView®	System z®
DataStage®	Optim™	Tivoli Enterprise Console®
DB2®	POWER6®	Tivoli®
developerWorks®	QualityStage™	WebSphere®
DS4000®	Redbooks®	z/OS®
GPFS™	Redpaper™	
IBM®	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel Xeon, Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

The IBM® Smart Analytics System is a fully-integrated and scalable data warehouse solution that combines software, server, and storage resources to offer optimal business intelligence and information management performance for enterprises.

This IBM Redbooks® publication introduces the architecture and components of the IBM Smart Analytics System family. We describe the installation and configuration of the IBM Smart Analytics System and show how to manage the systems effectively to deliver an enterprise class service.

This book explains the importance of integrating the IBM Smart Analytics System with the existing IT environment, as well as how to leverage investments in security, monitoring, and backup infrastructure. We discuss the monitoring tools for both operating systems and DB2®. Advance configuration, performance troubleshooting, and tuning techniques are also discussed.

This book is targeted at the architects and specialists who need to know the concepts and the detailed instructions for a successful IBM Smart Analytics System implementation and operation.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Whei-Jen Chen is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and DB2 system administration. Whei-Jen is an IBM Certified Solutions Expert in Database Administration and Application Development, and an IBM Certified IT Specialist.



Rafael Aiello is a Client Technical Specialist for Information Management in Software Group Brazil. His areas of expertise are DB2 database administration and data warehousing practices. As part of this team, he supported various projects in transactional and analytical databases subjects. Rafael was actively involved in the proof of concept and implementation of the first IBM Smart Analytics System 7600 in Brazil. He holds a bachelors degree in Computer Science and he is an IBM Certified Advanced Database Administrator for DB2 V8.1 and V9.1.



Silvio Luiz Correia Ferrari is a Senior IT Specialist and IBM Smart Analytics System Implementation Specialist in Sao Paulo, Brazil. For the past ten years, he has worked in the IBM Information Management area as a Business Intelligence consultant, providing support for customers who are planning and implementing data warehousing environments and OLAP systems.



Zohar Nissare-Houssen is an Advisory Software Analyst with IBM Canada. He is currently working with the IBM Smart Analytics System Core Engineering team, which performs the architecture, design, and performance testing for the IBM Smart Analytics System solutions on Linux® and AIX® platforms. He has been working in the IBM Toronto Lab for ten years and has an extensive experience with DB2 for Linux, UNIX®, and Windows®, specifically in the areas of problem determination, troubleshooting, and performance.

Acknowledgements

A special thanks to **David J Young** for his expertise and written content describing the operating system monitoring tools, backup and recovery. David is a Senior Accredited IT Specialist on the UKI CSL - Server Systems Operations Team, located in the United Kingdom.

A special thanks to **Joyce Coleman** for her advice, expertise, and in-depth review for the book. Joyce is a member of the IBM Smart Analytics System core engineering team who focuses on information development. She is located in the IBM Toronto laboratory, Canada.

Thanks to the following people for their contributions to this project:

Paul Bird
Rajib Sarkar
Chrissie Kwan
Bill Phu
Marcelo Arbore
Frank Goytisoló
Charles Lai
Konwen Kuan
Eddie Daghelian
Kevin Rose
Andrew Hilden
IBM Canada

Kevin Beck
Jo Ramos
Allen Kliethermes
Sermsak Sukjirawat
Gregg Snodgrass
Patrick Thoreson
IBM United State

Gregor Meyer
IBM Germany

Neil Toussaint
IBM United Kingdom

Garrett Fitzsimons
Robert Andrin
IBM Ireland

Nathan Gevaerd Colossi
Marcelo Arbore
IBM Brazil

Emma Jacobs
International Technical Support Organization, San Jose Center

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



IBM Smart Analytics System

In this chapter we introduce the IBM Smart Analytics System, including the benefits offered. We describe the features and architecture of the IBM Smart Analytics System.

We cover the following topics:

- ▶ An overview of the IBM Smart Analytics System
- ▶ The IBM Smart Analytics System portfolio

1.1 Overview

Nowadays, enterprises recognize the value of business analytics and are moving to apply these capabilities to add business value. However, implementing a data warehouse solution requires resources and expertise in business intelligence software, server hardware, storage, and the help of professional services. The traditional system implementation method for this complex integration effort costs a company both in time and money.

IBM Smart Analytics System, taking advantage of the appliance architecture, is a pre-integrated analytics system designed to deploy quickly and deliver fast time to value. Because the software is already installed and configured in the server, IBM clients are able to have their systems up and running in days instead of months. Engineered for the rapid deployment of a business-ready solution, the IBM Smart Analytics System includes the following features:

- ▶ A powerful data warehouse foundation
- ▶ Extensive analytic capabilities
- ▶ A scalable environment that is integrated with IBM servers and storage
- ▶ Set-up services and single point of support

The IBM Smart Analytics System comes in a number of offerings. IBM professionals with expertise in data warehouse applications help you select the proper IBM Smart Analytics System based on the data and user capacity needed. To add capacity over time, you can mix various generations of hardware, enabling you to protect your investment in the long term.

Every IBM Smart Analytics System offering offers a set of resources to support a complete data warehousing solution. At the heart of the IBM Smart Analytics System is a data warehouse based on DB2 Enterprise Server Edition software and the Database Partitioning Feature that incorporates best practices based on decades of IBM experience designing and implementing data warehouses.

The analytics, workload management, and performance analysis capabilities provided by the InfoSphere Warehouse software depend on the specific edition of the software that your offering includes, but in most cases include the following features:

- ▶ Data modeling and design provided through Design Studio
- ▶ Data movement and transformation provided through the SQL Warehouse Tool
- ▶ OLAP functions provided through Cubing Services
- ▶ OLAP visualization provided through Alphablox
- ▶ In-database data mining provided through Intelligent Miner and MiningBlox

- ▶ Data Mining Visualization provided through Intelligent Miner Visualization
- ▶ Unstructured text analysis provided through Text Analytics
- ▶ Integrated workload management provided through DB2 workload manager
- ▶ Deep compression for data, index, temporary tables, and XML provided by the DB2 Storage Optimization feature
- ▶ Performance tuning and analysis through DB2 Performance Expert

The analytics capabilities provided by the optional IBM Cognos 8 BI software include reporting, query, and dashboarding capabilities. These capabilities allow you to perform complex analysis on your data to identify trends in business performance, and represent your insight visually through reports or at-a-glance dashboards.

An important advantage of the IBM Smart Analytics System offerings is that they are delivered to you fully set up and configured using customized information about your environment such as the IP addresses, user names, and the database name. Before handing the system over to you, IBM professionals verify the setup, perform final quality validation, and provide training. The system is then ready for you to begin creating database objects and loading data. The time between the purchase decision and the delivery of the system to your site, ready for you to begin loading data, can be as little as two weeks!

Having the system built and configured by IBM not only speeds up the process, it ensures that every single piece of the system has been tested and verified for compatibility. This verification includes the operating system and software levels (including fix packs), firmware level of every hardware involved including switches, hard disk controllers, servers, and so on.

This solution is running on the IBM reliable server hardware platforms. Because redundant hardware components are used, there is no single point of failure on any of the servers, storage controllers, hard disks, Ethernet switches, and SAN switches, network interface cards, internal networks, power supplies, and input power. An additional high availability configuration can help a system recover from other hardware and software failures using server failover.

Another benefit for IBM Smart Analytics System customers is the single point of support. A single phone number is used for any support needed, be it for hardware or software.

IBM can also speed up the development of database models for industry solutions in many areas such as retail, insurance, banking, telecommunications, health care insurance, and health care providers.

1.1.1 Architecture

The IBM Smart Analytics System 5600, 7600, and 7700 are built upon a building block concept known as modules. Certain modules are mandatory and others are optional, depending on the quantity of data you have, your concurrency needs, and the analytics software you require. You can start from the required basic modules and add new modules when your business grows and the system requirement increases.

Figure 1-1 illustrates the concept of IBM Smart Analytics System modules.



Figure 1-1 IBM Smart Analytics System module concept

Management module

The *management module* is the starting point for all IBM Smart Analytics System offerings. The management module replaces the eliminated foundation module. This module provides the base functionality for all other modules. The basic management module contains one *management node*.

The management node is a server used to automate the process of building the other servers at installation time. It also houses management software such as IBM DS Storage Manager and DB2 Performance Expert. In certain configurations, it hosts the IBM Smart Analytics System Control Console, which provides automated system-level maintenance capability.

User modules

Each user module contains an *administration node*. Administration nodes, apart from the first one, are often called *user nodes*. The first administration node hosts a single database partition that stores the catalog tables for the core warehouse database, stores the non-partitioned data belonging to the core warehouse database, and acts as a coordinator for user connections.

User nodes can store non-partitioned data and act as a coordinator for user connections, but unlike the first administration node, they do not hold catalog tables. These nodes are optional. They can act as additional DB2 coordinator nodes by helping to balance database connections.

The first user module in an IBM Smart Analytics System is required because a configuration must have at least an administration node. The additional administration nodes are optional nodes that acts like an additional DB2 coordinator node for balancing the database connections. For example, all the user connections can be routed to the user node allowing the administration node to focus on the applications requests.

Data module

As the name implies, the data module is where the partitioned data is stored. Every data module includes one data node that hosts multiple database partitions. An IBM Smart Analytics System must have at least one data module. Depending on the IBM Smart Analytics System, there are four or eight DB2 database partitions per data module.

Failover module

The *failover module* is configured as a high availability module similar to the data module, but without storage disks. The failover module will standby and substitute for any failing administration, user, or data modules within its high availability group. Tivoli® Systems Automation for Multiplatforms constantly monitors the DB2 resources (hardware and software) and will substitute a failing module with the failover module to restore normal system operation. The failover process takes a few minutes to take place and all the uncommitted database operations are rolled back and will need to be resubmitted. Depending on the IBM Smart Analytics System server family, there is one failover module for each group of four or eight modules.

Warehouse applications module

The *warehouse applications module* is implemented using InfoSphere Warehouse software. Together with the business intelligence module, it provides analytics capability in an IBM Smart Analytics System. There can be a single warehouse applications module in an IBM Smart Analytics System.

A warehouse applications module can contain one or two nodes, the *warehouse applications node* is required, and the *warehouse OLAP node* is optional:

► Warehouse applications node:

The Warehouse applications node contains all of the InfoSphere Warehouse components that are in the application server tier of the InfoSphere Warehouse architecture.

The software components include the following InfoSphere Warehouse software components:

- InfoSphere Warehouse Administration Console
- SQL Warehousing Tool (SQW)
- Alphablox OLAP visualization tool
- Miningblox application programming interface (to extend Alphablox components with data mining functionality).

Certain components are hosted on the WebSphere Application Server software and are accessible through a web browser. The Cubing Services component can also execute in this node, if the optional OLAP node is not present.

► Warehouse OLAP node:

The warehouse OLAP node is an optional node for the warehouse applications module. It has two functions:

- Executes the Cubing Services Cube Server in a high OLAP utilization scenario.
- Allows for an active-active high availability (HA) configuration to be implemented. In this HA configuration, either node in the module can fail over to the other node.

Business intelligence module

The business intelligence module is implemented using Cognos 8 Business Intelligence software. It can contain two or more nodes where the maximum number depends on the specific offering. IBM Cognos Analytic Applications deliver the packaged reports and analysis for assessing performance of specific functional domains including finance, customer, supply chain and workforce.

These applications help you gain insight, helping you to make better business decisions and perform faster and in a far more cost effective way in each business area. The business intelligence module is available in IBM Smart Analytics System 7700, 7600, and 5600. Cognos software is included with other offerings, but not as part of a business intelligence module.

Expanding the IBM Smart Analytics System

The module architecture of the IBM Smart Analytics System provides flexibility in expanding your system as the business grows. As the customer database activities increase, new modules can be added:

- ▶ Add data modules to the existing system when the data volume increases.
- ▶ Extend your business intelligence module by adding more business intelligence extension nodes to manage to manage increased report users.
- ▶ Add user modules to manage a large number of users and to balance the data accessing load.

Figure 1-2 shows the IBM Smart Analytics System building block examples.

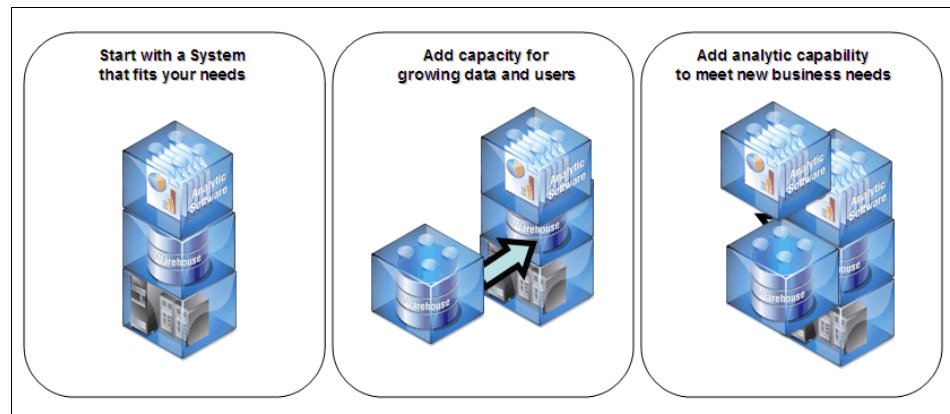


Figure 1-2 IBM Smart Analytics System building block examples

1.2 IBM Smart Analytics System portfolio

The IBM Smart Analytics System family offers a wide range of hardware platforms and architectures to provide customers with an optimal data warehousing system for their business size. From a small, all-in-one Linux or Windows powered server, to an AIX or mainframe enterprise solution, IBM Smart Analytics System is the perfect data warehouse solution.

1.2.1 IBM Smart Analytics System 1050 and 2050

The IBM Smart Analytics System 1050 and 2050 are the entry level solutions that are intended for midsize businesses and departmental usage.

The IBM Smart Analytics System 1050 is a single server system appropriate for database sizes ranging from 300 GB (using only internal disks) to 3.3 TB of data (using a dedicated storage controller). The sizes mentioned are for user space.

The IBM Smart Analytics System 2050 is the next step up in the family. It is also a single server system that is designed for database sizes from 3.3 TB to 13.2 TB. This system uses up to four dedicated storage controllers.

Both systems employ IBM System x® servers (Intel® based) and can be installed on Novell SUSE Linux 11 or Windows Server 2008. The Cognos Business Intelligence is offered as an optional feature.

1.2.2 IBM Smart Analytics System 5600

The IBM Smart Analytics System 5600 product family is built upon IBM System x hardware and uses SUSE Linux as the operating system. IBM Smart Analytics System 5600 is the IBM solution for medium to large companies that need powerful analytics capabilities and growth flexibility at an exceptional price-to-performance ratio.

The IBM Smart Analytics System 5600 data modules, when configured using the standard 300 GB disks, can store 6 TB of user space. For an increased data density, 450 GB and 600 GB disks are available.

The IBM Smart Analytics System 5600 has two offerings: 5600 V1 and 5600 V2.

IBM Smart Analytics System 5600 V1

The IBM Smart Analytics System 5600 V1 offering uses System x3650 M2 servers and DS3400 storage. In the standard configuration for this offering, each x3650 M2 server is configured with one quad-core processor and 32 GB of memory. Each data node is attached to two DS3400 external storage servers with 300 GB disks, for a total of 12 TB of user space per data node. For increased data density, 450 GB and 600 GB disks are available.

Figure 1-3 depicts a common IBM Smart Analytics System 5600 V1 layout.

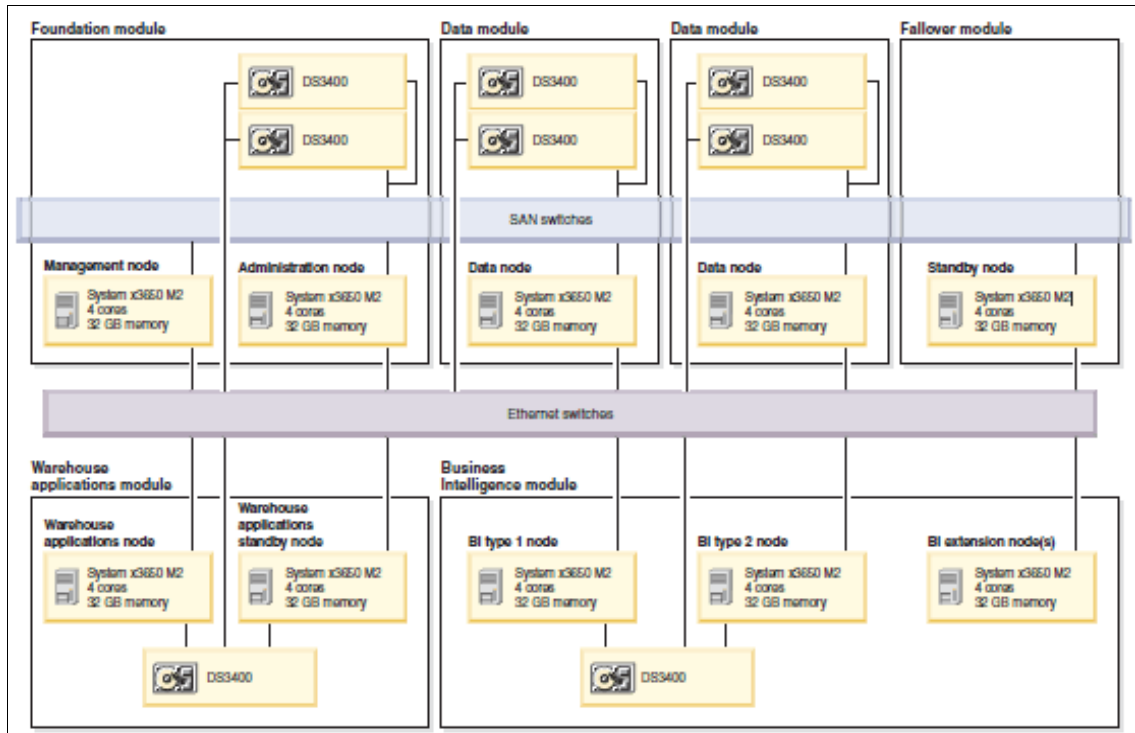


Figure 1-3 IBM Smart Analytics System 5600 V1 layout

The IBM Smart Analytics System 5600 V1 with SSD option is a more powerful version of this offering. This option adds one additional quad-core processor, an additional 32 GB of memory, and 640 GB of Solid State Devices (SSD) to each administration, data, and standby node. With this option, the DS3400s use 450 GB disks as standard, for a total of 9 TB of user space per data node.

IBM Smart Analytics System 5600 V2

The IBM Smart Analytics System 5600 V2 offering uses System x3650 M3 servers and DS3524 storage. In the standard configuration for this offering, each x3650 M3 server is configured with one six-core processor and 64 GB of memory (except for the management node, which has only 32 GB of memory). Each data node is attached to two DS3524 external storage servers with 300 GB or 600 GB disks, for a total of 12 TB to 24 TB of user space per data node.

Figure 1-4 depicts a common IBM Smart Analytics System 5600 V2 layout.

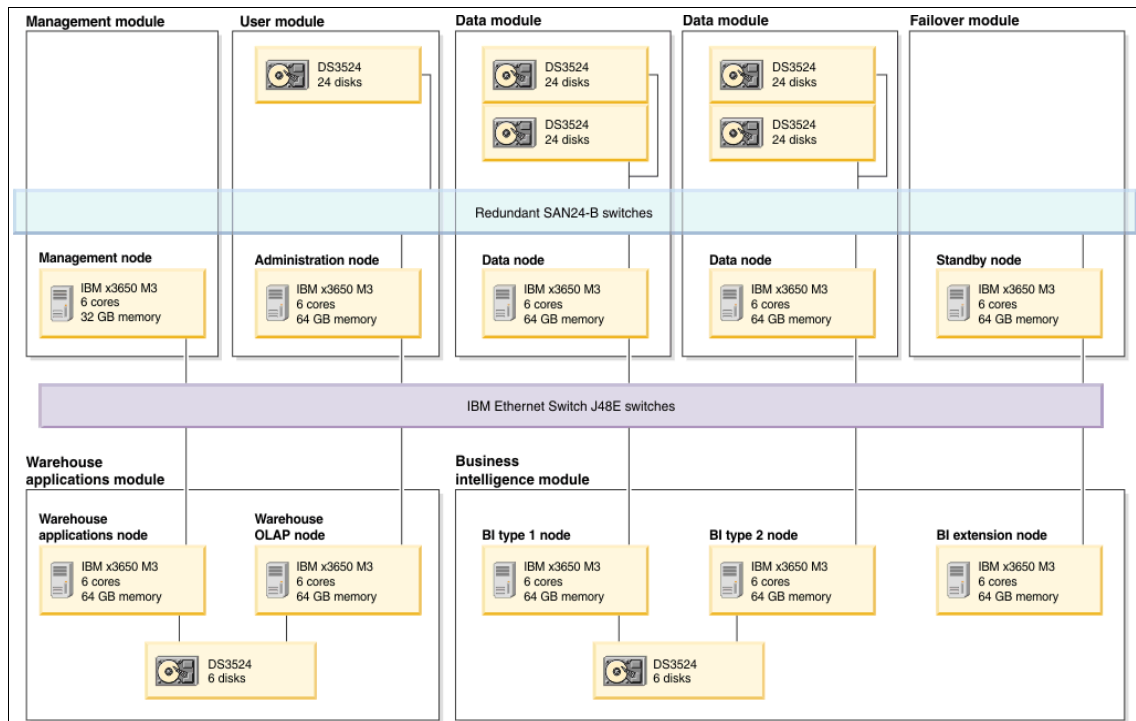


Figure 1-4 IBM Smart Analytics System 5600 V2 layout

The IBM Smart Analytics System 5600 V2 with SSD option is a more powerful version of this offering. This option adds one additional six-core processor, an additional 64 GB of memory, and 640 GB of Solid State Devices (SSD) to each administration, data, and standby node.

1.2.3 IBM Smart Analytics System 7600 and 7700

The IBM Smart Analytics System 7600 and 7700 offerings are designed for enterprise-wide reporting and analytics purpose. Members of this family use IBM POWER processors for mission critical performance and reliability, supporting complex workloads for large number of concurrent users.

The IBM Smart Analytics System 7600 utilizes the POWER6 processors, whereas the new Smart Analytics 7700 are POWER7 processor-based systems.

The IBM Smart Analytics 7600 has 4 TB of user space per data module. The IBM Smart Analytics 7700 systems can store 28 TB of user storage per data module.

Another benefit of the IBM Smart Analytics 7700 is that (as standard) it comes with 800 GB of Solid State Devices per data module, and provides an optional expansion to 4.8 TB.

Figure 1-5 illustrates the IBM Smart Analytics System 7600 layout. Each data module is allocated two EXP5000 disk enclosures. The administration node also houses an extra spare disk expansion drawer that houses hot spare drives.

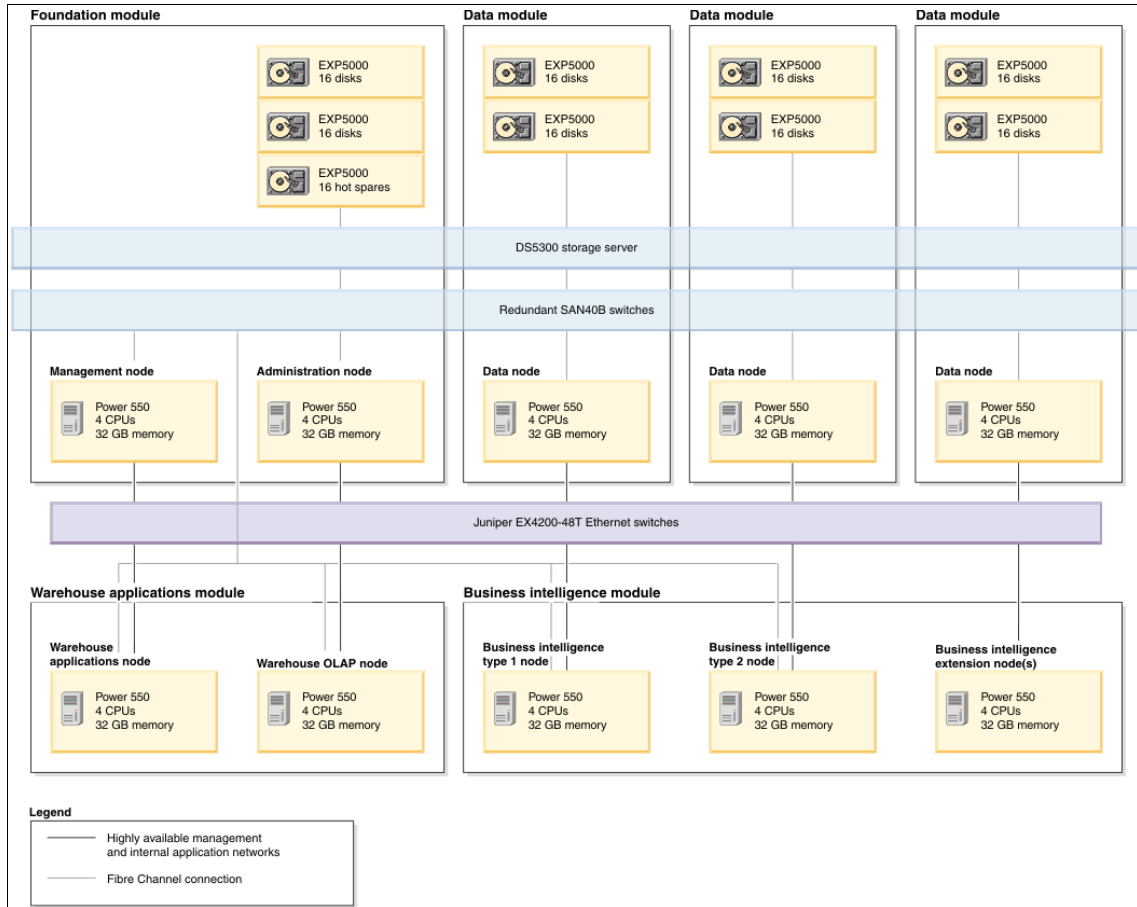


Figure 1-5 IBM Smart Analytics System 7600 layout

The IBM Smart Analytics System 7700 has a unique configuration, compared to the IBM Smart Analytics System 7600. Each 7700 data node has eight DB2 database partitions and is allocated four DS3524 storage servers. Each database partition is allocated half of a DS3524.

Figure 1-6 shows the IBM Smart Analytics System 7700 layout.

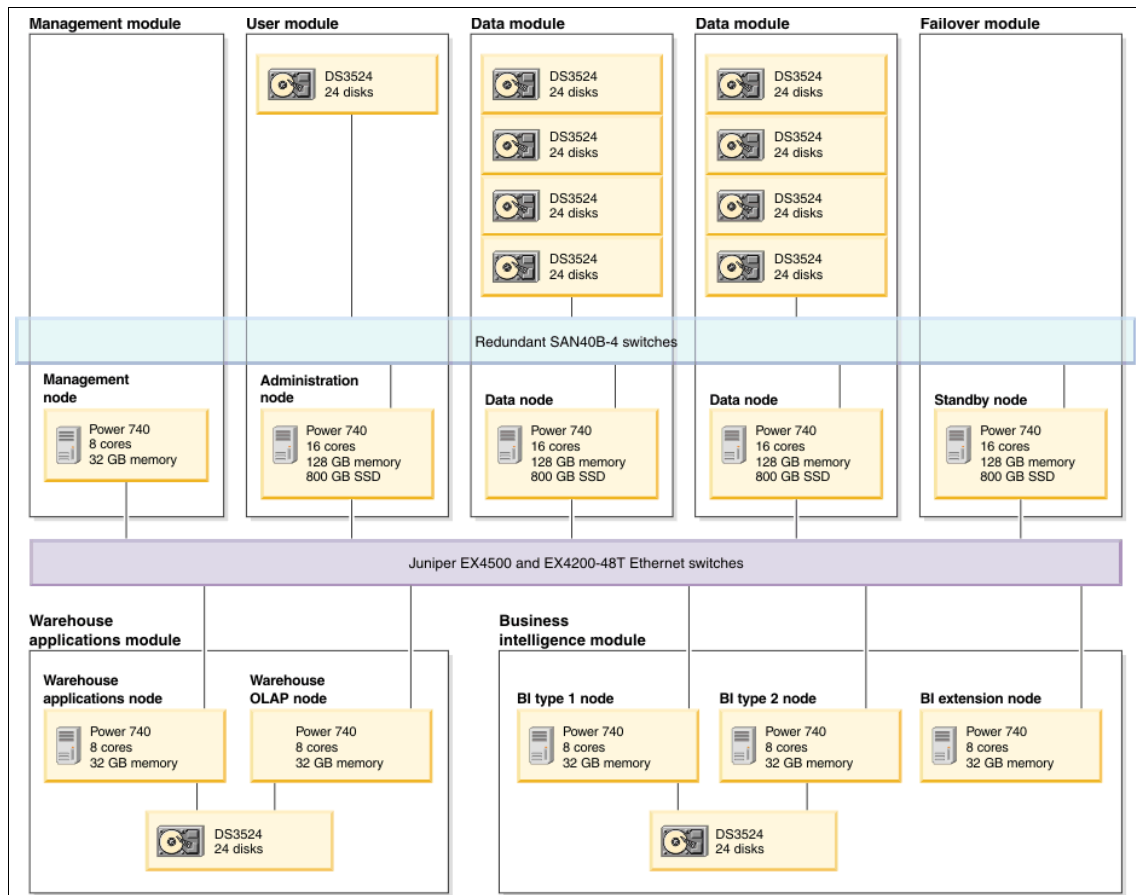


Figure 1-6 IBM Smart Analytics System 7700 layout

The IBM Smart Analytics System 7700 also has a unique configuration for the administration module. The administration node is allocated one DS3524 storage server, where half of the space is used for a single database partition that stores non-partitioned data and acts as a coordinator for user connections, and the other half is used for GPFS-shared directories.

1.2.4 IBM Smart Analytics System 9600

The IBM Smart Analytics System 9600 combines the availability and security of System z® with the characteristics of an appliance that leverages analytic information to the mainframe business.

In this book, we focus on the IBM Smart Analytics System offerings 5600, 7600, 7700 which are all based on IBM DB2 for Linux, UNIX, and Windows. For more information about the IBM Smart Analytics System 9600, go to these web addresses:

<http://www.ibm.com/software/data/infosphere/smart-analytics-system/>
<http://www-01.ibm.com/software/data/infosphere/warehouse-z/>

1.2.5 IBM Smart Analytics System family summary

Table 1-1 summarizes the key features of the models in the IBM Smart Analytics System family.

Table 1-1 IBM Smart Analytics family comparison

Parameter	1050	2050	5600 V1	5600 V1 with SSD	5600 V2	5600 V2 with SSD	7600	7700
Operating System	Win 2008 SuSE Linux 11	Win 2008 SuSE Linux 11	SuSE Linux 10 SP2 or SP3	SuSE Linux 10 SP2 or SP3	SuSE Linux 10 SP3	SuSE Linux 10 SP3	AIX 6.1	AIX 6.1
Server type	IBM System x3500 M3	IBM System x3850 X5	IBM System x3650 M2	IBM System x3650 M2	IBM System x3650 M3	IBM System x3650 M3	IBM Power 550®	IBM Power 740
DB2 database partitions per data node	1 to 4 partitions	4 to 16 partitions	4	4	8	8	4	8
Ext. storage per data node	0 to 1 DS3524	1 to 4 DS3524	2x DS3400	2x DS3400	2x DS3524	2x DS3524	2x EXP5000	4x DS3524
CPUs per data node	1x four-core or 1x six-core Intel Xeon®	2x four-core or 2x six-core Intel Xeon	1x four-core Intel Xeon X5570	2x four-core Intel Xeon X5570	1x six-core Intel Xeon X5680	2x six-core Intel Xeon X5680	2x dual-core POWER6	2x eight-core POWER7
Memory per data node	16 GB to 24 GB	16 GB to 64 GB	32 GB	64 GB	64 GB	128 GB	32 GB	128 GB
User space per data node	300 GB to 3,3 TB	3.3 TB to 13.2 TB	6 TB to 12 TB	6 TB to 12 TB	12 TB to 24 TB	12 TB to 24 TB	4 TB	28 TB
SSDs per data node	N/A	N/A	N/A	2x FusionIO (640 GB)	N/A	2x FusionIO (640 GB)	N/A	700 GB, up to 4.2 TB

1.3 IBM training

Available from IBM training are the newest offerings to support your training needs, enhance your skills, and boost your success with IBM software. IBM offers a complete portfolio of training options including traditional classroom, private onsite, and eLearning courses. Many of our classroom courses are part of the IBM “Guaranteed to run program,” ensuring that your course will never be canceled. We have a robust eLearning portfolio including Instructor-Led Online (ILO) courses, Self Paced Virtual courses (SPVC), and traditional Web Based Training (WBT) courses. A perfect complement to classroom training, our eLearning portfolio offers something for every need and every budget; simply select the style that suits you.

Be sure to take advantage of our custom training plans to map your path to acquiring skills. Enjoy further savings when you purchase training at a discount with an IBM Education Pack, online account, which is a flexible and convenient way to pay, track, and manage your education expenses online.

The key education resources listed in Table 1-2 have been updated to reflect the IBM Smart Analytics System. Check your local Information Management Training website or chat with your training representative for the most recent training schedule.

Table 1-2 *InfoSphere Warehouse courses*

Course title	Classroom	Instructor-Led Online	Self Paced Virtual Classroom	Web Based Training
InfoSphere Warehouse 9 Components	DW352	3W352	2W352	1W352
InfoSphere Warehouse 9 - SQL Warehouse Tool and Administration Console	DWA52	3WA52	2WA52	1WA52
InfoSphere Warehouse 9 - Cubing Service	DWB52	3WB52	2WB52	1WB52
InfoSphere Warehouse 9 - Data Mining and Unstructured Text Analysis	DWC52	3WC52	2WC52	1WC52
Introduction to TSA in IBM Smart Analytics Systems	DW040	3W040		1W040
Advanced TSA within an IBM Smart Analytics System	DW331	3W331	2W331	1W331

Descriptions of courses for IT professionals and managers are available at:
http://www.ibm.com/services/learning/ites.wss/tp/en?pageType=tp_search

Visit <http://www.ibm.com/training> or call IBM training at 800-IBM-TEACH (426-8322) for scheduling and enrollment.

1.3.1 IBM Professional Certification

Information Management Professional Certification is a business solution for skilled IT professionals to demonstrate their expertise to the world. Certification validates skills and demonstrates proficiency with the most recent IBM technology and solutions.

1.3.2 Information Management Software Services

When implementing an Information Management solution, it is critical to have an experienced team involved to ensure that you achieve the results you want through a proven, low risk delivery approach. The Information Management Software Services team has the capabilities to meet your needs, and is ready to deliver your Information Management solution in an efficient and cost effective manner to accelerate your Return On Investment.

The Information Management Software Services team offers a broad range of planning, custom education, design engineering, implementation and solution support services. Our consultants have deep technical knowledge, industry skills, and delivery experience from thousands of engagements worldwide. With each engagement, our objective is to provide you with a reduced risk, and expedient means of achieving your project goals. Through repeatable services offerings, capabilities, and best practices leveraging our proven methodologies for delivery, our team has been able to achieve these objectives and has demonstrated repeated success on a global basis.

The key Services resources listed in Table 1-3 are available for InfoSphere Warehouse.

Table 1-3 InfoSphere Warehouse services

Information Management Services offering	Short description
IBM Smart Analytics System Services	Foundation Services as part of the turn-key hardware, software solution; the goal is to deliver a Data Warehouse in a table ready state.
InfoSphere Warehouse Data Mining	This is a rapid deployment services for Data Mining focusing on a specific business case and limited sources of data for existing InfoSphere Warehouse customers.
InfoSphere Warehouse Data Migration	The objective is to migrate the data from an existing DB2 warehouse to a new InfoSphere warehouse (that is, the IBM Smart Analytics System) leveraging the IBM Data Movement Tool (IM Lab Services Asset).
Capacity Planning for an existing Data Warehouse	The objective is to evaluate the current DB2 Warehouse environment, to understand the new goals and to provide guidelines for updated design, hardware, and software configuration to meet those goals.
InfoSphere Warehouse HealthCheck	This service includes a complete review of the database configuration, the operating system, the storage subsystem, and operational considerations.
Data Warehouse Performance Optimization	This service is in response to a specific performance problem. The scope is limited to the analysis of maximum three load process or maximum five slow queries.

For more information, visit our website:

<http://www.ibm.com/software/data/services>

1.3.3 IBM Software Accelerated Value Program

The IBM Software Accelerated Value program provides support assistance for issues that fall outside normal “break-fix” parameters addressed by the standard IBM support contract, offering customers a proactive approach to support management and issue resolution assistance through assigned senior IBM support experts who know your software and understand your business needs. Benefits of the Accelerated Value Program include:

- ▶ Priority access to assistance and information
- ▶ Assigned support resources
- ▶ Fewer issues and faster issue resolution times
- ▶ Improved availability of mission-critical systems
- ▶ Problem avoidance through managed planning
- ▶ Quicker deployments
- ▶ Optimized use of in-house support staff

To learn more about IBM Software Accelerated Value Program, visit our website:

<http://www.ibm.com/software/data/support/acceleratedvalue/>

To talk to an expert, contact your local Accelerated Value Sales Representative at this website:

<http://www.ibm.com/software/support/acceleratedvalue/contactus.html>

1.3.4 Protect your software investment: Ensure that you renew your Software Subscription and Support

Complementing your software purchases, Software Subscription and Support gets you access to our world-class support community and product upgrades, with every new license. Extend the value of your software solutions and transform your business to be smarter, more innovative, and cost-effective when you renew your Software Subscription and Support. Staying on top of on-time renewals ensures that you maintain uninterrupted access to innovative solutions that can make a real difference to your company's bottom line.

To learn more, visit:

<http://www.ibm.com/software/data/support/subscriptionandsupport>



Installation and configuration

In this chapter we describe the installation, configuration, and deployment processes of an IBM Smart Analytics System. We provide a brief explanation of the planning process, highlighting the details to be considered before IBM builds an IBM Smart Analytics System. We also describe the installation process conducted at the IBM Customer Solution Center and at the customer's data center.

2.1 Planning

The IBM Smart Analytics System offerings are pre-integrated systems with the installation and configuration conducted at the IBM Customer Solution Center (CSC) based on the information collected from the customer with the assistance of IBM specialists. The system is then shipped and deployed to the customer site.

Most of the information required for building the system is collected and kept in the IBM Smart Analytics System *customer worksheet*. In addition, IBM and customer architects will design a floor diagram that describes server placement and data center environment requirements.

In this section, we briefly describe the information collected for building an IBM Smart Analytics System. We highlight the details that should be considered during this planning stage to ensure a smooth deployment.

2.1.1 Smart Analytics System customer worksheet

The customer worksheet provides the baseline for the IBM Smart Analytics System build and deployment. Each IBM Smart Analytics System model has a customer worksheet designed specifically for that model. Though the information collected can be similar, the worksheet layouts vary.

Offerings: This book focuses only on the IBM Smart Analytics System 5600, 7600, and 7700 offerings. These offerings are all based on IBM DB2 for Linux, UNIX, and Windows.

The information required for building an Smart Analytics System can be briefly categorized into four main groups:

- ▶ Server information
- ▶ Network information
- ▶ Database and operating system configuration information
- ▶ Data center and system delivery information

Server information

The customer worksheet specifies the information about all management, administration, data, standby, warehouse applications, and business intelligence nodes in the system.

Figure 2-1 shows an example of the IBM Smart Analytics System component configuration section for IBM Smart Analytics System 7700.

IBM Smart Analytics System 7700 R1.0 Customer Worksheet
Version 1.3.4, Last Updated: Sept 12, 2010 6:00PM

IBM and customer's confidential when completed. This worksheet should be filled before placing an order for IBM Smart Analytics System. The information supplied in this document will be used by IBM Manufacturing and Customer Solution Center to install and configure software in IBM Smart Ar IBM will not be able to start installing your system until all required information in this worksheet are filled.

Please enter information in required green and optional light blue fields below

Green fields contain the minimum information required by IBM to install and pre-configure your system in our Manufacturing and Customer Solution Center Light blue are optional fields, not required by IBM to successfully pre-configured your system in our facility. However, they may be needed to complete the se Important: Please review and ensure that you entered all information correctly. An incorrect information in this worksheet could mean that software and oper may have to be reinstalled and reconfigured at your additional cost.

STEP 1: To avoid accidentally over-written default calculation, please save your spreadsheet as another name. Then, enter pieces of

Servers	Number of servers	No. of cores per node	Memory per node
Management Module	1	16	128
Administration Module	1	16	128
User Module	0	n/a	n/a
Data Module	4	16	128
Fallover Module	1	16	128
Application (Analytics) Module	2	16	128
Cognos BI Module	0	n/a	n/a

Other equipment	Pieces of equipment (Update if needed)
HMC	2
RSM	1
DS3500	18
SAN40B Switches	4
1 GbE Network Switches	2
10 GbE Network Switches	2

Additional Information

Please indicate number of BLUEDARTER adapters (for SSD drives) per Data Module

Please specify DS3500 drive size GB

Do you have LAN-free backup HBA Adapter?

DB2 version Note: All software components will be installed per the IBM Smart Analytics System 77

Environment Type

Customer Worksheet / Optional HMC Worksheet / Smart Analytics Private Network / Customer Netw-Fibre Requiremen / Software

Figure 2-1 Components information listed on customer worksheet for IBM Smart Analytics System 7700

Network information

The network information section specifies the corporate network used by administrators, end users, and client applications that will access the IBM Smart Analytics System. The required information includes IP addresses, host names, gateway, and dynamic name server (DNS) information.

There are various types of networks used in the IBM Smart Analytics System offerings that can be categorized as follows:

- ▶ Internal networks such as the DB2 Fast Communications Manager (FCM) network and, on AIX only, the Hardware Management Console (HMC) network
- ▶ Public networks such as customer corporate and application networks

The IP addresses should also be planned for future expansion; remember that IBM Smart Analytics System has the capacity to grow in a modular basis, to scale-out the environment. The network information must be prepared to accommodate the new modules.

Figure 2-2 shows an example of the customer worksheet fields for host names, IP addresses, DNS, and gateway information listed on a customer worksheet for IBM Smart Analytics System 7700.

STEP 2: Enter hostname and User Network's IP Addresses information

Definition: User network refers to your corporate network used by administrators, end users and client applications that will access the IBM Smart Analytics System.
In a default configuration, each POWER System 740 server is connected to your user network using two 10 Gbit Ethernet ports in an active/passive configuration. If, instead, you need to connect the user network using 1Gb Ethernet port(s), please select "1 Gbit User Network" (In this case, your server must have at least one 10Gbit User network to servers)
 All administration user's connections to other devices-- such as HMC, Juniper network switches, and RSM-- remain on the 1Gbit network (CAT5 or CAT6 cables).

Click this button before proceeding Sample values are provided below. Please overwrite the values with your own.

Server	Hostname (Must be <= 15 characters, lower case, and NOT contain underscore)	First 3 octet of User Network IP Address	Last octet of User Network IP Address	Subnet Mask	User Network IP Address	Default Gateway IP Address
Management Server	smasmgt01	9.10.23	10	255.255.252.0	9.10.23.10	9.10.23.1
Admin Server	smasedw01	9.10.23	11	255.255.252.0	9.10.23.11	9.10.23.1
Data Module 1	smasedw02	9.10.23	12	255.255.252.0	9.10.23.12	9.10.23.1
HA Standby 1	smasedw04	9.10.23	14	255.255.252.0	9.10.23.14	9.10.23.1
1Gb Network Switch Chassis	jnpr4200sw1	9.10.23	17	255.255.252.0	9.10.23.17	9.10.23.1
10Gb Network Switch 1	jnpr4500sw1	9.10.23	18	255.255.252.0	9.10.23.18	9.10.23.1
10Gb Network Switch 2	jnpr4500sw2	9.10.23	19	255.255.252.0	9.10.23.19	9.10.23.1
RSM	smasrsm01	9.10.23	20	255.255.252.0	9.10.23.20	9.10.23.1
HMC 1	smashmc01	9.10.23	21	255.255.252.0	9.10.23.21	9.10.23.1
HMC 2	smashmc02	9.10.23	22	255.255.252.0	9.10.23.22	9.10.23.1

Tivoli System Automation Service IP Addresses for the User Network

User Network Service IP	Name Alias (Must be <= 18 characters)	First 3 octet of User Network IP Address	Last octet of User Network IP Address	Subnet Mask	User Network Service IP Address
Admin Node's DB2 Service IP	smasedw01sip	9.10.23	23	255.255.252.0	9.10.23.23

DS3500 Storage information

By default, each DS3500 will be connected to the Smart Analytics System's internal network using private IP addresses. A storage administrator can access to each DS3500 through the Smart Analytics System's internal network from the management server's X-windows. Therefore, it is not mandatory that DS3500s need to be connected into your user network.

Customer Worksheet / Optional HMC Worksheet / Smart Analytics Private Network / Customer Network-Fibre Requirements / Software & Firmware Stack / IBM Use Only

Figure 2-2 The network information listed on customer worksheet

Database and operating system configuration information

Database and operating system configuration information collected includes the operating system users, groups, user IDs, and groups IDs. The DB2 instance and database information, with details about IBM InfoSphere Warehouse users and groups, is also required because the system deployment will install and configure a DB2 instance and will create the customer database.

UIDs and GIDs: Be sure to provide the user identification numbers (UIDs) and group identification numbers (GIDs) that are not being used in the existing enterprise to avoid the conflict when the IBM Smart Analytics System is deployed and allow a seamless integration of the IBM Smart Analytics System with the existing customer environment.

Figure 2-3 shows an example of the customer worksheet section for the user and group information for an IBM Smart Analytics System 7700.

TIMEZONE	America/New_York				
AIX Locale	en_US	Default is en_US			
The following table is for your information only. Changes from the default are not necessary.					
System admin Users	User Name	Password (<= 10 characters and contain an alpha numeric character)			
AIX Administrator	root	Ch@ngeme1			
DS3500	n/a	Ch@ngeme1			
SAN40B	admin	Ch@ngeme1			
Juniper Network switches	root	Ch@ngeme1			
	admin	Ch@ngeme1			
HMC	hscroot	Ch@ngeme1			
	root	Ch@ngeme1			
RSM	admin	passw0rd			
STEP 5: DB2 Configuration Worksheet					
Please update the "green" fields below.					
InfoSphere Warehouse Users	User Name (Must be <=8 characters and lowercase)	Password (<= 10 characters)	Group Name (FYI only. Do not change)	UID	GID
DB2 Instance Owner	bcuaix	Ch@ngeme1	bcuigrp	1100	999
DB2 Fenced User	bcufenc	Ch@ngeme1	bcufgrp	1101	998
DB2 Administration Server	bcudasp	Ch@ngeme1	daspgrp	1102	997
InfoSphere Warehouse Administrator	dweadmin	Ch@ngeme1	dweagrp	1104	996
			dwemgrp	n/a	995
			dweogrp	n/a	994
InfoSphere Warehouse WAS Admin	wasadmin	Ch@ngeme1	wasagrp	1105	993
DB2 Performance Expert Instance Owner	bcupe	Ch@ngeme1	bcupegrp	1106	992
DB2 Performance Expert Fenced User	bcufpe	Ch@ngeme1	bcufpgrp	1107	991
<p>► \Customer Worksheet / Optional HMC Worksheet / Smart Analytics Private Network / Customer Netwk-Fibre Requiremen / Software & Firmware Stack / IBM</p>					

Figure 2-3 Example of user and group information listed on customer worksheet

Data center and system delivery information

The data center and delivery information section specifies the shipping information and data center details for deploying the system, such as customer data center cable racks position (overhead or underfloor racks).

Figure 2-4 shows an example of data center and delivery information section for IBM Smart Analytics System 7700.

STEP 8: Data Center and Delivery Information

Power Supply and Cable Management
 Where do network cables run in your data center?
 Where do fibre cables run in your data center?
 Where do power supply cords run in your data center?

Underfloor
Overhead above racks
Underfloor

Delivery Information

Customer name:	
Delivery address: (Street Address, City, State, Zip code, Country)	
Primary contact:	
Primary contact telephone :	
Installation contact:	
Installation contact telephone:	
Alternate contact:	
Alternate contact telephone:	

Delivery access:

Appointment required: NO YES Hours: between

Secured facility: NO YES

Loading dock: YES

Truck size options:
 53 ft 24 ft 18 ft Flat Bed (Optional)

Lift gate required: NO YES

Elevator: NO YES

If there is an elevator, has the maximum weight limit been checked? NO YES

Maximum weight per rack is approximately 1800 lbs.

Door access:

Customer Worksheet / Optional HMC Worksheet / Smart Analytics Private Network / Customer Netwk-Fibre Requiremen / Software

Figure 2-4 Data center and delivery information listed on customer worksheet

Hardware Management Console and the Remote Support Manager

The Hardware Management Console (HMC) is a required component of IBM Smart Analytics System 7600 and 7700 (AIX based) and is supported on these two configuration only. The Remote Support Manager for Storage is a required component of 7600 and 7700, and is optional for the 5600 offering.

The IBM Smart Analytics System can take the advantage of features such as the call home support of the Hardware Management Console (HMC) and the Remote Support Manager (RSM) for IBM Storage Systems. The information required to implement these components should be provided prior to the system deployment. When the system is deployed at the customer site, it will be ready to take proactive actions.

Figure 2-5 illustrates an example of the HMC optional settings for IBM Smart Analytics System 7700 configurations.

Preinstallation configuration worksheet for the HMC

You may optionally complete this worksheet so that IBM can complete call-home connectivity for the Hardware Management Console. If the information in this worksheet is not completed, IBM may not be able to complete call-home and notification configuration on the HMC.

More information about planning, installing and configuring the HMC can be found at:
<http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/iphai/iphai.pdf>

Local host information

Some information is pre-filled based on information you provided in the "Customer Worksheet" tab. You may provide additional information.

	HMC1	HMC2	HMC3	HMC4
HMC hostname:	smashmc01	smashmc02	N/A	N/A
Domain name:	<enter info here>	<enter info here>	N/A	N/A
Description of HMC:				
Gateway address (nnn.nnn.nnn.nnn)	9.10.23.1	9.10.23.1	N/A	N/A
Gateway device:				

Do you want to use DNS? (yes/no)				
If "yes", specify DNS Server Search Order below:				
1				
2				
Domain suffix search order:				
1				
2				

Local Host Information
 To identify your Hardware Management Console (HMC) to the network, enter the HMC's host name and domain name. Unless you are fully qualified host name. Domain name example: name.yourcompany.com

Gateway Information
 To define a default gateway, fill in the TCP/IP address to be used for routing IP packets. The gateway address informs each computer is not located on the same subnet as the source.

DNS Enablement

Customer Worksheet
 Optional HMC Worksheet
 Smart Analytics Private Network
 Customer Network-Fibre Requirements

Figure 2-5 HMC information listed on Customer Worksheet for IBM Smart Analytics System 7700

2.1.2 Floor diagram and specification review

As part of the installation planning, the IBM team works with the customer to build a rack diagram with all IBM Smart Analytics System components and prepare an environment worksheet that specifies the environment requests such as the floor space, power, and cooling needed for the IBM Smart Analytics System components.

When the system is delivered, the physical space should be available with all the environment requirements fulfilled for the deployment process.

2.2 Installation of IBM Smart Analytics System

The installation service is part of the IBM Smart Analytics System contract package and it is governed by an IBM Statement of Work. After the order is placed and all information needed is provided on the customer worksheet, the IBM Smart Analytics System Services assembling process is ready to start. This process can be divided into two main activities:

- ▶ IBM CSC assembles, installs, configures, and tests the system. The integrated system is then shipped to the customer's data center.
- ▶ At the customer's data center, the racks are reconnected and tested to ensure that the system meets the specifications.

2.2.1 Installation at the IBM Customer Solution Center

The customer worksheet generates the input files needed by the automatic installer to properly deploy the custom IBM Smart Analytics System ordered.

The following main activities are performed at the IBM facility:

- ▶ Cabling, installing, and testing following the IBM Smart Analytics System specification:
 - The IBM specialists will prepare the servers in the rack, do the initial cabling, and install the base software for the system.
- ▶ Storage subsystem configuration:
 - The IBM Storage subsystem will be configured following the IBM Smart Analytics System practices. Also it will be loaded with the validated software and microcode stack.
- ▶ System optimization:
 - The IBM specialists will bring the system to the IBM Smart Analytics System standard configuration. The system settings, such as operating system and storage parameters, will then be applied.
- ▶ Database configuration:
 - Following the information given by the customer, the customer database will be created. This base setup follows the IBM Smart Analytics System best practices.
- ▶ Quality assurance:
 - When the system is ready, it will be thoroughly tested and checked to see if it meets the IBM Smart Analytics System standards.

- ▶ Installation report:

At the end of the installation and configuration, a report will be generated with all the information gathered during the installation process. The report also documents all the system information, from the architecture overview to the network point-to-point diagram. This report will be updated after the deployment process at the customer site before the system is turned over to the customer. The installation report should be used as a reference when contacting the IBM Customer Support for any questions regarding the system.

- ▶ Packing and shipping to the customer:

When the system is properly installed, tested, and documented, it is ready to be shipped to the customer site.

2.2.2 Installation at the customer site

Before an IBM Smart Analytics System is ordered, IBM has worked with the customer to evaluate the data center requirements for the system. When the system arrives at the customer site, the data center should be ready for deploying the servers. IBM will coordinate with the customer to perform this final installation task.

The main activities are as follows:

- ▶ Power up the systems:

This is the systems startup. At this point, IBM specialists check the health of the IBM Smart Analytics System components.

- ▶ Perform internal system cabling:

The internal cabling is reconnected after the servers are powered up and tested by IBM specialists.

- ▶ Rerun the installation and performance tests to ensure the system is performing as expected:

The overall system is thoroughly tested again to ensure the environment is performing as expected.

- ▶ Complete the final quality assurance check:

After the test results are gathered, the final checklist is updated.

- ▶ Perform mentoring and knowledge transfer session:
A skill transfer session is conducted by the IBM professionals for customers about the IBM Smart Analytics System. The information provided in the session includes the system overview, the installation report, how to deal with the new environment, and where to gather further information.
- ▶ Update the installation report:
The final update is done at the installation report with all documented results and system information. The installation report is given to the customers when the system is turned over to them.

At this point the system is at the “ready-to-load data” state.

2.3 Documentation and support for the IBM Smart Analytics System

The IBM Smart Analytics System product documentation is a good starting point to learn more about the new system. You can download the documentation from this web address:

https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=idwbcu

The IBM Smart Analytics System documentation provides the following information:

- ▶ Managing users and passwords:
When the system is delivered it comes with the default passwords. These password should be changed according to your IT regulations.
- ▶ Table space design considerations:
Look at this topic to know more about how to create table spaces. To ease the system administration tasks, IBM Smart Analytics System takes advantage of DB2 Automatic Storage feature to manage database space automatically.
- ▶ DB2 workload manager (WLM):
It is extremely important to implement WLM to protect the system from overload, rogue queries, and to ensure the SLAs requirements are met. To manage the system workload, IBM Smart Analytics System explores the DB2 WLM feature.

IBM Smart Analytics System installation report

During the installation and deployment of the IBM Smart Analytics System, many details are generated about the system architecture, servers, network, operating system, software stack, performance tests, and so on. All this information is collected and documented in a worksheet called *IBM Smart Analytics System installation report*.

On the installation report, you can find information such as this:

- ▶ Architecture and hardware profile
- ▶ Rack diagram
- ▶ Software stack
- ▶ Networking configuration
- ▶ Storage configuration
- ▶ High Availability configuration
- ▶ Network point-to-point
- ▶ Fiber point-to-point

This report is delivered by IBM when the system is turned over to the customer. You should update this installation report with system changes performed on the system, for example, database parameters settings, if changed. This report also can be used as a reference and be updated by the IBM Smart Analytics System Health Check services.

IBM Smart Analytics System installation report provides a single view of the entire system stack (hardware, software and configuration). This report will be useful and needed in case of opening a Problem Management Report (PMR), for example.

When opening a PMR, have as much information as you can about the situation and the environment. For example, to report a disk problem, you will need to inform the Type, Model and S/N of the storage system and disk enclosure, for example. This information can be found on the installation report. If RSM is enabled, it can handle all these, including opening the PMR.

If a copy of the installation report is needed by the customer, contact IBM Smart Analytics Customer Support for a copy.

For the most update information about IBM Smart Analytics System, see the IBM website at this address:

<http://www.ibm.com/software/data/infosphere/smart-analytics-system/>



High availability

The IBM Smart Analytics System offerings each include high availability solutions that automate failover from any active node to another node in the cluster. Together with the many redundant hardware components in the system, these high availability features can minimize the down time caused by many hardware and software problems.

In this chapter we describe the high availability characteristics present on the IBM Smart Analytics System.

3.1 High availability on IBM Smart Analytics System

The IBM Smart Analytics System offers the high availability capabilities through both the hardware and software features. The IBM Smart Analytics System is designed with redundant hardware components to help minimizing the downtime through single points of failure with any one hardware component. The following components are designed with redundancy:

- ▶ Disk mirroring for internal storage (mirrored volume groups)
- ▶ RAID disk arrays for external storage
- ▶ Dual port Fibre Channel adapters
- ▶ Redundant SAN switches
- ▶ Dual-port network adapters
- ▶ Redundant network switches
- ▶ Dual active RAID controllers
- ▶ Dual hot-swappable power/cooling units

The IBM Smart Analytics System utilizes IBM Tivoli System Automation for Multiplatforms (SA MP) to provide or extend the high availability features at the software or application level. IBM Tivoli SA MP integration provides the capability to take specific actions when a detectable resource failure occurs. This action can be as simple as restarting a software components or moving a software components to the standby node. A resource failure can include:

- ▶ Network failure
- ▶ A server failure caused by accidentally rebooting or power failure
- ▶ DB2 instance crash
- ▶ Database partitions failure

There is a separate Tivoli System Automation for Multiplatforms high availability configuration for each of the following IBM server groups:

- ▶ Core warehouse servers that host the DB2 database partitions on which the data resides
- ▶ The warehouse applications modules that host the IBM InfoSphere Warehouse application
- ▶ Business intelligence (BI) module that host the IBM Cognos components

To learn more about managing high availability for the IBM Smart Analytics System, see this IBM training course:

https://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course_description&courseCode=DW330

3.2 IBM Tivoli System Automation for Multiplatforms on IBM Smart Analytics System

IBM Tivoli SA MP manages and provides automated recovery for the IBM Smart Analytics System components that require high availability. The infrastructure of Tivoli System Automation for Multiplatforms is based on the Reliable Scalable Cluster Technology (RSCT), which is an IBM software product that provides a highly available and scalable clustering environment for applications and businesses running on AIX and Linux platforms. Tivoli SA MP allows you to configure high availability systems through the use of policies that define the relationships among the various components. After the relationships are established, Tivoli SA MP assumes responsibility for managing the resources on the specified nodes as configured in the policies. When a resource failed, Tivoli System Automation for Multiplatforms can quickly and consistently perform a restart either on the same server or on the standby server.

The relationships among the resources managed by Tivoli SA MP are controlled cluster-wide. If one application needs to be moved from one server to other, Tivoli System Automation for Multiplatforms automatically handles the start and stop sequences, node requirements, dependencies, and any further follow-on actions. You can group the resources managed by Tivoli System Automation for Multiplatforms to establish the relationships among the members of the group as a location or start/stop relationship. When grouped, the operation against the resources can be referenced to the resource group as a single entry, and is applied to the entire group.

In a Tivoli SA MP configuration, a set of nodes in the system, commonly called a cluster, is referred as a *peer domain*. All nodes in a peer domain continually send and receive heartbeats over *communication groups*. A communication group is a set of nodes that can talk to each other over a common communication medium. An example of a communication group would be network interface cards residing on various nodes connected to the same network.

Tivoli SA MP terminology frequently seen in the IBM Smart Analytics System configuration:

- ▶ Peer domain: A peer domain or cluster is a group of host systems where the Tivoli System Automation for Multiplatforms managed resources reside. A peer domain can consist of one or more *systems* or *nodes*.
- ▶ Resource: A resource is any piece of hardware or software that can be defined for Tivoli System Automation for Multiplatforms to manage, for example, a network interface card or a DB2 database partition.
- ▶ Resource group: A resource group is a set or collection of resources.

- ▶ Relationships: It defines the relationships between the resources within a cluster. There are two types of relationships:
 - Start-stop relationship: This relationship defines the start and stop dependencies between resources.
 - Location relationship: This relationship is used when resources must, if possible, be started on the same or another node in the cluster.
- ▶ Equivalency: An equivalency is a set of resources that provide the same functionality. Tivoli System Automation for Multiplatforms can select any resource in the equivalency to provide an operation. On the IBM Smart Analytics System, the network adapters that require high availability have equivalencies, for example, the DB2 Fast Communications Manager network adapters.

For more information about Tivoli System Automation for Multiplatforms, see the IBM Tivoli System Automation for Multiplatforms manual, *Administrator's and User's Guide*, SC33-8415-01, at the following web address:

<http://publib.boulder.ibm.com/tividd/td/IBMTivoliSystemAutomationforMultiplatforms3.1.html>

3.3 High availability overview for the core warehouse servers

An active-passive configuration can have a standby node for one or more active nodes. Figure 3-1 shows a high availability group configuration with two active nodes.

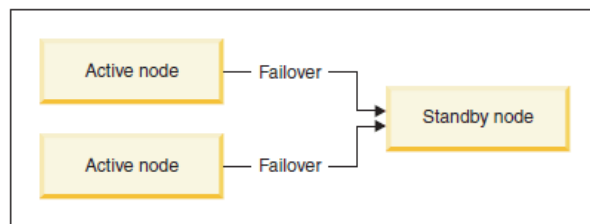


Figure 3-1 Two active nodes high availability group configuration

The IBM Smart Analytics System core warehouse servers have this type of active-passive high availability configuration. For a number of administration and data nodes, there is a standby node to receive the managed resources from a failed server.

The core warehouse can have one or more *high availability group*. Each high availability group has a number of administration nodes and data nodes, storage servers (IBM DS), SAN switches, and one standby node. When a server failure occurs, its storage and workload are moved to the standby node of the high availability group. Only one failure is supported and enforced per high availability group.

In a high availability group, if a node is failed over to the standby node, the standby node cannot fail over to another node within the same high availability group nor to the node on other high availability group.

The production instances or the core warehouse nodes of the IBM Smart Analytics System 5600 has the following high availability group configuration:

- ▶ Maximum of five active nodes (administration, user, or data)
- ▶ Ten storage server (either DS3400 or DS3524)
- ▶ One standby node
- ▶ Two SAN switches (redundant pair)

For the IBM Smart Analytics System 5600, one high availability group has one standby node for one to five administration, user, or data nodes. If a sixth node is needed, then a new high availability group is initiated, thus, requiring a second standby node. This new standby node will manage the next set of five nodes. That is, for the IBM Smart Analytics System 5600, each group of five nodes always has its own standby node.

Figure 3-2 illustrates the high availability group for the IBM Smart Analytics System 5600 V1.

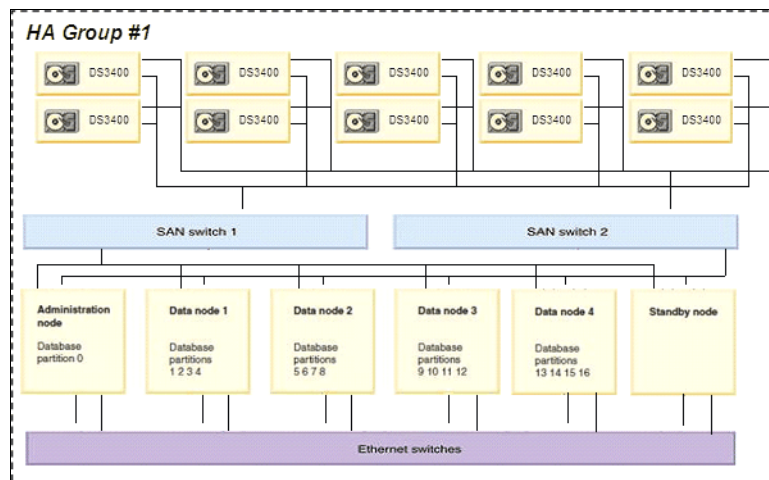


Figure 3-2 Example of a high availability group for IBM Smart Analytics System 5600

The core warehouse nodes of the IBM Smart Analytics System 7600 has the following high availability group configuration:

- ▶ Maximum of eight active nodes (administration, user, or data)
- ▶ Two DS5300 storage server
- ▶ One standby node
- ▶ Two SAN switches (redundant pair)

For the IBM Smart Analytics System 7600, each high availability group always has one standby node for one to eight administration, user, or data nodes. If a ninth node is needed, then a new high availability group is initiated, thus, requiring a second stand by node. This new stand by node will manage the next set of eight nodes. That is, for the IBM Smart Analytics System 7600, each group of eight nodes always has its own standby node.

Figure 3-3 illustrates the high availability group for the IBM Smart Analytics 7600.

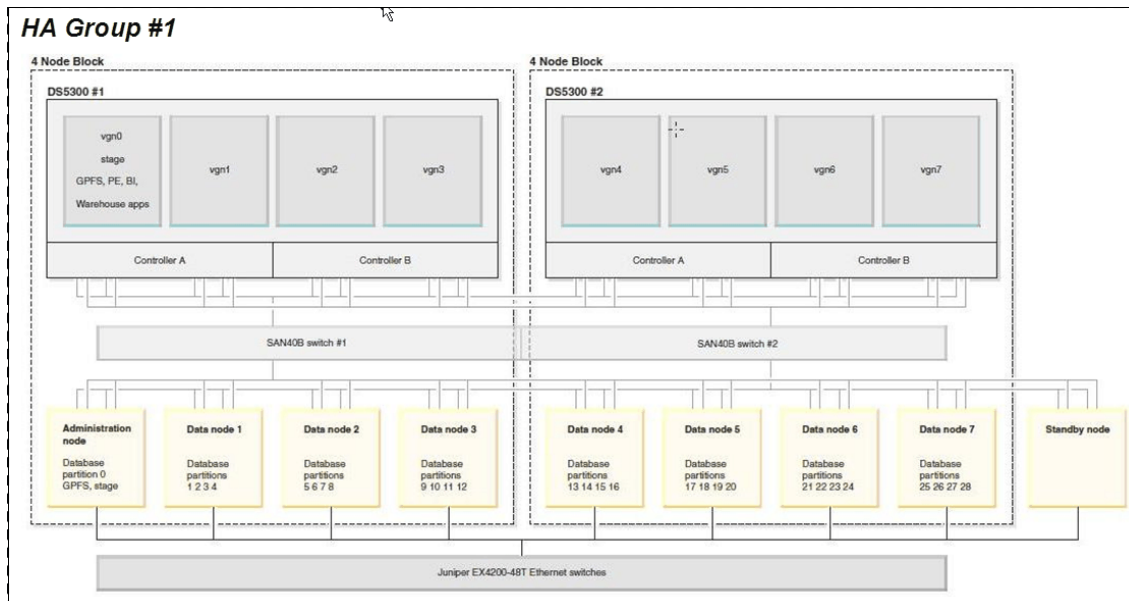


Figure 3-3 Example of a high availability group for IBM Smart Analytics System 7600

The core warehouse nodes of the IBM Smart Analytics System 7700 has the following high availability group configuration:

- ▶ A maximum of four active nodes (administration, user, or data)
- ▶ 13 DS3524 storage server for one administration node and three data nodes, or 16 DS3524 storage server for four data nodes
- ▶ One standby node
- ▶ Two SAN switches (redundant pair)

For the IBM Smart Analytics System 7700, each high availability group has one standby node for one to four administration, user, or data nodes. If a fifth node is needed, then a new high availability group is initiated, thus, requiring a second standby node. This new standby node manages the next set of four nodes. That is, for the IBM Smart Analytics System 7700, each group of four nodes always has its own standby node.

Figure 3-4 illustrates the high availability group for the IBM Smart Analytics 7700.

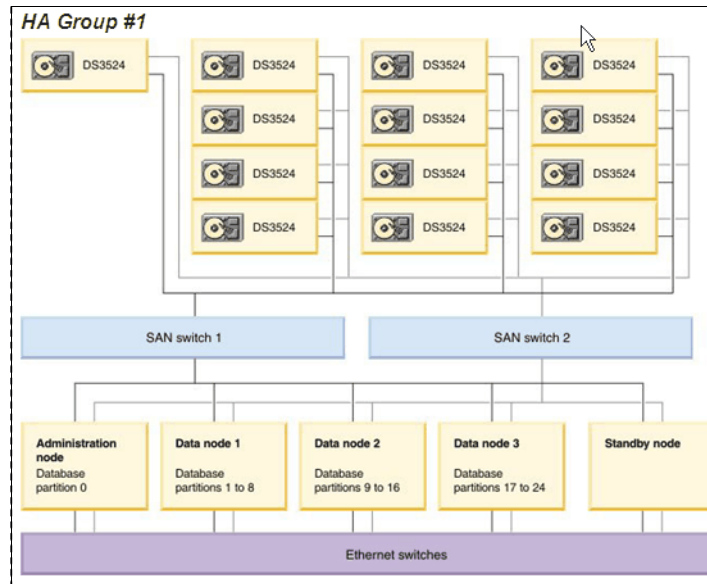


Figure 3-4 Example of a high availability group for IBM Smart Analytics System 7700

For a high availability (HA) group, when a failure is detected, all managed resources are automatically moved from the failing node to a standby node of the same high availability group by Tivoli System Automation for Multiplatforms.

The standby node has the same software stack and code level of the other nodes. The standby node is always powered up and ready to assume the resources from a failed node. One HA group is connected to a redundant pair of SAN switches so the standby node is able to access the storage for the remaining nodes. When a standby node takes the managed resources, it is able to access the storage resources and start the DB2 database partitions.

To connect to services in an IBM Smart Analytics System environment, clients of those services refer to and use specified host names or IP addresses to access the services. The IBM Smart Analytics System environment utilizes *Service IPs* that are always associated with those services regardless of which server hosts those services. The service IPs are managed as Tivoli SA MP resources.

To allow Tivoli SA MP to manage the resources, each resource in the IBM Smart Analytics System must be grouped to at least one resource group. All resources are organized into a hierarchy of resource groups. The top resource group in the hierarchy is at the node level. Below this level, resources are grouped based on their dependencies and requirements, as follows:

- ▶ The resources required by the DB2 database partition are grouped in a partition-level resource group.
- ▶ A volume group and its file systems is defined as a resource group.
- ▶ All resources hosted under the same node are also grouped as a resource group.

Typically, each DB2 database partition has a resource group associated with it, and each resource group contains all required resources:

- ▶ DB2 database partition resource: For DB2 instance
- ▶ File systems resources: For database directory, logs, and table space containers
- ▶ Service IPs: For virtual IPs moved from one primary node to the standby node

For example, for the IBM Smart Analytics System 7700 administration node, the node-level partition group has the DB2 coordinator partition and the volume group resource group of the file systems associated with the coordinator partition. For the first data node, the node-level resource group has the eight database partitions it hosts and the volume group resource group of the table space containers.

The Tivoli SA MP resources for the network interfaces, are automatically defined when the resource domain is created. On the IBM Smart Analytics System, the domain for the core warehouse is named *bcudomain* by default. After being defined, the resources are grouped in *equivalencies*. On top of the equivalencies, the dependency relationship is created between the DB2 database partitions and the high availability networks. The network resource is automatically started during the operating system boot.

In addition to the network interface resources, the DB2 instance home directory is also defined as an equivalency. For the IBM Smart Analytics System 7700 and 7600, this directory is a General Parallel File System (GPFS) and has a dependency relationship with the database partition resources. This dependency relationship prevents the resources from being started if the home directory is not available.

On the IBM Smart Analytics System 5600, there is a NFS services related resource group for the DB2 instance home directory. There are also dependency relationships defined between DB2 database partition resources and NFS server (DB2 instance home directory). These dependencies prevent Tivoli SA MP from starting a database partition if the NFS server is not available.

3.4 Managing high availability resources for the core warehouse

IBM Smart Analytics System provides the *high availability management toolkit* with a set of scripts for managing the high availability configuration. The scripts are located under the `/usr/SmartAnalytics/ha_tools` directory on the management node. The root user can run each script as a command from the management node or copy the script to other nodes and run from there. These scripts manage the servers for the core warehouse (administration, data, and user nodes) only.

The following scripts are used for managing the high availability configuration for the core warehouse:

► **ha1s:**

This command lists the DB2 database partitions resources and their high availability status.

Syntax: **ha1s**

► **hastartdb2:**

This command starts the DB2 database partition resources. To start the resources for a specific node, the node name list must be specified as the **nodelist** argument. If a node name is not specified, the command is applied to all resources. This command starts the DB2 instance service and also mounts the volume group resources.

Syntax: **hastartdb2 [nodelist]**

► **hastopdb2:**

This command stops the DB2 database partition resources. To stop the resources for a specific node, the node name list must be specified as **nodelist** argument. If a node name is not specified, the command is applied to all resources. This command stops the DB2 instance service and unmounts the volume group resources.

Syntax: **hastopdb2 [nodelist]**

► **hafailover:**

This command fails over the node to the standby manually. The command moves the DB2 database partition resources from the node specified on the command argument to the standby node.

Syntax: **hafailover** *nodename*

► **hafailback:**

This command fails back the node manually. It moves back the DB2 database partition resources from the standby node to the primary node specified on the command argument.

Syntax: **hafailback** *nodename*

► **hareset:**

This command attempts to reset the resources with the Pending, Failed, or Stuck nominal state. This command stops the high availability scripts to prevent the DB2 instance from being started. Next, the resources are reset. If the optional argument is not specified, the nominal state for all resources is changed to Offline. If the argument is specified, the command attempts to change the nominal state only for the ones that are in Pending, Failed or Stuck state.

Syntax: **hareset** [*nooffline*]

On the IBM Smart Analytics System 5600, additional scripts are provided for managing the NFS server high availability resource:

► **hastartnfs:**

This command starts the NFS services and attempts to mount the /db2home file system for all nodes. If the argument nomount is specified, it starts the NFS services only but does not mount the /db2home file system. Always start the NFS services before you start the DB2 database partition resources.

Syntax: **hastartnfs** [*nomount*]

► **hastopnfs:**

This command unmounts the /db2home file system on all nodes. If the argument force is specified, it stops the DB2 database partitions resources that are still running, then unmounts the /db2home file system.

Syntax: **hastopnfs** [*force*]

3.4.1 Monitoring the high availability resources for the core warehouse

To monitor the resources and nodes for the core warehouse, you can use either the scripts in the high availability management toolkit (**ha1s** and **hacknode**) or Tivoli SA MP commands.

Monitoring the high availability using the toolkit

You can use the **ha1s** command to check the resource status on an IBM Smart Analytics System environment. These scripts are available only for the core warehouse servers.

The **ha1s** command returns a summary of the resources and its nominal state for the entire cluster.

Example 3-1 shows output from a **ha1s** command run on an IBM Smart Analytics System 7600. It has one administration node, one user node, and two data nodes.

Example 3-1 IBM Smart Analytics System 7600 ha1s output

PARTITIONS	PRIMARY	SECONDARY	CURRENT LOCATION	RESOURCE OPSTATE	HA STATUS
1,2,3,4	dataNode01	standbyNode	dataNode01	Online	Normal
5,6,7,8	dataNode02	standbyNode	dataNode02	Online	Normal
0	adminNode	standbyNode	adminNode	Online	Normal
990	userNode	standbyNode	userNode	Online	Normal

The following fields are shown in this **ha1s** command output:

- ▶ **Partitions:** Shows a list of DB2 database partitions for the node in the node level group.
- ▶ **Primary:** Shows the primary node for the DB2 database partition level resources.
- ▶ **Secondary:** Shows the secondary node or standby for the DB2 database partition level resources.
- ▶ **Current location:** Shows the current node location for the DB2 database partition level resources where the resources are running at the moment.
- ▶ **Resource opstate:** Shows the nominal state for the node level resource group. The following nominal states are possible:
 - **Online:** This state is shown when all resources at the node level resource group are online.
 - **Offline:** This state is shown when all resources at the node level resource group are offline.

- Pending: When the resources are in a pending state, it can appear either as “Pending Offline” or “Pending Online”.
- ▶ HA status: Shows the current status for the node. If the resources are running on its primary location, the status will be “Normal”. If the resources are running on the standby node, the status will appear as “Failover”. It can also show “Pending” if the resources are on either “Pending Online” or “Pending Offline” state. Another status is “Stuck” if the resources get into a *stuck* operational state.

In this example all nodes are running on their primary location and also under Normal operational state.

Monitoring the high availability using Tivoli SA MP

The Tivoli SA MP native commands are tools for monitoring the high availability of the IBM Smart Analytics System environments.

To obtain a more complete output, run the commands using the root user. Using the DB2 instance owner user to run the Tivoli SA MP commands will not show the locked state of a resource.

The command used for monitoring the resources status is `Issam`. It shows all the resources, resource groups, the current state of the resource or group (OpState), the desired state (Nominal State), and other useful information.

Example 3-2 shows the output format of the `Issam` command.

Example 3-2 Example of an Issam command output format

```
<OpState> IBM.ResourceGroup:<ResourceGroup> Nominal=<NominalState>
| - <Opstate> <ResourceClass>:<Resource>
| - <OpState> <ResourceClass>:<Resource>:<NodeName>
| - <OpState> <ResourceClass>:<Resource>:<NodeName>
```

Here, the various parameters have the following meanings:

- ▶ **<OpState>**: The current operational state of the resource or resource group. For example,
- ▶ **<ResourceGroup>**: The resource group name.
- ▶ **<NominalState>**: The nominal state for the resource group.
- ▶ **<ResourceClass>**: The resource type which can be, for example, IBM.Application or IBM.ServiceIP on the IBM Smart Analytics System environments.
- ▶ **<Resource>**: The resource name.
- ▶ **<NodeName>**: The host name for the node holding the resources.

Example 3-3 illustrates an output excerpt of `Issam` showing the administration node information of an IBM Smart Analytics System 7700.

Example 3-3 Issam output for an administration node

```

Online IBM.ResourceGroup:db2_bcuai_x_server_AdminNode-rg Nominal=Online
|- Online IBM.ResourceGroup:db2_bcuai_x_0-rg Nominal=Online
   |- Online IBM.Application:db2_bcuai_x_0-rs
      |- Offline IBM.Application:db2_bcuai_x_0-rs:AdminNode
      |- Online IBM.Application:db2_bcuai_x_0-rs:StandbyNode1
   |- Online IBM.Application:db2mnt-db2fs_bcuai_x_NODE0000-rs
      |- Offline IBM.Application:db2mnt-db2fs_bcuai_x_NODE0000-rs:AdminNode
      |- Online IBM.Application:db2mnt-db2fs_bcuai_x_NODE0000-rs:StandbyNode1
   |- Online IBM.Application:db2mnt-db2mlog_bcuai_x_NODE0000-rs
      |- Offline IBM.Application:db2mnt-db2mlog_bcuai_x_NODE0000-rs:AdminNode
      |- Online IBM.Application:db2mnt-db2mlog_bcuai_x_NODE0000-rs:StandbyNode1
   |- Online IBM.Application:db2mnt-db2path_bcuai_x_NODE0000-rs
      |- Offline IBM.Application:db2mnt-db2path_bcuai_x_NODE0000-rs:AdminNode
      |- Online IBM.Application:db2mnt-db2path_bcuai_x_NODE0000-rs:StandbyNode1
   |- Online IBM.Application:db2mnt-db2plog_bcuai_x_NODE0000-rs
      |- Offline IBM.Application:db2mnt-db2plog_bcuai_x_NODE0000-rs:AdminNode
      |- Online IBM.Application:db2mnt-db2plog_bcuai_x_NODE0000-rs:StandbyNode1
   '- Online IBM.ServiceIP:db2ip_172_23_1_111-rs
      |- Offline IBM.ServiceIP:db2ip_172_23_1_111-rs:AdminNode
      |- Online IBM.ServiceIP:db2ip_172_23_1_111-rs:StandbyNode1
   '- Online IBM.ResourceGroup:db2_bcuai_x_AdminNode_vg-rg Nominal=Online
      '- Online IBM.Application:db2_bcuai_x_vgp0
         |- Offline IBM.Application:db2_bcuai_x_vgp0:AdminNode
         |- Online IBM.Application:db2_bcuai_x_vgp0:StandbyNode1

```

The `Issam` command output organizes the resources and group information by indentation. In the foregoing example, we have the following resources:

- ▶ The first line (the most left) is the node level resource group, `db2_bcuai_x_server_AdminNode-rg`. The naming convention includes DB2 instance name and node host name.
- ▶ At the second level is the partition level resource group, `db2_bcuai_x_0-rg`. The naming convention includes DB2 instance name and database partition number.
- ▶ The third level is the resources. The resources are listed by resource types (Application, ServiceIP). The naming convention includes the node that holds the resource, for example, `db2_bcuai_x_0-rs` and `db2ip_172_23_1_111-rs`.

3.4.2 Starting and stopping resources with the high availability management toolkit

The preferred method to start and stop high availability resources for the core warehouse is by using the high availability management toolkit scripts:

- ▶ **hastartdb2**: Use to start the DB2 resources managed by Tivoli SA MP:
You have the option to specify a node list to startup resources in multiple nodes

- ▶ **hastopdb2**: Use to stop the DB2 resources managed by Tivoli SA MP:

This command stops the DB2 database partition level resources and the volume group resource group. You have the option to specify a node list to stop resources in multiple nodes.

On the IBM Smart Analytics System 5600, you can start and stop the NFS server using the **hastartnfs** and **hastopnfs** commands. For more references about the high availability management toolkit commands see 3.4, “Managing high availability resources for the core warehouse” on page 39.

The commands must be run as the root user. The commands are asynchronous. To check the progress of the command running and the resources state use the **hal**s or **lssam** commands.

Before starting the high availability resources, check if all equivalencies and resources are available using the **lssam** command. Example 3-4 shows an output excerpt with the equivalency status from an IBM Smart Analytics System 7600.

Example 3-4 Issam ouptup showing the equivalencies status

```
Online IBM.Equivalency:db2_db2home_gpfs_DataNode01-StandbyNode1-equ
  |- Online IBM.AgFileSystem:db2homefs_StandbyNode1:StandbyNode1
  '- Online IBM.AgFileSystem:db2homefs_DataNode01:DataNode01
Online IBM.Equivalency:db2_db2home_gpfs_DataNode02-StandbyNode1-equ
  |- Online IBM.AgFileSystem:db2homefs_StandbyNode1:StandbyNode1
  '- Online IBM.AgFileSystem:db2homefs_DataNode02:DataNode02
Online IBM.Equivalency:db2_db2home_gpfs_AdminNode-StandbyNode1-equ
  |- Online IBM.AgFileSystem:db2homefs_AdminNode:AdminNode
  '- Online IBM.AgFileSystem:db2homefs_StandbyNode1:StandbyNode1
Online IBM.Equivalency:db2_db2home_gpfs_UserNode-StandbyNode1-equ
  |- Online IBM.AgFileSystem:db2homefs_StandbyNode1:StandbyNode1
  '- Online IBM.AgFileSystem:db2homefs_UserNode:UserNode
Online IBM.Equivalency:db2_private_network
  |- Online IBM.NetworkInterface:en11:AdminNode
  |- Online IBM.NetworkInterface:en11:DataNode01
  |- Online IBM.NetworkInterface:en11:DataNode02
  |- Online IBM.NetworkInterface:en11:StandbyNode1
  '- Online IBM.NetworkInterface:en11:UserNode
```

This output shows the equivalencies of FCM network and /db2home file systems are online on all nodes. If any of them are Offline, troubleshoot the cause of the resource offline first to prevent an unintentional failover action initiated by Tivoli SA MP. For example, if a network cable is malfunctioning or unplugged, it will prevent the node level resource group from being started and cause a failover to the standby node where the resource (equivalency) is online. To avoid this situation, check the equivalency status before attempting to start the resource.

The /db2home file system is deployed as a GPFS on IBM Smart Analytics System 7600 and 7700. It is managed by the operating system and is mounted and shared across all nodes in the cluster automatically. You can mount and unmount a GPFS file system using the following commands:

- ▶ **mmumount a11 -a**: Use to unmount the GPFS file systems on all nodes.
- ▶ **mmmumount a11 -a**: Use to mount the GPFS file systems on all nodes.

For further GPFS information, see the website:

<http://www-03.ibm.com/systems/clusters/software/gpfs/resources.html>

Also check the node status before attempting to start the node level resource. To check the node status, use the Tivoli SA MP command **lsrpnode**.

Example 3-5 shows an **lsrpnode** command output from an IBM Smart Analytics System 7600.

Example 3-5 lsrpnode output

Name	OpState	RSCTVersion	NodeNum	NodeID
standbyNode	Online	2.5.3.0	5	6ab290305fef0199
adminNode	Online	2.5.3.0	1	ee2fd0af24445c0a
dataNode02	Online	2.5.3.0	3	c7aac9e92d6291ca
dataNode01	Online	2.5.3.0	4	f4601f8a0cce90a
userNode	Online	2.5.3.0	2	77d623b025a00520

The database administrator still can use **db2start** and **db2stop** to start and stop a DB2 instance. When the DB2 instance owner user issues **db2stop**, the DB2 instance service for each database partition is stopped. The cluster management software will not attempt to bring the instance online. The **lssam** output will show the DB2 database partition resource is in the Offline and Suspended Propagated state. Tivoli SA MP interprets that a database administrator is doing a maintenance activity and suspends the high availability monitoring over the resource.

Important: Do not attempt to start or stop database partition level resources when the DB2 database partition resources are in “Suspended Propagated” mode. Tivoli SA MP will prevent the resources from starting or stopping because it interprets that a maintenance task is in place.

3.4.3 Starting and stopping resources with Tivoli SA MP commands

Though the native Tivoli SA MP commands can be used to manage the core warehouse resources on the IBM Smart Analytics System environments, the preferred method is using the high availability management toolkit scripts.

This method is used because the high availability toolkit scripts automatically check to see if the required dependencies are available and the command syntax is simplified.

Resources cannot be started and stopped individually in an HA cluster. Instead, you can control resources by changing the nominal state of the resource groups that contain them.

As a root user, use the command `chrg` to change the nominal state for a resource group. To change a nominal state for a node level resource group, you must change the DB2 database partition level resource group first, then change the node level resource group. Use the following commands to change a nominal state for a resource group:

```
chrg -o <NominalState> <ResourceGroup>
```

Here, the parameters have the following meanings:

- ▶ *<NominalState>* is the state defined for the resource. The value is either Offline or Online.
- ▶ *<ResourceGroup>* is the resource group name.

Use this order to change the resources for an administration node to Offline:

```
chrg -o Offline db2_bcuaix_0-rg  
chrg -o Offline db2_bcuaix_server_AdminNode-rg  
chrg -o Offline db2_bcuaix_AdminNode_vg-rg
```

Bring the DB2 database partition level resource group offline first, then the node level resource group, and the volume group level resource group last.

To monitor the resource status, use the `lssam` command. During the changing state of a resource, there is “Pending Offline” state if the status changing is from Online to Offline, and “Pending Online” if the change is from Offline to Online.

3.4.4 Manual node failover for maintenance

Moving resources from the primary node to the standby node reduces the service interruption required for system maintenance activities. After completing the maintenance tasks, the resources have to be moved back to the primary node. The manual failover for system maintenance must be performed on a maintenance window.

The high availability toolkit scripts do not check for inflight transactions nor warn against them. Always verify that the system is quiesced properly and that there are no in flight transactions when perform manual node failover to avoid a service interruption and a service rollback for any inflight transactions.

To move the resources for the core warehouse server, use the high availability management toolkit script **hafailover**. This command moves the resources from the specified server to the standby node of its high availability group.

Root user: All high availability management toolkit commands must be performed as the root user.

For example, to move the resources from the administration node to the standby node, issue this command:

```
hafailover <adminNode>
```

In this command, *<adminNode>* is the administration node host name in the cluster. The command stops all Tivoli SA MP managed resources groups on the administration node, then starts the resources on the standby node. The Tivoli SA MP managed resources are DB2 database partition level resource, volume group resource, and Service IP resource.

You can use either **lssam** or **hals** to monitor the resource status and the failover progress.

Example 3-6 shows the **hals** output after the **hafailover** command was run and the resources are moved to the standby node.

Example 3-6 lssam output in failover state

PARTITIONS	PRIMARY	SECONDARY	CURRENT LOCATION	RESOURCE OPSTATE	HA STATUS
1,2,3,4	dataNode01	standbyNode	dataNode01	Online	Normal
5,6,7,8	dataNode02	standbyNode	dataNode02	Online	Normal
0	adminNode	standbyNode	standbyNode	Online	Failover
990	userNode	standbyNode	userNode	Online	Normal

When the node requires long maintenance hours, after failing the resources over to the standby node, you can place the node in an ineligible list by using the **samctr1** Tivoli SA MP command as the root user:

```
samctr1 -u a <adminNode>
```

Here, the *<adminNode>* is the host name of the node to be placed in the ineligible list in the cluster domain.

Then you can unmount all shared file systems. for the IBM Smart Analytics System 7600 and 7700 where the file systems are managed by GPFS, use the commands:

```
mmumount /db2home
mmumount /home
```

For the IBM Smart Analytics System 5600, Tivoli SA MP manages the NFS server and will unmount the shared home directories automatically.

Nodes: Removing a node from the cluster domain is an *optional* step when doing a manual failover.

3.4.5 Manual node failback

When the maintenance tasks complete, fail the resources back to the primary nodes from the standby node or reintegrate the cluster domain if the node was placed in the ineligible list. Just as with the failover process, the failback must be performed during a planned maintenance window.

Check if all equivalencies and the node to be failed back are online before start the failback process. Check also if the automation is active at the cluster using the following command:

```
lssamctrl
```

Example 3-7 shows that the automation is active.

Example 3-7 Tivoli SA MP control output

```
# lssamctrl
Displaying SAM Control information:

SAMControl:
  Timeout                = 60
  RetryCount             = 3
  Automation           = Auto
  ExcludedNodes          = {}
  ResourceRestartTimeout = 5
  ActiveVersion           = [3.2.0.0,Mon Oct 11 04:23:55 CDT 2010]
  EnablePublisher        = Disabled
  TraceLevel             = 31
  ActivePolicy            = []
  CleanupList            = {}
  PublisherList          = {}
```

If automation is inactive, use this command to activate it:

```
samctrl -M F
```

Important: Use the capital “M” option to take the cluster out of manual mode. (The small caps “m” means “migrate”.)

If the node was placed in the ineligible list during the failover, it must be removed from the list before starting the resources. To take the node from the ineligible list, run the following command:

```
samctrl -u d <adminNode>
```

Here, <adminNode> is the host name for the node that is being taken from the ineligible list in the domain.

After the node is back in the domain, check the node status using **lsrpdomain**, and run the **lssam** command to check if the equivalencies are online.

When all equivalencies and the node are online, you can perform failback using the high availability management toolkit script **hafailback**. For example, to fail back the administration node, the command is:

```
hafailback <adminNode>
```

Here, the parameter <adminNode> is the host name of the administration node to be failed back to the primary position.

Alternatively, you can use the Tivoli SA MP commands to fail the resources back to the primary position. To move the resources back to the primary position, the nominal state of the DB2 database partition level resource group and the node level resource group must be changed to offline. After the resources groups are brought to the online nominal state again, the resources are started at their preferred node position which, in this case, is the primary node.

See “Starting and stopping resources with Tivoli SA MP commands” on page 45 for information about how to stop and start the resources using Tivoli SA MP commands.

Sometimes the resource nominal status for the equivalencies is something other than “Online”. It can be, for example, Stuck Offline or Failed Offline. If this is the case for the /db2home file system, try to restart the NFS server for the IBM Smart Analytics System 5600 using **hastopnfs** and **hastartnfs**. For the IBM Smart Analytics System 7600 and 7700, because /db2home is GPFS, unmount and mount the file system on that node using these commands:

```
mmumount /db2home  
mmmount /db2home
```

If the nominal states for the network equivalencies are showing as Failed or Stuck, reset the resource using the **hareset** command from the high availability management toolkit:

```
hareset nooffline
```

If resetting the resource cannot bring the resource online, you can restart the entire node and then start the resources within the domain by performing the following steps:

1. Stop all high availability managed resources using **hastopdb2**. This high availability management toolkit command is the preferred method for this task.
2. Unmount the shared file system:
 - Use **hastopnfs** for the IBM Smart Analytics 5600.
 - Issue **mmount a11 -a** for the IBM Smart Analytics System 7600 and 7700.
3. Bring the domain to the offline state using **stoprpdomain bcudomain**.
Bring the domain offline after all resources are stopped and the shared file systems are unmounted. The IBM Smart Analytics System high availability configuration always uses **bcudomain** as the default domain name.
4. Start the domain using **starttrpdomain bcudomain**.
5. Check if the domain is online using the **1srpdomain** and **1srpnode**.
6. Mount the shared file systems:
After the domain is online, mount the shared file systems with the commands:
 - **hastartnfs** for the IBM Smart Analytics System 5600
 - **mmount a11 -a** for the IBM Smart Analytics System 7600 and 7700
7. Check to see if all equivalencies are online using **1ssam**.
8. Start the resources using **hastartdb2**.
9. The resources must be started at their primary location. Check the status using **1ssam**.

If for any reason the resources are not brought online at their primary location or are failing, more troubleshooting must be done. You can find the Tivoli SA MP log on the operating system syslog. Examine the syslog file for the node that has the problem.

To check to see if a resource can be started on the failing node, start the resource manually. To do that, you must place the Tivoli SA MP in the manual mode using the following command:

```
samctr1 -M T
```

Then start the resource that failed to start. For example, if the DB2 database partition resource failed to start, then start the DB2 database partition using the command:

```
db2start nodenum N
```


Or, if a file system resource does not come online, manually mount the file system on the node. If the problem is resolved, place Tivoli SA MP back to automatic mode using the command:

```
samctrl -M F
```

If the problem persists, contact the IBM Smart Analytics System Support with detailed information about the situation.

For further information about high availability management and configuration for the core warehouse servers, see the *IBM Smart Analytics System User's Guide* for your respective version.

3.5 High availability for the warehouse application module

An IBM Smart Analytics System warehouse applications module can consist of either one or two nodes:

► Warehouse application node:

A required node that hosts all the IBM InfoSphere Warehouse application components:

- InfoSphere Warehouse Administration Console
- InfoSphere Warehouse SQL Warehousing Tool
- IBM Alphablox
- InfoSphere Warehouse Miningblox
- Cubing Services component
- WebSphere Application Server

For IBM Smart Analytics System offerings running DB2 for Linux, UNIX, and Windows Version 9.7, the application server metadata is hosted in a DB2 database (iswmeta). For IBM Smart Analytics System offerings running DB2 for Linux, UNIX, and Windows Version 9.5, the metadata is hosted in a table space (DWEDEFAULTCONTROL) at the warehouse production instance.

► OLAP node:

An optional node that hosts the Cubing Services components. The OLAP node is added to scale out the cube servers. The cube server runs on its own Java™ virtual machine (JVM) space and does not require hosting on a WebSphere Application Server.

For most IBM Smart Analytics System offerings, the optional high availability configuration for the warehouse applications module uses an active-active failover configuration.

Figure 3-5 shows a diagram of an active/active configuration. Note that the 5600 V1 offering uses active-passive failover for the warehouse applications module. This chapter focuses on active-active failover, which is used in all other IBM Smart Analytics System offerings.

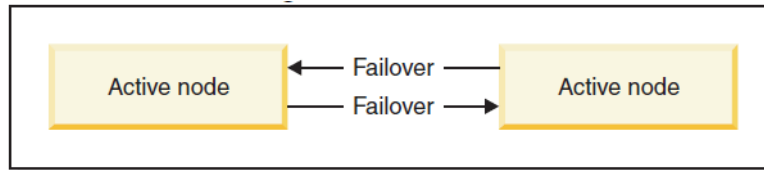


Figure 3-5 Mutual failover high availability configuration

High availability can be implemented for a warehouse applications module only when the module includes two nodes (the warehouse applications node and the OLAP node).

Each application server has its file system defined on external storage, /usr/IBM/dwe/appserver_001 for the warehouse application node and /usr/IBM/dwe/appserver_002 for the OLAP node. Both nodes in the application server high availability cluster have access to these file systems. On offerings based on DB2 9.7, there is an additional file system named /iswhome on external storage. This file system is assigned to the warehouse applications node.

Tivoli SA MP manages the application servers to provide high availability. The default domain name is DB2WHSE_HA. Example 3-8 shows an `lsrpdomain` command output from the warehouse application server.

Example 3-8 Application servers high availability cluster domain

```
lsrpdomain
Name      OpState  RSCTActiveVersion  MixedVersions  TSPort  GSPort
DB2WHSE_HA  Online   2.5.3.0             No              12347   12348
```

This domain is independent from the core warehouse high availability cluster domain. The DB2WHSE_HA has its own set of resources and resource groups. The resources are grouped as follows:

► Warehouse application node:

This resource group includes the following resources:

- Application resources: Includes the InfoSphere Warehouse Administration Console, Alphablox platform, and the appserver_001 file system.
- Metadata resources: Includes the IBM InfoSphere Warehouse metadata database resources.
- Service IPs: Includes IP addresses for accessing warehouse application resources.

In most IBM Smart Analytics System offerings, the warehouse application node manages another service IP dedicated for accessing the metadata resources.

- User-created Alphablox applications and cube servers (when there is no OLAP node).

► OLAP node:

This resource group includes the following resources:

- OLAP resources: Includes the Cubing Services cube server resources and appserver_002 file system.
- Service IPs: Includes IP addresses for accessing OLAP node.
- User-created cube server resources: When a new cube server is created by the user, it must be defined as a resource group to Tivoli SA MP so the new cube server can be highly available. In addition to the Tivoli SA MP, you can manage the user-created cube servers with the IBM InfoSphere Warehouse Administration Console (an option).

Figure 3-6 shows a high availability configuration for the warehouse application servers.

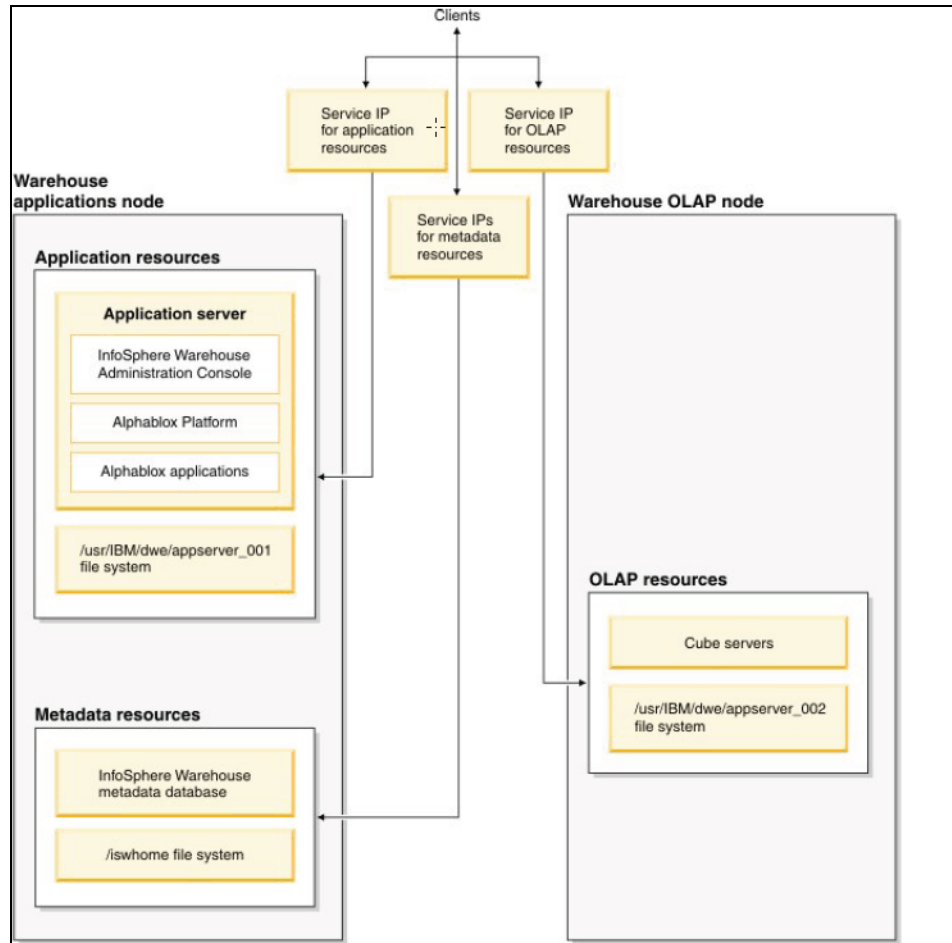


Figure 3-6 Warehouse applications modules high availability configuration

The high availability management for the warehouse applications and OLAP nodes is performed using the Tivoli SA MP commands only. There are no high availability management toolkit scripts available.

Example 3-9 shows an `lssam` output taken from an IBM Smart Analytics System 7600 with one warehouse application node (whsrv) and one OLAP node (olapnode). You can see the location where each resource is running. Both DB2 dweadmin instance service and the /iswhome file system are running on the warehouse application server (whsrv).

Example 3-9 IBM Smart Analytics System 7600 lssam output

```
Online IBM.ResourceGroup:db2_dweadmin_0-rg Nominal=Online
|- Online IBM.Application:db2_dweadmin_0-rs
  |- Online IBM.Application:db2_dweadmin_0-rs:whsrv
  '- Offline IBM.Application:db2_dweadmin_0-rs:olapnode
'- Online IBM.Application:db2mnt-iswhome-rs
  |- Online IBM.Application:db2mnt-iswhome-rs:whsrv
  '- Offline IBM.Application:db2mnt-iswhome-rs:olapnode
Online IBM.ResourceGroup:db2whse_ha_whsrv_type1.AppserverFilesystemsServiceIps.rg Nominal=Online
|- Online IBM.Application:db2whse_ha_whsrv_type1.appserver_001_filesystem
  |- Online IBM.Application:db2whse_ha_whsrv_type1.appserver_001_filesystem:whsrv
  '- Offline IBM.Application:db2whse_ha_whsrv_type1.appserver_001_filesystem:olapnode
|- Online IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_64_126
  |- Online IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_64_126:whsrv
  '- Offline IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_64_126:olapnode
'- Online IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_65_126
  |- Online IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_65_126:whsrv
  '- Offline IBM.ServiceIP:db2whse_ha_whsrv_type1.10_199_65_126:olapnode
Online IBM.ResourceGroup:db2whse_ha_whsrv_type1.abxplatform.rg Nominal=Online
'- Online IBM.Application:db2whse_ha_whsrv_type1.abxplatform
  |- Online IBM.Application:db2whse_ha_whsrv_type1.abxplatform:whsrv
  '- Offline IBM.Application:db2whse_ha_whsrv_type1.abxplatform:olapnode
Online IBM.ResourceGroup:db2whse_ha_whsrv_type1.adminconsole.rg Nominal=Online
'- Online IBM.Application:db2whse_ha_whsrv_type1.adminconsole
  |- Online IBM.Application:db2whse_ha_whsrv_type1.adminconsole:whsrv
  '- Offline IBM.Application:db2whse_ha_whsrv_type1.adminconsole:olapnode
Online IBM.ResourceGroup:db2whse_ha_whsrv_type1.appserver.rg Nominal=Online
'- Online IBM.Application:db2whse_ha_whsrv_type1.appserver
  |- Online IBM.Application:db2whse_ha_whsrv_type1.appserver:whsrv
  '- Offline IBM.Application:db2whse_ha_whsrv_type1.appserver:olapnode
Online IBM.ResourceGroup:db2whse_ha_olapnode_type2.AppserverFilesystemsServiceIps.rg Nominal=Online
|- Online IBM.Application:db2whse_ha_olapnode_type2.appserver_002_filesystem
  |- Offline IBM.Application:db2whse_ha_olapnode_type2.appserver_002_filesystem:whsrv
  '- Online IBM.Application:db2whse_ha_olapnode_type2.appserver_002_filesystem:olapnode
|- Online IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_64_127
  |- Offline IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_64_127:whsrv
  '- Online IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_64_127:olapnode
'- Online IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_65_127
  |- Offline IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_65_127:whsrv
  '- Online IBM.ServiceIP:db2whse_ha_olapnode_type2.10_199_65_127:olapnode
Online IBM.Equivalency:db2_dweadmin_0-rg_group-equ
|- Online IBM.PeerNode:whsrv:whsrv
  '- Online IBM.PeerNode:olapnode:olapnode
Online IBM.Equivalency:db2whse_ha_whsrv_type1_networkadapter_equ
|- Online IBM.NetworkInterface:en12:olapnode
  '- Online IBM.NetworkInterface:en12:whsrv
Online IBM.Equivalency:db2whse_ha_whsrv_type1_networkadapter_equ_en13
|- Online IBM.NetworkInterface:en13:olapnode
  '- Online IBM.NetworkInterface:en13:whsrv
Online IBM.Equivalency:db2whse_ha_whsrv_type1_nodes_equiv
|- Online IBM.PeerNode:whsrv:whsrv
  '- Online IBM.PeerNode:olapnode:olapnode
Online IBM.Equivalency:db2whse_ha_olapnode_type2_networkadapter_equ
|- Online IBM.NetworkInterface:en12:olapnode
  '- Online IBM.NetworkInterface:en12:whsrv
Online IBM.Equivalency:db2whse_ha_olapnode_type2_networkadapter_equ_en13
|- Online IBM.NetworkInterface:en13:olapnode
  '- Online IBM.NetworkInterface:en13:whsrv
Online IBM.Equivalency:db2whse_ha_olapnode_type2_nodes_equiv
|- Online IBM.PeerNode:olapnode:olapnode
  '- Online IBM.PeerNode:whsrv:whsrv
```

3.5.1 Starting and stopping high availability resources for warehouse application servers

For the IBM Smart Analytics System 5600 V2 and 7700, the warehouse applications module is always managed using Tivoli SA MP commands, whether it contains one node or two nodes. Use Tivoli SA MP commands to start and stop the resources.

With the 5600 V1 and 7600, the nodes are set up to be managed using Tivoli SA MP only when the configuration has two nodes. Use Tivoli SA MP commands to start and stop the resources. If the configuration has a single node, you must use the regular InfoSphere Warehouse methods to start and stop components. For example, to start the InfoSphere Warehouse metadata database, issue **db2start** using the DB2 instance owner (dweadmin).

To check to see if there is a Tivoli SA MP cluster domain, use **lsrpdomain**. If the output shows that the DB2WHSE_HA domain exists, always use Tivoli SA MP **chrg** command to start and stop the resources.

Example 3-10 shows how to start an InfoSphere Warehouse metadata database resource.

Example 3-10 Changing resource group nominal state to Online

```
chrg -o Online -s "Name like 'db2_%-rg'"
```

There are dependencies defined on the warehouse applications nodes. For example, the IBM InfoSphere Warehouse metadata resource must always be started before the WebSphere Application Server Resource.

3.5.2 Manual failover warehouse application node

The following active resources are running on the warehouse application node:

- ▶ IBM InfoSphere Warehouse metadata database resources
- ▶ Service IPs and file system appserver_001 resources
- ▶ WebSphere Application Server resource
- ▶ Alphablox platform resources
- ▶ IBM InfoSphere Warehouse Administration Console resource

The following active resources are running on a OLAP node:

- ▶ Service IP
- ▶ File system appserver_002
- ▶ OLAP resources

Before failing over resources, ensure that all activity is quiesced on the Warehouse applications module. One technique to move resources from one node to another, is to add that node to the ineligible node list using the Tivoli SA MP command **samctrl**:

```
samctrl -u a <node name>
```

Here, <node name> is the host name of the warehouse application node or OLAP node.

This command brings down all resources which are currently running. When all resources are stopped, Tivoli SA MP attempts to start those resources on the next available node. You can verify the status and location of the resources using the **lssam** command.

You can use this method to move the resources running on the warehouse application node to the OLAP node or vice versa to perform maintenance activities.

3.5.3 Manual failback of the warehouse application node

When maintenance is concluded, you can move the failed-over resources back to their designated primary node. When the warehouse application resources have failed over to the OLAP node, they can be failed back to the primary location without impacting the running activities on the OLAP node. That is the case if the cube server are not running on the IBM InfoSphere Warehouse Administration Console. When the resources from the OLAP node have failed over the warehouse application node, the resources can be failed back to the OLAP node without stopping the running activities on the warehouse application node.

Use the **lsrpnod** command to check the node status. Use the **lssamctrl** command to check if the node is in the excluded list, and then use the **lssam** command to check the equivalencies status before restarting the resources.

To fail back, first check if the node is in the excluded list using **lssamctrl**. Example 3-11 shows that when a warehouse application node resources were failed over to the OLAP node manually, the node is included in an list for the non-available nodes.

Example 3-11 The warehouse application node in the excluded node list

```
# lssamctrl
Displaying SAM Control information:

SAMControl:
    Timeout           = 60
    RetryCount        = 3
```

```
Automation           = Auto
ExcludedNodes       = {WarehouseAppNode}
ResourceRestartTimeOut = 5
ActiveVersion         = [3.1.0.7,Wed Oct 27 10:44:27 EDT 2010]
EnablePublisher       = Disabled
TraceLevel           = 31
ActivePolicy          = []
CleanupList           = {}
PublisherList         = {}
```

Set the node to online in the domain and remove it from the ineligible category using the following command:

```
samctr1 -u d <nodename>
```

The node is eligible to accept resources again.

To move resources back to the designated primary node, if the servers are set up as active/active configuration, use the **chrg** commands to stop all resource groups associated with the node which has resources that are failed over. Do not use the **samctr1** command because this will result in all resources moving, not just failed-over resources.

To move the resources from the OLAP node back to the warehouse application node, stop the resources with the Warehouse application node as the primary server on the OLAP node using the Tivoli SA MP command **chrg** in the following sequence:

1. Stop the resources for the Alphablox platform (including the user-created applications), and InfoSphere Warehouse Administration Console.
2. Stop the WebSphere Application server resource.
3. Stop the application server file system (/appserver_001) and service IPs resources.
4. Stop the InfoSphere Warehouse metadata database resources.

Start the resources using the **chrg** command on the warehouse application node in the following order:

1. Start the application server file system (/appserver_001) and service IPs resources.
2. Start the InfoSphere Warehouse metadata database resources.
3. Start the WebSphere Application server resource.
4. Start the resources for the Alphablox platform (including the user-created applications), and InfoSphere Warehouse Administration Console.

In the case of failing back the resources to the OLAP node, after the OLAP node is brought online with the `samctr1 -u d <OLAP_Node>` command, stop the resources designated to the OLAP node from the warehouse node using the Tivoli SA MP command `chrg` in the following sequence:

1. Stop the cube server resources.
2. Stop the application server file system (/appserver_002) and service IPs resources.

After these resources are stopped, then start the resources from the OLAP node using the `chrg` command in the following order:

1. Start the application server file system (/appserver_002) and service IPs resources.
2. Start the cube server resources.

In this section, we provide the basic information about how to manage the high availability resources for the warehouse application nodes. For further details about the IBM Smart Analytics System warehouse application nodes, see the *IBM Smart Analytics System User's Guide* for your respective version.

For more information about IBM Tivoli System Automation, see this address:

http://www-947.ibm.com/support/entry/portal/Overview/Software/Tivoli/Tivoli_System_Automation_for_Multiplatforms

For more information about the IBM InfoSphere Warehouse Application component, see this address:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.dwe.navigate.doc/welcome_warehouse.html

3.6 High availability for the business intelligence module

The IBM Smart Analytics System business intelligence (BI) module hosts the IBM Cognos software providing the BI capabilities such as dashboarding, query reporting, and analysis. All nodes have the same software stack when the BI module is deployed. The following major software is installed on the BI nodes:

- ▶ IBM Cognos BI Server
- ▶ IBM Cognos Go! Dashboard
- ▶ IBM WebSphere Application Server
- ▶ IBM HTTP Server
- ▶ IBM DB2 Enterprise Server Edition
- ▶ IBM Tivoli System Automation

From the high availability configuration point of view, the most related Cognos components are as follows:

- ▶ Gateway: This component receives the user requests, validates and encrypts the password, and captures the required information for sending the request to the IBM Cognos Server. The gateway then passes the request to a dispatcher for later processing. High availability for this component is managed by Tivoli SA MP.
- ▶ Report service: This component manages the report requests and delivers the results through a web portal named Cognos Connection. The report requests is managed by the Cognos dispatcher. High availability for this component is managed by the Cognos server.
- ▶ Content Manager: This component manages the storage of application data such as models, report specifications, report outputs, configuration data, and security. This information is required for publishing packages, retrieving schedule information, restoring report specifications, and managing the Cognos namespace. The high availability for this component is managed by the Cognos Server. The database resources for the content store is managed by Tivoli SA MP.

From the functionary point of view, the BI nodes are classified into three types:

- ▶ BI type 1 node: This node type is in charge of managing user requests through the Cognos gateway and to process the report requests. The number of reports processed is based on the weight associated with the Cognos dispatcher and varies depending on the number of BI extension nodes. The BI type 1 node also hosts the standby Cognos Content Manager and provides high availability support for the content store. All IBM Smart Analytics System BI module have one type 1 node.
- ▶ BI type 2 node: This node hosts the active Cognos Content Manager, the content store database, and the audit database. It also processes report requests. The gateway on this node provides high availability support to the Cognos gateway on the BI type 1 node. All IBM Smart Analytics System BI module have one type 2 node.
- ▶ BI extension node: The primary role of this type of node is the report processing. This node is configured to have a maximum report processing capacity. The gateway is installed but not used, and the Content Manager is installed but not started. IBM Smart Analytics System 5600 BI module can have zero to four extension nodes. IBM Smart Analytics System 7600 can have zero to two extension nodes.

The IBM Smart Analytics System 5600 consists of two to six BI nodes: one type 1, one type 2, and zero to four extension nodes. Figure 3-7 on page 61 shows the minimum configuration for the IBM Smart Analytics System 5600 BI module, one type 1 BI node and one type 2 BI node.

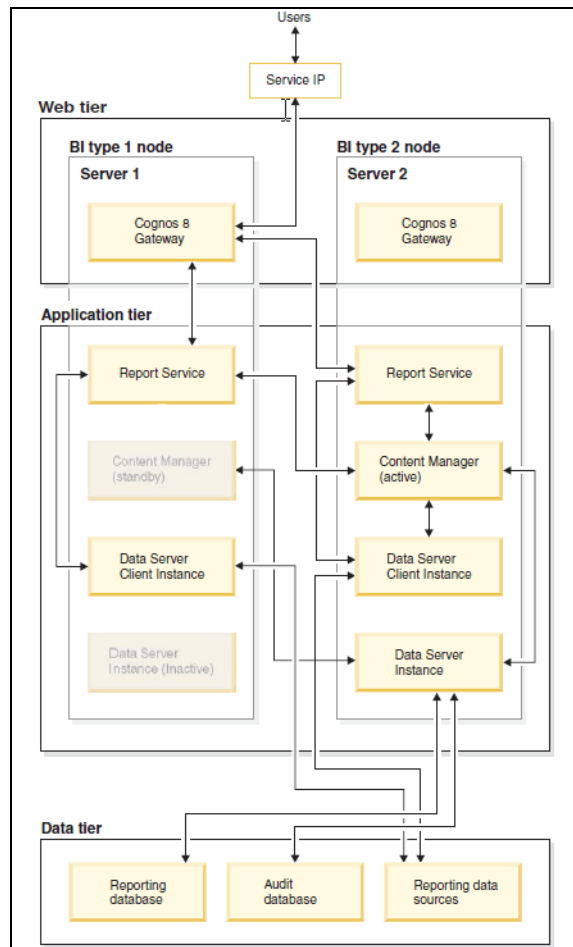


Figure 3-7 BI module with two nodes HA configuration

The IBM Smart Analytics System 7600 consists of two to four BI nodes, one type 1, one type 2, and zero to two extension nodes. Figure 3-8 shows a configuration with one type 1 BI node, one type 2 BI node, and one BI extension node.

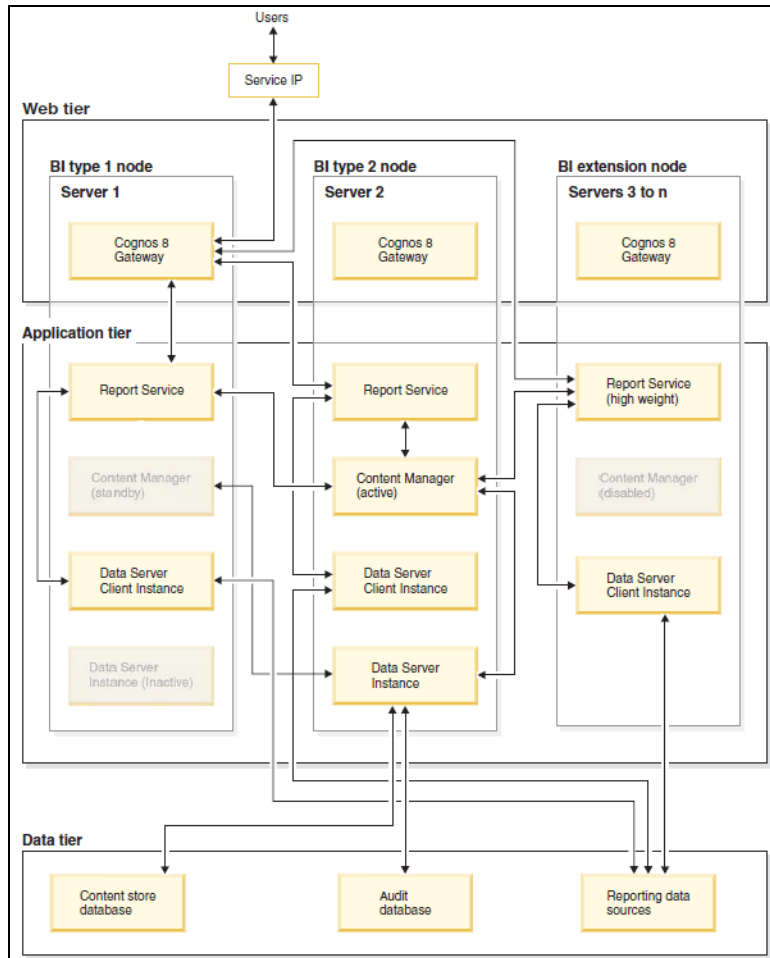


Figure 3-8 BI module with three nodes HA configuration

The high availability strategy for the IBM Smart Analytics System BI module is the mutual failover (active/active configuration). It is implemented for the BI module using native Cognos BI Server functionality to manage the active Cognos Content Manager and Tivoli SA MP to manage high availability resource groups for the Cognos gateway and BI module database resources.

If the BI type 1 node fails, high availability resources managed by Tivoli SA MP detects the failure and transfers the Cognos gateway resource to the BI type 2 node. When a failure occurs on the BI type 2 node, the Cognos BI Server application detects the failure and designates the BI type 1 node as the active Content Manager. Tivoli SA MP also manages the database resources on the BI type 2 node. When a failure occurs, the database resources are transferred to the BI type 1 node.

Because the report processing is available on every node, there is no need to set up the failover node for the BI extension nodes. If a failure occurs on any BI extension node, the report processing can start on another server. The performance to manage the workloads can be impacted.

The high availability resources present on the IBM Smart Analytics System BI nodes are managed by native IBM Cognos BI Server and IBM Tivoli System Automation. The following highly available resources are managed by IBM Cognos BI Server:

- ▶ IBM Cognos Content Manager
- ▶ IBM Cognos Gateway
- ▶ IBM Cognos Report Service

The IBM Tivoli System Automation manages these resources:

- ▶ IBM Cognos gateway resource group:

This resource group, `ihs-rg`, has the service IP for the end users to reach the IBM Cognos gateway. It also has the HTTP server. There are dependencies among the resources managed by Tivoli SA MP, in case of failure of one of these resources, both the failed resources and its dependencies are moved to the secondary node.

- ▶ BI node database resource group:

This resource group, `db2_coginst_0-rg`, holds the BI node database instance with the content-store database, auditing database, and Samples database. It has the resources for the file systems `/cogfs` and `/coghome`, the BI node instance, and the service IP for the management network that handles the connection to the databases under the BI node instance.

The primary role of BI type 1 node is to process the user requests coming from the IBM Cognos gateway. The Tivoli SA MP resource group for the IBM Cognos gateway is assigned to this node. The IBM Cognos Content Manager is in a standby state on the BI type 2 node. If the Tivoli SA MP detects that the server failed or is unreachable, it will automatically transfer the resources in the IBM Cognos gateway resource group to the BI type 2 node, and make the BI type 2 node the active IBM Cognos gateway.

When the BI type 1 node becomes operational again, the gateway service IP address remains assigned to the type 2 node. IBM Smart Analytics System provides a set of scripts to fail back the resources to the BI type 1 node. Run these scripts when there is no activity on the cluster to prevent workload disruption.

When the BI type 2 node server fails or is unable to connect to the internal application network, both the IBM Cognos BI Server application and the Tivoli SA MP detect the failure and automatically perform the failover. The IBM Cognos BI Server application conducts an election to designate a new active Content Manager. As a result of the election process, the BI type 1 node becomes the new active Content Manager. The Tivoli SA MP then transfers the resources in the BI module database resource group to the BI type 1 node, mounts the /cogfs file system, and starts the BI module instance, as well as, the service IP.

Notes: The BI extension nodes are not part of the HA configuration for the IBM Smart Analytics System BI module. The BI extension nodes will only carry out report processing workloads, they cannot assume any resource from the BI type 1 node and BI type 2 node in case of failure.

To fail back the resources for the BI module instance and activate the IBM Cognos Content Manager at the BI type 2 node, activate the IBM Cognos Content Manager on the BI type 2 node first, then move back the BI module instance resource group using the provided script.

Use `lssam` to check the nominal state for the high availability resources. Example 3-12 shows an `lssam` output for a BI module containing one type 1 node and one type 2 node.

Example 3-12 BI module lssam output

```

Online IBM.ResourceGroup:db2_coginst_0-rg Nominal=Online
  |- Online IBM.Application:db2_coginst_0-rs
    |- Offline IBM.Application:db2_coginst_0-rs:BI_module_tp1
    '- Online IBM.Application:db2_coginst_0-rs:BI_module_tp2
  |- Online IBM.Application:db2mnt-cogfs-rs
    |- Offline IBM.Application:db2mnt-cogfs-rs:BI_module_tp1
    '- Online IBM.Application:db2mnt-cogfs-rs:BI_module_tp2
  '- Online IBM.ServiceIP:db2ip_192_168_122_104-rs
    |- Offline IBM.ServiceIP:db2ip_192_168_122_104-rs:BI_module_tp1
    '- Online IBM.ServiceIP:db2ip_192_168_122_104-rs:BI_module_tp2

Online IBM.ResourceGroup:ihs-rg Nominal=Online
  |- Online IBM.Application:ihs-rs
    |- Online IBM.Application:ihs-rs:BI_module_tp1
    '- Offline IBM.Application:ihs-rs:BI_module_tp2
  '- Online IBM.ServiceIP:ihs-sip-rs
    |- Online IBM.ServiceIP:ihs-sip-rs:BI_module_tp1
    '- Offline IBM.ServiceIP:ihs-sip-rs:BI_module_tp2

Online IBM.Equivalency:FCM_network
  |- Online IBM.NetworkInterface:en11:BI_module_tp2
  '- Online IBM.NetworkInterface:en11:BI_module_tp1

Online IBM.Equivalency:db2_coghome_gpfs_BI_module_tp1-BI_module_tp2-equ
  |- Online IBM.AgFileSystem:coghome_BI_module_tp1:BI_module_tp1
  '- Online IBM.AgFileSystem:coghome_BI_module_tp2:BI_module_tp2

```

```

Online IBM.Equivalency:db2_coginst_0-rg_group-equ
    |- Online IBM.PeerNode:BI_module_tp2:BI_module_tp2
    '- Online IBM.PeerNode:BI_module_tp1:BI_module_tp1
Online IBM.Equivalency:ihs_network_equiv
    |- Online IBM.NetworkInterface:en12:BI_module_tp1
    '- Online IBM.NetworkInterface:en12:BI_module_tp2
Online IBM.Equivalency:ihs_nodes_equiv
    |- Online IBM.PeerNode:BI_module_tp1:BI_module_tp1
    '- Online IBM.PeerNode:BI_module_tp2:BI_module_tp2

```

Commands: For the commands described in this section, see the IBM Smart Analytics System 5600 and 7700. For the IBM Smart Analytics System 7600, the installation path is under the /usr directory instead of /opt/.

3.6.1 Starting and stopping high availability resources for BI module

This section describes the procedure to start and stop the high availability resources for BI module.

Starting the BI module

To start the resources on the BI module, perform these steps:

1. Start the cluster domain.

Check if the cluster domain is online using the Tivoli SA MP command **lsrpdomain**. The output is similar to Example 3-13.

Example 3-13 IBM Tivoli System Automation cluster domain for BI module

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
cognos_bi	Online	2.5.4.1	No	12347	12348

If the cluster domain is offline, log in as *root* from the BI type 2 node and start the domain using the following Tivoli SA MP command:

```
starttrpdomain cognos_bi
```

2. Check the status of the Tivoli SA MP resources using **lssam**. Check if all equivalencies are online.
3. Start the resources from the BI type 2 node using the command:

```
chrg -o "Online" -s "1=1"
```

This command starts the BI module database resource and the IBM Cognos gateway resource.

4. Check the status using `lssam`.

Example 3-14 shows a sample output. The resource group for the BI module database resources (`db2_coginst_0`) and the resource group for the IBM Cognos gateway (`ihs-rg`) must appear as Online. If any of the resources failed to start, troubleshoot the problem before starting the application servers. Contact IBM Customer Support for the IBM Smart Analytics System for further assistance.

Example 3-14 BI module lssam output

```
Online IBM.ResourceGroup:db2_coginst_0-rg Nominal=Online
|- Online IBM.Application:db2_coginst_0-rs
|   |- Offline IBM.Application:db2_coginst_0-rs:BI_module_tp1
|   '- Online IBM.Application:db2_coginst_0-rs:BI_module_tp2
|- Online IBM.Application:db2mnt-cogfs-rs
|   |- Offline IBM.Application:db2mnt-cogfs-rs:BI_module_tp1
|   '- Online IBM.Application:db2mnt-cogfs-rs:BI_module_tp2
'- Online IBM.ServiceIP:db2ip_192_168_122_104-rs
|   |- Offline IBM.ServiceIP:db2ip_192_168_122_104-rs:BI_module_tp1
|   '- Online IBM.ServiceIP:db2ip_192_168_122_104-rs:BI_module_tp2

Online IBM.ResourceGroup:ihs-rg Nominal=Online
|- Online IBM.Application:ihs-rs
|   |- Online IBM.Application:ihs-rs:BI_module_tp1
|   '- Offline IBM.Application:ihs-rs:BI_module_tp2
'- Online IBM.ServiceIP:ihs-sip-rs
|   |- Online IBM.ServiceIP:ihs-sip-rs:BI_module_tp1
|   '- Offline IBM.ServiceIP:ihs-sip-rs:BI_module_tp2

Online IBM.Equivalency:FCM_network
|- Online IBM.NetworkInterface:en11:BI_module_tp2
'- Online IBM.NetworkInterface:en11:BI_module_tp1

Online IBM.Equivalency:db2_coghome_gpfs_BI_module_tp1-BI_module_tp2-equ
|- Online IBM.AgFileSystem:coghome_BI_module_tp1:BI_module_tp1
'- Online IBM.AgFileSystem:coghome_BI_module_tp2:BI_module_tp2

Online IBM.Equivalency:db2_coginst_0-rg_group-equ
|- Online IBM.PeerNode:BI_module_tp2:BI_module_tp2
'- Online IBM.PeerNode:BI_module_tp1:BI_module_tp1

Online IBM.Equivalency:ihs_network_equiv
|- Online IBM.NetworkInterface:en12:BI_module_tp1
'- Online IBM.NetworkInterface:en12:BI_module_tp2

Online IBM.Equivalency:ihs_nodes_equiv
|- Online IBM.PeerNode:BI_module_tp1:BI_module_tp1
'- Online IBM.PeerNode:BI_module_tp2:BI_module_tp2
```

5. Check if the DB2 database for the IBM Cognos Content store is available.

From the BI type 2 node, issue `lssam` and make sure that the BI module database resources are Online at the BI type 2 node as shown in Example 3-14. If the resources are not online or are running on the BI type 1 node, try to restart the resource group using the command:

```
chrg -o Offline -s "Name like 'db2_coginst_0'"
```


Then:

```
chrg -o Online -s "Name like 'db2_coginst_0'"
```

If they failed to start at the BI type 2 node, more troubleshooting needs to be performed. Contact IBM Customer Support for the IBM Smart Analytics System for further assistance if required.

6. Verify database connection.

Log on to the BI type 2 node using the content store instance owner *coginst* user. Connect to the Content Store database:

```
db2 connect to csdb
```

Here, csdb is the content store database.

If the database is failing to connect, look at the database logs to find out what causes the connection to fail.

7. Start the application server.

Log onto the BI type 2 node as the *cognos* user (this is the default the user ID). Start the application server on the BI nodes, beginning from the BI type 2 node and then the BI type 1 node, using the following command:

```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin/  
./startServer.sh server1
```

8. Verify if the IBM Cognos Content Manager is running.

Check if the IBM Cognos Content Manager is running on the BI type 2 node by access the Content Manager status page at the URL:

<http://<hostBInode2>:9081/p2pd/servlet>

Here, *<hostBInode2>* is the host name of the BI type 2 node.

Make sure that the state is Running. If it is Running as Stand By, activate the Content Manager at the BI node Type 2 using the procedure described at “Designating the active IBM Cognos Content Manager” on page 72.

Content store: The application server must be started when the content store database is available. If it was started when the content store was not available, refresh the application server using the following commands (logged as the *cognos* user):

```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin/  
./stopServer.sh server1  
./startServer.sh server1
```

9. After the application server is started, verify if the dashboarding and the reporting features is available by accessing the IBM Cognos Connection Portal.

For further reference about how to access IBM Cognos applications, see the IBM Smart Analytics System product documentation.

Stopping the BI module

To stop the resources on the IBM module, perform the following steps:

1. Make sure that there are no running workloads on the environment.

The stop procedure must be planned and performed within a maintenance window to avoid interrupting the reporting processing.

2. Stop the application server.

To avoid unnecessary failover, stop the application server on the BI extension node first (if there is a BI extension node), then stop the application server on the BI type 1 node before stop the application server on the BI type 2 node.

To stop the application server, use the following command:

```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin
./stopServer.sh server1
```

If an application server is stopped properly, you see a message similar to this:

```
Server server1 stop completed
```

3. Stop the high availability resources.

To stop the high availability resources managed by Tivoli SA MP, use the following command:

```
chrg -o offline -s "1=1"
```

Verify the resource status using the **1ssam** command. All resources must be offline.

4. Bring the cluster domain off line.

After all resources are offline, you can place the cluster domain offline using the following command.

```
stoprpdomain cognos_bi
```

Check the domain status using the **1srpdomain** command. Example 3-15

Example 3-15 BI module HA cluster domain offline

Name	OpState	RSCTActiveVersion	MixedVersions	TSPort	GSPort
cognos_bi	Offline	2.5.4.1	No	12347	12348

3.6.2 Manual failover BI type 1 node to BI type 2 node

The IBM Smart Analytics System BI type 1 node hosts the high availability resources for the IBM Cognos gateway and handles the report processing workloads for the data warehouse environment. You can manually move the resources from the node type 1 over to the BI type 2 node for performing maintenance activities.

To manually fail over the resources running on BI type 1 node to the BI type 2 node, perform the following steps:

1. Stop the application server running on the BI type 1 node.

Log onto the BI type 1 node with the *cognos* user, stop all dashboard and reporting processing using the following commands:

```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin
./stopServer.sh server1
```

2. Move the IBM Cognos gateway resources to the BI type 2 node.

Log on to the BI type 1 node as *root* and run the following commands:

```
cd /root/scripts/tsa
./failover_sip.sh
```

Using **1ssam** to check if the high availability resources are stopped on the BI type 1 node.

3. Access the IBM Cognos Connection Portal to verify if the dashboarding and reporting activities are available.

After the BI type 1 node becomes available to resume its duty, use the process described in “Manual failback BI type 1 node” on page 70 to fail back its resources manually.

3.6.3 Manual failover BI type 2 node to BI type 1 node

The IBM Smart Analytics System BI type 2 node hosts the high availability resources for the IBM Cognos Content Manager, BI module instance, and the report processing workloads for the data warehouse environment. You can manually move the BI type 2 node resources over to the BI node type1 for maintenance activities.

To manually fail over the resources running on BI type 2 node to the BI type 1 node, perform the following steps:

1. Stop the application server on the BI type 2 node.

Log onto the BI type 2 node with the *cognos* user, run these commands to stop the application server:

```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin
./stopServer.sh server1
```

2. Move the BI module instance resources to the BI type 1 node.

Log onto the BI type 2 node with the root user, run the following commands:

```
cd /root/scripts/tsa
./failover_db2.sh
```

Check the high availability resource status with the **Issam** command. Verify if all BI module instance resources, *db2_coginst_0*, *cogfs*, *coghome*, and the database resource *ServiceIP*, are online on the BI type 1 node.

3. Access the IBM Cognos Connection Portal to check if the dashboarding and report services are available.

If you experience problems to access the IBM Cognos Connection Portal, find the problem using the procedure describe in “Troubleshooting connections after failover of BI type 2 node” on page 74.

3.6.4 Manual failback BI type 1 node

After the BI type 1 node is available again, the node type 1 resource running on BI type 2 node must be manually failed back to the BI type 1 node. Similar to all failback operations, this activity must be planned to avoid disrupting the running workloads.

Before performing the failback, check the nominal status for the resources using the Tivoli SA MP command **Issam**. The resource for the gateway when running in failover mode appears like in Example 3-16.

Example 3-16 Issam excerpt output showing gateway resources

```
Online IBM.ResourceGroup:ihs-rg Nominal=Online
|- Online IBM.Application:ihs-rs
   |- Offline IBM.Application:ihs-rs:BI_module_tp1
   '- Online IBM.Application:ihs-rs:BI_module_tp2
'- Online IBM.ServiceIP:ihs_192.168.182.200
   |- Failed Offline IBM.ServiceIP:ihs_192.168.182.200: BI_module_tp1
   '- Online IBM.ServiceIP:ihs_192.168.182.200: BI_module_tp2
```

If the BI type 1 node is operational and all equivalencies are online the gateway resources, you can start the failback process using these steps:

1. Log in to the BI type 2 node as the root user.
2. Verify that the BI type 2 node currently owns the gateway Service IP using the `1ssam` command.
3. Move the resources contained in the gateway resource group to the BI type 1 node:

```
cd /root/scripts/tsa
./failover_sip.sh
```
4. Verify that the BI type 1 node owns both gateway Service IP resource and HTTP server resource:
 - a. Log in to the BI type 1 node as the root user.
 - b. Issue the `1ssam` command, and verify the following conditions:
 - BI type 1 node has an Online operational status for both resources.
 - BI type 2 node has an Offline operational status for both resources.
 - The resource group has an Online operational status.
 - c. Use the `ifconfig -a` command to check if the BI type 1 node has one Ethernet adapter with a second *inet* entry that corresponds to the gateway Service IP address.

3.6.5 Manual failback BI type 2 node

The IBM Smart Analytics System BI type 2 node, in normal situations, is holding the high availability resources for the IBM Cognos BI Server application, IBM Cognos Content Manager, and the BI module instance. There can be three resource failure scenarios on the BI type 2 node:

- ▶ System failure of type 2 node: The content store database resource group and active IBM Cognos Content Manager are transferred to the BI type 1 node.
- ▶ The IBM Cognos BI Server application failure: The IBM Cognos Content Manager on the BI type 2 node fails and the IBM Cognos Content Manager on the BI type 1 node becomes active.
- ▶ The BI module instance failure: The databases managed by the BI module instance become inaccessible but the IBM Cognos Content Manager remains active on the BI type 2 node.

Depending on the cause of the failure, various failback procedures are applied. To find out which component failed on BI type 2 node, use the following methods:

- ▶ Check the IBM Cognos Content Manager status page:

Go to the URL:

`http://<hostBItype2>:9081/p2pd/servlet`

Here, *<hostBItype2>* represents the host name of the BI type 2 node.

If the state of the Content Manager is Running, then the Content Manager on the BI type 2 node is the active. Otherwise, either the IBM Cognos BI Server application has failed on the BI type 2 node, or that the BI type 2 node has experienced a system failure.

- ▶ Check the nominal state for the high availability resources:

Use Tivoli SA MP command `Issam` to check the nominal state the resources contained in the BI module instance resource group (db2_coginst_0-rg). If the operational state is Failed Offline, then either the BI module instance has failed or the BI type 2 node has experienced a system failure. If the BI module instance resources have an operational state of Online, then the Cognos BI Server application is the resource that has failed.

After the failed resources are identified and fixed, the BI type 2 node is ready to reassume its resources. Before proceed with the failback procedure, check if all equivalencies are online using the `Issam` command. Just as with all failback operations, this activity must be planned to avoid disrupting the running workloads.

Apply the failback procedures based on the failure occurred and found on the BI type 2 node:

- ▶ If the BI type 2 node has experienced a system failure, proceed with “Designating the active IBM Cognos Content Manager” and “Moving BI module database resources” on page 73.
- ▶ If the IBM Cognos BI Server has failed on the BI type 2 node, proceed with “Designating the active IBM Cognos Content Manager” on page 72.
- ▶ If the BI module instance has failed, and the database resources were moved to BI type 1 node, proceed with “Moving BI module database resources” on page 73 to manual failback the resources to BI type 2 node.

Designating the active IBM Cognos Content Manager

When the IBM Cognos Content Manager was failed over to BI type 1 node, the IBM Cognos Content Manager at BI type 1 node, the primary location, becomes standby. To fail the IBM Cognos Content Manager back from BI type 1 node to BI type 2 node, use the following procedure to designate the active IBM Cognos Content Manager:

1. Access the IBM Cognos Connection portal at the URL:
`http://serviceIP/cognos8`
Here, *serviceIP* represents the IBM Cognos gateway Service IP address.
2. Click **Launch** → **IBM Cognos Administration** to start the Cognos Administration portlet.
3. Click the **System link** on the left pane.
4. Click the host name of the node that you want to designate as the active IBM Cognos Content Manager. For example, to designate the Content Manager on the BI type 2 node as the active Content Manager, click **http://hostBInode2** (where *hostBInode2* represents the host name of the type 2 node).
5. Click the dispatcher for the node, which is represented by the suffix :9081/pdpd on the URL for the node. For example, the dispatcher for the BI type 2 node is represented as `http://hostBInode2:9081/pdpd`.
6. Click the **ContentManagerService** entry.
7. Click the **Action** drop-down arrow next to the ContentManagerService entry identified in the previous step.
8. Click **Activate** to activate the Content Manager on the type 2 node. If you do not see the Activate link, the Content Manager on the type 2 node is already active.
9. Check the status page for both BI nodes, and confirm that the State for the Content Manager is displayed as *Running* on the BI type 2 node, and is displayed as *Running as Standby* on the BI type 1 node.

Caution: When performing a failback, do *not* select “Set as active by default”. This option will make the designated Content Manager assume the role of the active Content Manager when the type 2 node is brought online. It will make the failback of the Content Manager automatic and will result in a disruption of the running workloads.

Moving BI module database resources

Follow these steps to move the BI module database resources back to its primary location, the BI type 2 node from BI type 1 node:

1. Log in to the BI type 1 node or type 2 as the root user.
2. Move the resources contained in the content store resource group to the BI type 2 node:

```
cd /root/scripts/tsa
./failover_db2.sh
```

3. Verify that the BI type 2 node has the resources contained in the BI module database resource group including the BI module database Service IP resource, BI module instance, and resources for the /cogfs file system.
 - a. Issue the **lssam** command and verify:
 - The BI type 2 node has an Online operational status for all resources.
 - The BI type 1 node has an Offline operational status for all resources.
 - The BI module database resource group has an Online operational status.
 - b. Issue the **ifconfig -a** command to verify that the type 2 node has one adapter with the second *inet* entry that corresponds to the content store Service IP address.

Troubleshooting connections after failover of BI type 2 node

The IBM Smart Analytics System BI type 2 node holds the Content Store database. In case of the BI type 2 node experiences a system failure situation and connections to the content store remains open, you might not be able to log onto the IBM Cognos Connection Portal. To work around this situation, perform the following steps:

1. Stop the application server that is running on all BI nodes using the *cognos* user:


```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin
./stopServer.sh server1
```
2. Log on to the BI type 1 node as the DB2 instance owner *coginst*, and check if there are remain connections:


```
db2 list applications for database csdb
```

Here, csdb is the Content Store database.

If there are connections, force all applications connected off the csdb.
3. Restart the application server *first* on the BI type 1 node, then on the BI type 2 node, followed by the BI extension nodes (if existing):


```
cd /opt/IBM/WebSphere_7/AppServer/profiles/AppSrv01/bin
./startServer.sh server1
```

Test the connection on the IBM Cognos Connection Portal, if the problem persists, more troubleshooting must be performed. For further assistance, contact IBM Customer Support for the IBM Smart Analytics System.

For further information about the topics discussed in this section, see the *IBM Smart Analytics System User's Guide* for your respective version.



Maintenance

Similar to other systems, the IBM Smart Analytics System requires maintenance. When an IBM Smart Analytics System is assembled at the IBM Customer Service Center (CSC), all hardware components with firmware are verified for the right release level and updated as required. Every selected software and firmware level are tested as an integrated stack.

IBM periodically tests new firmware or software versions as part of a validated stack for the IBM Smart Analytics System. It is important to follow the IBM specifications on upgrades to the IBM Smart Analytics System so that your system remains on a validated stack level.

In this chapter we discuss the maintenance procedures to be followed for the IBM Smart Analytics System, including the backup and recovery strategies for key components such as the database and the operating system.

4.1 Managing DB2 message logs

To help you administer and monitor the database activities, DB2 provides diagnostic files, notification files, error logs, trap files, and dump files. For example, DB2 logs the database activity messages to the db2diag log file, and when a problem occurs, many diagnostic files might be created in the db2dump directory. By default, DB2 does not delete these log files, you have to manage them to prevent these log files from taking up too much disk space.

In this section, we discuss the tools and tips for managing the DB2 logs.

4.1.1 The db2dback shell script

The **db2dback** script is a shell script that allows you to back up the DB2 diagnostic data from the diagpath directory to a specified destination. This script works on both single database partition and multi-database partition environments with rotating and non-rotating logs. You must run this script on the first administration node. It will connect to all database partitions and archives the diagnostic and message data to a specified file system. This script supports both AIX and Linux environments.

In addition to archiving the diagnostic data, **db2back** also allows you to maintain the data that has already been archived at the destination directory. The archived files might be compressed (an option) and purged after a specified number of days.

For example, to archive and compress logs that are older than three days and delete if older than seven days, use this command:

```
./db2dback.ksh -a tz 3 -r 7
```

For more details about this command and to download the **db2dback.ksh** shell script, see this developerWorks article, *Archive and maintain DB2 message logs and diagnostic data with db2back*, located at this address:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-0904db2message1ogs/index.html?ca=dth-grn&ca=dgp-my>

4.1.2 db2support -archive

The **db2support** problem analysis and environment collection tool has a new feature to help managing DB2 logs, the **-archive** option. This option, introduced in DB2 9.7 Fix Pack 1, creates a copy of the contents of the log directory specified by the diagpath database manager configuration parameter into an archive path that you specify.

The naming convention of the archive directory is `DB2DUMP_<host name>_<current timestamps>`. All files under the source log directory are deleted after archive.

Example 4-1 shows how to use the `db2support -A` command or the `db2support -archive` command to archive the DB2 diagnostic log files to the `/db2home/bculinux/SLCF/` directory.

Example 4-1 db2support output

```
bculinux@ISAS56R1D1:~> db2support -A /db2home/bculinux/SLCF
```

```
_____ D B 2   S u p p o r t   _____
```

This program generates information about a DB2 server, including information about its configuration and system environment. The output of this program will be stored in a file named 'db2support.zip', located in the directory specified on the application command line. If you are experiencing problems with DB2, it may help if this program is run while the problem is occurring.

NOTES:

1. By default, this program will not capture any user data from tables or logs to protect the security of your data.
2. For best results, run this program using an ID having SYSADM authority.
3. On Windows systems you should run this utility from a db2 command session.
4. Data collected from this program will be from the machine where this program runs. In a client-server environment, database-related information will be from the machine where the database resides via an instance attachment or connection to the database.

Attempting to archive files from DIAGPATH "/db2fs/bculinux/db2dump".

DIAGPATH data have been successfully archived into
"/db2home/bculinux/SLCF/DB2DUMP_ISAS56R1D1_2010-10-06-21.59.31".

4.1.3 The db2diag utility

Though the db2diag log files are intended for use by IBM Software Support for troubleshooting purposes. DB2 database administrators frequently check the db2diag log files for system messages. In a partitioned environment such as the IBM Smart Analytics System, every administration node and data node has its own db2diag log files.

Finding information in multiple error log files can become time consuming work. The **db2diag** utility is meant to filter and format messages in the db2diag log files. These two options can be helpful for combining the db2diag log files:

- ▶ **-global**: This option includes all the db2diag log files from all the database partitions on all the hosts in the log file processing.
- ▶ **-merge**: This option merges diagnostic log files and sorts the records based on the timestamp.

Specifying the **-global** and **-merge** options together consolidates all the db2diag log files and sorts the records based on the timestamp. Both options support rotating diagnostic log files and files located in split diagnostic data directories.

You must be the instance owner to run **db2diag** from the administration node.

The following **db2diag** command example searches all the data nodes for the db2diag log files, extract any messages classified as severe, and writes them to a single output file named db2diag.test:

```
db2diag -global -merge -sdir /db2home/bculinux -l severe > ./db2diag.test
```

Example 4-2 shows an excerpt of the combined db2diag log file.

Example 4-2 Contents of the combined db2diag log file

```
2010-09-22-13.15.29.668213-300 I1E1563 LEVEL: Event
PID : 26707 TID : 47168874669168PROC : db2stop
INSTANCE: bculinux NODE : 000
FUNCTION: DB2 UDB, RAS/PD component, pdLogInternal, probe:120
START : New Diagnostic Log file
DATA #1 : Build Level, 152 bytes
Instance "bculinux" uses "64" bits and DB2 code release "SQL09072"
with level identifier "08030107".
Informational tokens are "DB2 v9.7.0.2", "s100514", "IP23089", Fix Pack "2".
DATA #2 : System Info, 440 bytes
System: Linux ISAS56R1D1 6 2 x86_64
CPU: total:16 online:16 Cores per socket:8 Threading degree per core:1
Physical Memory(MB): total:64436 free:58831
Virtual Memory(MB): total:97210 free:91605
Swap Memory(MB): total:32774 free:32774
Kernel Params: msgMaxMessageSize:65536 msgMsgMap:65536 msgMaxQueueIDs:64512
msgNumberOfHeaders:65536 msgMaxQueueSize:65536
msgMaxSegmentSize:16 shmMax:9223372036854775807 shmMin:1
shmIDs:16128 shmSegments:16128 semMap:256000 semIDs:16128
semNum:256000 semUndo:256000 semNumPerID:250 semOps:32
semUndoSize:20 semMaxVal:32767 semAdjustOnExit:32767
Cur cpu time limit (seconds) = 0xFFFFFFFF
Cur file size limit (bytes) = 0xFFFFFFFF
Cur data size (bytes) = 0xFFFFFFFF
Cur stack size (bytes) = 0x00800000
Cur core size (bytes) = 0x00000000
```

```
Cur memory size (bytes) = 0xFFFFFFFF
nofiles (descriptors) = 0x00000800
...
```

You also can filter the message and format the text of the combined log file. To select just messages classified as error, severe, or critical, and write them to a single output file named `db2diag.<current_date>.out`, use the following command:

```
db2diag -global -merge -sdir /db2home/instance_name -level
Error,Severe,Critical -fmt "%{ts}\tSeverity: %{level} \nInstance:
%{inst}\tNode:%{node}\nFunction: %{function}\nError: %{msg}\nDescription:
%{rc}\n" > /db2home/instance_name/db2diag.`date +%Y%m%d`.out
```

Example 4-3 shows an excerpt of the formatted `db2diag` file content.

Example 4-3 Formatted db2diag file contents

```
2010-09-22-13.15.50.555028      Severity: Error
Instance: bculinux      Node:001
Function: DB2 UDB, common communication, sqlcctcpconnmgr, probe:50
Error: ADM7006E The SVCENAME DBM configuration parameter was not
        configured. Update the SVCENAME configuration parameter using the
        service name defined in the TCP/IP services file.
Description:

2010-09-22-13.15.52.365632      Severity: Error
Instance: bculinux      Node:005
Function: DB2 UDB, common communication, sqlcctcpconnmgr, probe:50
Error: ADM7006E The SVCENAME DBM configuration parameter was not
        configured. Update the SVCENAME configuration parameter using the
        service name defined in the TCP/IP services file.
Description:

2010-09-22-13.16.09.155469      Severity: Error
Instance: bculinux      Node:000
Function: DB2 UDB, common communication, sqlcctcpconnmgr, probe:50
Error: ADM7006E The SVCENAME DBM configuration parameter was not
        configured. Update the SVCENAME configuration parameter using the
        service name defined in the TCP/IP services file.
...

```

For more details on `db2diag` utility, see DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html>

4.2 Changing the date and time

An IBM Smart Analytics System is comprised of many servers running copies of a DB2 relational database. Certain subdirectories in the administration node are mounted in all other DB2 nodes (data nodes, user nodes, and failover nodes) through NFS or GPFS mount points, for example, /home or /db2home.

A few software and operating system components of the IBM Smart Analytics System require that the clocks on all the involved servers are synchronized. These components include software such as DB2 partitioned database, Tivoli System Automation for Multiplatform, NFS, and GPFS.

Date and time: The date and time settings for all IBM Smart Analytics System servers must be the same, within a few minutes of tolerance. Otherwise, the NFS and the GPFS directories will be mounted but inaccessible.

To run commands across all servers, you can build a script as shown in Example 4-4. The password-less `ssh` is set up for the root user between all nodes in the cluster, and for the DB2 instance owner for all nodes in the core warehouse instance.

Example 4-4 Running command across all IBM Smart Analytics System Servers

```
for i in IBMSMAS56R1ADM1 IBMSMAS56R1DTA1 IBMSMAS56R1DTA2 IBMSMAS56R1STDB1
do
    echo "\nRunning command $1 on NODE $i \n"
    ssh $i "$1"
done
```

Here, IBMSMAS56R1ADM1, IBMSMAS56R1DTA1, IBMSMAS56R1DTA2, and IBMSMAS56R1STDB1 are the host names for the servers belonging to the IBM Smart Analytics System installation.

On the example, the file was saved as `run_cmd.sh` under `/BCU_share` directory. This directory is an NFS file system mounted across all IBM Smart Analytics System servers.

Be sure to test if the script is properly configured with a simple command such as **date** or **uptime**. Run the command and carefully check if all the IBM Smart Analytics System servers are listed, including the management node, administration node, data nodes, user nodes, standby nodes, warehouse applications node, warehouse OLAP node, and business intelligence nodes.

Example 4-5 shows how to test the run_cmd.sh script with date.

Example 4-5 Testing the run_cmd.sh script

```
ISAS56MGMT:/BCU_share # ./run_cmd.sh "date"
```

To change the date and time for an IBM Smart Analytics System, follow these steps:

1. Stop all activities on business intelligence (BI) module and warehouse application module.
2. Stop DB2 Performance Expert.
3. Stop all user and application connections to the database. Then deactivate the database.
4. Optional: Back up, then delete db2diag.log and notify.log.
5. If HA is implemented, stop the DB2 resources using the **hastopdb2** command. Verify that the resources are offline. If HA is not implemented, stop the instance using the **db2stop** command.
6. Verify that the application servers resources are offline.
7. Verify that the BI module resources are offline.

Important: Do not change the date when the system is running with more than one user. For more details, see this address:

<http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=/com.ibm.aix.baseadm/doc/baseadmndita/datecommand.htm>

8. Unmount the NFS or GPFS shared subdirectories.
9. Update the date and time of the management node using the **date** command. For example, to set the date to December 25 14:53:00 (2010), use the command **date 1225145310**.
10. Update the date and time on all other IBM Smart Analytics System servers.
Example 4-6 shows how to update the date and time using the run_cmd.sh script.

Example 4-6 Updating the date and time using run_cmd.sh script

```
ISAS56MGMT:BCU_share # ./run_cmd.sh "date 1225145310"
```

11. List the date and time for all servers and check the output carefully to make sure they are all synchronized to the same minute. Example 4-7 shows how to check the date and time.

Example 4-7 Checking the updated date and time

```
ISAS56MGMT:/BCU_share # ./run_cmd.sh "date"
```

12. Mount the NFS or GPFS subdirectories.

13. Verify the changes.

As the instance owner, change the current directory to /db2home and list its contents using the **ls** command. If the command prompt hangs, then one or more servers are not at the same date and time as the other servers.

Unmount the NFS or GPFS subdirectories again and check the date and time on all the DB2 servers. Correct this problem before proceeding.

Mount the NFS or GPFS subdirectories again. If you can now list the contents of the /db2home directory, the date and time of all DB2 Servers are updated and correct. The NFS or GPFS subdirectories are also properly mounted and the IBM Smart Analytics System can be put back to work:

- a. If HA is implemented, start the DB2 resources. If HA is not implemented, start the DB2 instance.
- b. Activate the database.
- c. Start the DB2 Performance Expert at the management node.
- d. Start the resources in the warehouse applications module and the BI module.

14. Take a full database backup.

It is best to take a full database backup after verification because as this can compromise the transaction log files for rollforward depending on the date and time change.

4.3 IBM Smart Analytics System upgrades

In this section we discuss the upgrade procedures for firmware and software in the IBM Smart Analytics System family offering.

4.3.1 IBM Smart Analytics System software and firmware stacks

The IBM Smart Analytics System and InfoSphere Balanced Warehouse® validated stack pages website provides links to all validated software and firmware stack pages for IBM Smart Analytics System and InfoSphere Balanced Warehouse configurations. Consult this site for the information about the firmware at this address:

<http://www-01.ibm.com/support/docview.wss?rs=0&uid=swg21429594>

It is best to keep your IBM Smart Analytics System on a validated stack level. Do not apply a fix pack just because it is the latest release. Doing so will cause your system to move off of a validated stack and you might experience problems by using an untested configuration.

If you must deviate from the validated firmware and software stack, see the Frequently Asked Questions about software and firmware upgrades for the IBM Smart Analytics System and the InfoSphere Balanced Warehouse at this address:

<http://www-01.ibm.com/support/docview.wss?rs=3354&uid=swg21328726>

4.3.2 The Dynamic System Analysis tool

To collect software and firmware information about the Linux-based IBM Smart Analytics System offerings, use the Dynamic System Analysis (DSA) tool.

The DSA is usually already on the management node at the `/opt/IBM/DSA` directory. If it is not installed or if it is an older version, download the new version from IBM. Make sure to select the *installable* version of DSA, not the *portable* version. Download the software and the *User's Guide* from this address:

http://publib.boulder.ibm.com/infocenter/toolctr/v1r0/index.jsp?topic=/dsa/dsa_main.html

To run DSA, log on as the *root* user on the management node. The DSA command that generates report is `collecta11` that, if run without any options, generates a compressed XML file to be sent to IBM support.

To obtain help about the command enter this command `collecta11 -h` or `collecta11 -?`.

To change the default report from compressed XML file to HTML format, navigate to the `/opt/IBM/DSA` directory and run this command: `collecta11 -x -v`

The parameters on the command provide this function:

- ▶ `-x`: Suppress the compressed XML file.
- ▶ `-v`: Generate HTML output.

Example 4-8 shows how to generate a software and firmware report using the DSA `collecta11` command.

Example 4-8 Generating a software and firmware report

```
ISAS56R1D1:/opt/IBM/DSA # ./collectall -x -v
Dynamic System Analysis Version 3.02.56

Logging console output to file
/var/log/IBM_Support/DSA_Output_ISAS56R1D1_20101004-202326.txt
Logging level set to Status.

Running DSA collector providers pass 1.
  libasupriver: Advanced Setting Utility(ASU) Setting Collector
  libbist: BIST
  libdiskmgt: Disk Management Information Collector
...
```

The generated report files are under the subdirectories of the default DSA output directory `/var/log/IBM_Support`. Example 4-9 shows the directories and files of the DSA report.

Example 4-9 DSA report directories and files

```
ISAS56R1D1:/var/log/IBM_Support # ls -l
total 8
drwxr-xr-x 2 root root 4096 2010-10-04 20:26 7947AC1_KQXFCGM_20101004-202326
-rw-r--r-- 1 root root 347 2010-10-04 20:13 DSA_Output_ISAS56R1D1_20101004-201325.txt
ISAS56R1D1:/var/log/IBM_Support #
ISAS56R1D1:/var/log/IBM_Support # cd 7947AC1_KQXFCGM_20101004-202326
ISAS56R1D1:/var/log/IBM_Support/7947AC1_KQXFCGM_20101004-202326 # ls -la
total 19732
drwxr-xr-x 2 root root 4096 2010-10-04 20:26 .
drwxr-xr-x 3 root root 4096 2010-10-04 20:51 ..
-rwxr-xr-x 1 root root 4941 2010-10-04 20:26 banner_left.jpg
-rwxr-xr-x 1 root root 9744 2010-10-04 20:26 banner_right.jpg
-rw-r--r-- 1 root root 1556 2010-10-04 20:26 bist.html.html
-rwxr-xr-x 1 root root 509 2010-10-04 20:26 call.jpg
-rwxr-xr-x 1 root root 59936 2010-10-04 20:26 calendarPopup.js
-rw-r--r-- 1 root root 52634 2010-10-04 20:26 chassis_event.html
-rw-r--r-- 1 root root 252 2010-10-04 20:26 diags.html
-rwxr-xr-x 1 root root 35551 2010-10-04 20:26 dom.js
...
-rw-r--r-- 1 root root 718 2010-10-04 20:26 index.html
...
```

In this example, a subdirectory named `7947AC1_KQXFCGM_20101004-202326` was generated. Copy the whole directory to a Windows client, open the **index.html** file with a web browser. The output is detailed and easy to use, click the link for the information you want to know more about. Figure 4-1 shows a Dynamic Systems Analysis HTML report.

Software
[System Overview](#)
[Installed Packages](#)
[Kernel Modules](#)
[Network Settings](#)
[Resource Utilization](#)
[Processes](#)
[OS Configuration](#)
[Linux Config Files](#)

Hardware
[Hardware Inventory](#)
[PCI Information](#)
[Firmware/VPD](#)
[IMM Configuration](#)
[Environmentals](#)
[Drive Health](#)
[LSI Controller](#)
[LSI IDE Controller](#)
[ServeRAID](#)
[ServeRAID Logs](#)
[Qlogic](#)
[Emulex](#)
[VMware ESXi](#)
[Light Path](#)
[IMM Built-in Self Test](#)

Linux Logs
[/var/log/boot.msg](#)
[/var/log/mail.err](#)
[/var/log/mail.warn](#)
[/var/log/messages](#)
[/var/log/warn](#)

System Overview

Computer System

Manufacturer	IBM
Version	00
Product Name	System x3650 M2-[7947AC1]
Serial Number	KQXFCGM
System UUID	ad578699-c974-3904-aa3a-5ad485f07e4f

Operating System

Computer Name	ISAS56R1D1
Product Name	LINUX
Build Number	SUSE Linux Enterprise Server 10 (x86_64) (PATCHLEVEL = 2)
Vendor	Novell, Inc.
Kernel Name	Linux
Kernel Release	2.6.16.60-0.21-smp
Hardware Platform	x86_64
Uptime	48 days 11 hours 52 minutes 42 seconds
Time of Last Boot	08/17/2010 09:15:40

TimeZone

LocalDateTime	10/04/2010 20:24:10
Current Time Zone	EST

Current User

User Name	root
-----------	------

Report Highlights

Message	/var/log/warn: Log collection truncated after 1024 entries
---------	--

Figure 4-1 Dynamic Systems Analysis HTML report.

4.3.3 IBM Smart Analytics System Control Console

The IBM Smart Analytics System Control Console provides systems management capability for the IBM Smart Analytics System. The console can update firmware and software by installing fix packs downloaded from Fix Central. The IBM Smart Analytics System Control Console also allows you to view detailed information about the hardware and software components in the system, and to change passwords for specific users across the entire system.

The IBM Smart Analytics System Control Console supports certain 5600, 7600, and 7700 offerings. For the details about IBM Smart Analytics System Control Console, see the *IBM Smart Analytics System User's Guide* for your respective version.

4.4 IBM HealthCheck Service

The IBM HealthCheck Service is designed to ensure that an IBM Smart Analytics System is still performing at its optimal level. This service must be carried out by the IBM services team after the sixth and twelfth months of an IBM Smart Analytics System installation. After that, it must be run once a year.

The HealthCheck Service provides an in-depth IBM Smart Analytics System analysis on the following areas:

- ▶ Overall IBM Smart Analytics System configuration review:
 - Adherence to standard IBM Smart Analytics System methodology
 - Operating system, database, database tools and storage subsystem levels
- ▶ Operating System review:
 - Conformance to IBM Smart Analytics design
 - Error logs
- ▶ DB2 instance review:
 - Conformance to IBM Smart Analytics methodology
 - DB2 instance-level settings
 - Instance level error logs
- ▶ DB2 database review:
 - Conformance to IBM Smart Analytics methodology
 - Object layout and definition
 - Use of DB2 new features
 - DB2 database-level settings
- ▶ Hardware management review:
 - Logged system events
 - Customer notification settings
 - Call home settings
- ▶ Storage Subsystem review:
 - Profile and configuration conformance to IBM Smart Analytics methodology
 - Storage Subsystem event logs
 - Firmware level
- ▶ Operational considerations:
 - Monitoring of database during peak and off-peak times

For more information about IBM HealthCheck Service, visit this address:

http://public.dhe.ibm.com/software/data/sw-library/services/InfoSphere_Warehouse_HealthCheck.pdf

4.5 IBM Smart Analytics System installation report

When an IBM Smart Analytics System installation and configuration is completed at the IBM Customer Solution Center, an installation report is created. This report includes system information and the results of the quality assurance tests performed. The same tests are run again at the customer site as part of the deployment, and the test results are added to the installation report. Customers are encouraged to read this report and use it as a reference in case information about the IBM Smart Analytics System is needed. It contains detailed information about the entire system.

Figure 4-2 shows the front page of an installation report.

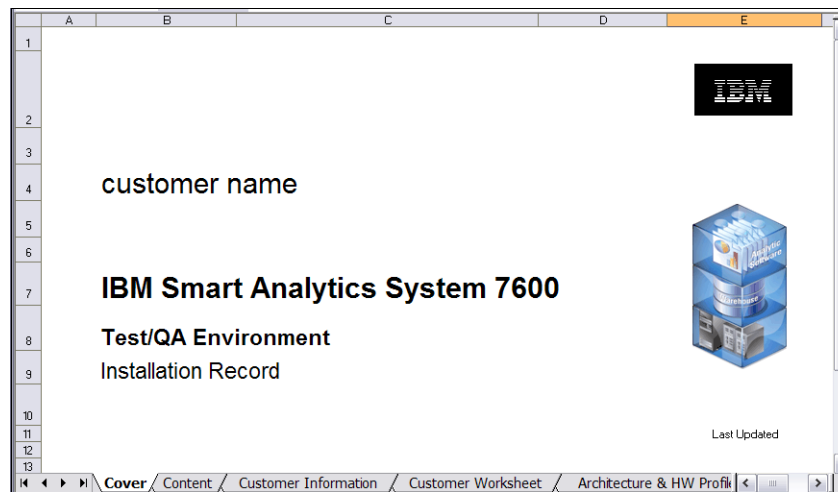


Figure 4-2 Installation report front page

You can find details about system configuration and test results in the area shown in Figure 4-3. The initial AIX and DB2 configurations, as well as the high availability configurations, are listed.

	A	B	C	D	E
2					
3		Table of Contents			
4					
5		Customer Information			
6		Customer Worksheet			
7		Architecture and Hardware Profile			
8		Rack Diagram			
9		Software Stack			
10		Network Configuration			
11		Network Point-to-point Diagram			
12		Storage Subsystem Configuration			
13		Fibre Point-to-point Diagram			
14		AIX Configuration			
15		File System			
16		DB2 Configuration			
17		DB2 HA Configuration			
18		InfoSphere Warehouse Application Server HA Configuration			
19		Network Throughput Validation			
20		I/O Performance Validation			
21		DB2 Performance Validation			
22		High Availability Test			
23		System Certification			
24		References and Next Steps			
< > Cover / Content / Customer Information / Customer Worksheet / Architecture & HW Profile					

Figure 4-3 Installation report table of contents

4.6 IBM Smart Analytics System backup and recovery

Data is one of the critical assets for an enterprise. Ensuring the availability of this asset is the most important matter when architecting a data warehouse environment. Implementing an efficient backup and recovery strategy that meets the business service level agreements becomes an essential task in designing a data warehouse environment.

The IBM Smart Analytics System consists of various components, all of which need to be considered in the backup and recovery strategy in order to provide optimal protection of data in the data warehouse environment. In this section we discuss the backup and recovery plans and strategies for the IBM Smart Analytics System. For further references about backup and recovery strategies in IBM Smart Analytics System environments, see the *IBM Smart Analytics System User's Guide* for your respective version and the best practices documents that are available at IBM developerWorks®:

<http://www.ibm.com/developerworks/data/bestpractices/>

4.6.1 Operating system backup and recovery

In this section we discuss techniques and considerations for backing up and recovering the operating system of the individual nodes in the IBM Smart Analytics System. The methods used vary for AIX and Linux-based systems, as the two platforms have quite unique backup and recovery methods available.

AIX

IBM Smart Analytics System 7600 and 7700 are AIX-based systems. In this section, we discuss AIX backup and recovery techniques.

Backup

For AIX systems, the operating system provides a very mature and reliable backup and recovery method in the **mksysb** utility. This command allows creation of a bootable backup image of an AIX system that can be written to tape, CD, or file. A good way to use **mksysb** in an AIX based IBM Smart Analytics System is to create **mksysb** backup files of each node in the cluster that can then be used to perform a network boot to recover the contents of the node if required.

The management node in an AIX based IBM Smart Analytics System is configured as a network installation manager (NIM) server, and can be used to restore any of the other nodes in the cluster using **mksysb** backup images. For this reason, the good place to store the **mksysb** backups created for the various nodes is on the management node. This activity is done so that they are easily accessible in the event of a restore.

The **mksysb** command must be run on the node being backed up. The image can be written either to a local file then transferred to the management node, or to a remote file system on the management node that is GPFS or NFS mounted locally.

We suggest that a script be written to automate the creation of **mksysb** backup images for all nodes, and scheduled to run regularly (weekly or monthly) through **cron** so that a reasonably up-to-date **mksysb** image for all nodes is always available.

Create a new file system named `/sysbackup` on the management node to accommodate the **mksysb** backup files created on all nodes. Exporting this file system with GPFS or NFS gives an easy way to write **mksysb** files created on each node to a central location on the management node.

The `/etc/exclude.rootvg` file must be created on each server and populated with a list of directories in the root volume group which must not be backed up. Example 4-10 shows a sample `exclude.rootvg` file.

Example 4-10 Sample exclude.rootvg file

```
/sysbackup/  
/tmp/  
/var/tmp/  
/etc/vg/
```

The exclusive file list must be expanded to include any other file systems in your environment that are not required to be included in the **mksysb** backup. As a general guideline, any large or frequently changing file systems in the root volume group will be better backed up to Tivoli Storage Manager and excluded from the **mksysb**.

Example 4-11 shows a simple backup script which can be run from the management node to create a **mksysb** backup on each node and write the backup image to the NFS mounted **/sysbackup** file system on the management node. This script assumes that **ssh** keys are in place to allow passwordless access by root from the management node, which must be the case on most IBM Smart Analytics System. The script also removes **mksysb** backups for each node that are older than 15 days (assuming weekly **mksysb** backups, this will leave at least two copies).

Example 4-11 Backup script

```
#!/bin/ksh  
for NODE in `lsnode`  
do  
    ssh $NODE "mksysb -ie /sysbackup/${NODE}_`date +%d_%b_%Y`.mksysb"  
    # Keep mksysb backups older than 15 days  
    find /sysbackup -name "${NODE}*.mksysb" -mtime +15 -exec rm {} \;  
done
```

This example is very simple and in a production environment has to be expanded to include error handling. For example, the NFS file system is mounted, the directory has sufficient space for the **mksysb**, and the **mksysb** command completes successfully. It is also wise to make the housekeeping command dependent on successful completion of the **mksysb** backup.

The **/sysbackup** file system must be included in your Tivoli Storage Manager file system backups to ensure that offline copies of the **mksysb** backups are available in the event of problems with the management node or its disks.

Restore

Restoring a **mksysb** backup from the management node to one of the nodes entails preparation and configuration work on the Management node, configured as a NIM master, then initiating a network boot on the node to be restored.

The restore steps are as follows:

1. Check if the server is an NIM server.

Use the **lsnim** command to check if the server you are restoring is defined to the NIM server and ready for a NIM operation. Example 4-12 shows a NIM machine status listing.

Example 4-12 Listing a NIM machine status

```
[mgmtnode:root:/home/root:] lsnim -l datanode1
datanode1:
  class          = machines
  type           = standalone
  connect        = nimsh
  platform       = chrp
  netboot_kernel = mp
  if1            = Network-1 datanode1 0
  cable_type1    = bnc
  Cstate         = ready for a NIM operation
  prev_state     = BOS installation has been enabled
  Mstate         = not running
  Cstate_result  = reset
```

The output in Example 4-12 shows that `datanode1` is defined to the NIM master as a machine, and is ready to start an operation (`Cstate = ready for a NIM operation`). If the status was not showing as ready, for example, if a previous NIM operation had failed, then reset the definitions using the following commands:

```
nim -Fo reset datanode1
nim -o deallocate -a subclass=all datanode1
```

These commands reset any current NIM operations for the server and deallocate any resources that might remain assigned to the server. In the event of problems with the NIM restore, always clear any previous restore attempts with these commands before trying again.

2. Define the **mksysb** resource to the NIM server.

Use **smitty** or **run** commands from the command line to define the **mksysb** resource to the NIM server. To define the **mksysb** using **smitty**, use command:

```
smitty nim_mkres
```

This command brings up the Define a Resource panel. Select **mksysb** as the resource type, then fill in a name for the **mksysb** resource, **master** for the server name, the location of the file, and optionally a descriptive comment (see Example 4-13).

Example 4-13 Define a mksysb resource smitty panel

Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
* Resource Name	[datanode1_mksysb]	
* Resource Type	mksysb	
* Server of Resource	[master]	+
* Location of Resource	[/sysbackup/datanode1_15oct2010.mksysb]	/
NFS Client Security Method	<input type="checkbox"/>	+
NFS Version Access	<input type="checkbox"/>	+
Comments	[Mksysb resource for datanode1]	

3. Create SPOT resource.

After the **mksysb** resource has been defined, you have to create an associated shared product object tree (SPOT) resource before the **mksysb** resource can be used for a network boot. Either create a subdirectory of the **/sysbackup** file system to contain the SPOTs, or create a new file system. Use command **smitty nim_mkres** again, and select a resource type of **spot**. Complete the panel by specifying an appropriate name for the spot resource, **master** for the server, and the **mksysb** resource you have just defined as the source for the install images (shown in Example 4-14).

Example 4-14 Define a spot resource smit panel

Define a Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
* Resource Name	[datanode1_spot]	
* Resource Type	spot	
* Server of Resource	[master]	+
Source of Install Images	[nxtsmprod_mksysb]	+
* Location of Resource	[/sysbackup/spot/datanode1_spot]	/
NFS Client Security Method	<input type="checkbox"/>	+
NFS Version Access	<input type="checkbox"/>	+
Expand file systems if space needed?	yes	+
Comments	[Spot created from datanode1 mksysb image]	

4. Set up the NIM master.

To set up the NIM master to enable a network boot using the **mksysb** image, perform these steps:

- a. Run the **smitty** command: **smitty nim_bosinst**.
- b. Select the server to be restored.

- c. Select **mksysb** as the installation type.
 - d. Select the **mksysb** resource you have just defined.
 - e. Select the spot resource you have just defined.
 - f. Change “Initiate reboot and installation now?” to **No**.
 - g. Press Enter to set the network boot up.
5. Boot the LPAR.
- Perform the following steps to boot the LPAR being restored to system management services (SMS), configure the correct client and server IP addresses, and boot from the appropriate network adapter. The LPAR that you are restoring has to be inactive:
- a. Log on to the Hardware Management Console (HMC) web interface as the **hscroot** user
 - b. In the left hand panel, select **Systems Management** → **Servers**, then the correct managed system.
 - c. In the main panel, select the correct logical partition, and in the tasks list next, select **Operations** → **Activate** → **Profile**.
 - d. Check the option to open a console window, and change the boot mode in the advanced options to SMS.
 - e. Select **OK** to start the LPAR booting.
 - f. When the console windows displays the SMS menu, select the option to configure remote IPL, then select the correct network adapter.
 - g. Select **BOOTP** as the network service.
 - h. Configure the client IP address as the management IP address of the LPAR being restored.
 - i. Configure the server IP address as the management IP address of the management node.
 - j. Perform a ping test to ensure that the IP addresses configured are working correctly.
 - k. Click **Escape** until you return to the main SMS menu.
 - l. Select the Select Boot Options item from the menu.
 - m. Select the Select Install or Boot Device item from the menu.
 - n. Select **Network** as the device type.
 - o. Select the network adapter which you configured with an IP address earlier.
 - p. Select **Normal Boot Mode**.
 - q. Confirm your selection to exit and start the network boot.

The LPAR is now performing a network boot from the management node's NIM server. After the LPAR has contacted the NIM server and started booting the **mksysb** image, prompts will appear on the console, which you will need to respond to in order to start the restore:

- When prompted, confirm the console to be used.
 - When prompted, confirm the language to be used during installation.
 - When the “Welcome to Base Operating System Installation and Maintenance” panel appears, select option **2** to change settings.
 - The “System Backup Installation and Setting” pane appears. Confirm the details listed and click **0** to start the installation.
6. The system will now restore the LPAR from the **mksysb** image. After the restore has completed, log on to the LPAR and confirm the restore has completed successfully.

For further information about the process of booting from NIM and performing **mksysb** restores, see the *AIX Installation and Migration Guide*, SC23-6616-04 at the following link

http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.install/doc/insgdrf/insgdrf_pdf.pdf

You can learn the best practice of AIX backup and restore from the developerWork article *Best practices: AIX operating system-level backup and recovery for an IBM Smart Analytics System* at:

<http://www.ibm.com/developerworks/data/bestpractices/smartanalytics/osbackup/index.html>

Linux

For Linux-based IBM Smart Analytics System, the operating system does not provide an equivalent function to the AIX **mksysb**, so the operating system backup and recovery is not as straightforward. There are various options available for performing backup and recovery on a Linux system, each with advantages and disadvantages. Here we discuss the available options, but you need to test the solution you decide upon in your own environment and discuss with your IBM support representative.

Tivoli Storage Manager

IBM Tivoli Storage Manager (TSM) protects and manages a broad range of data, from workstations to the corporate server environment. The centralized, policy-based backup and recovery function is ideal for backup and restore Linux system files.

Backup

Tivoli Storage Manager (TSM) can be used to back up all files on the server, including those belonging to the operating system. Use Tivoli Storage Manager to perform incremental backup of all file systems daily.

Keep database backups separated from file system backups on the Tivoli Storage Manager server by creating two nodes for each server, one for file system backups and one to be used by DB2 to back up the database. The Tivoli Storage Manager client has to be configured such that the database file systems are excluded from the file system backup. You can achieve this task by creating an include/exclude file containing entries for each of the file systems. This file also has to exclude the /db2home and /home file systems if they are mounted over NFS.

Example 4-15 shows a sample include/exclude file.

Example 4-15 Sample Tivoli Storage Manager include/exclude file contents

```
exclude.fs "/db2fs/bcu1inux/NODE0001"  
exclude.fs "/db2fs/bcu1inux/NODE0002"  
exclude.fs "/db2fs/bcu1inux/NODE0003"  
exclude.fs "/db2fs/bcu1inux/NODE0004"  
exclude.fs "/home"  
exclude.fs "/db2home"
```

You can schedule Tivoli Storage Manager backups by creating a schedule on the Tivoli Storage Manager server, and associate the Tivoli Storage Manager file system nodes with the schedule. The clients must then be configured to run the Tivoli Storage Manager Client Acceptor Daemon which runs the local Tivoli Storage Manager scheduler daemon at regular intervals. The local Tivoli Storage Manager scheduler daemon will run the backups at the appropriate times. For details about configuring the Tivoli Storage Manager client acceptor daemon, see the following web page:

http://www-01.ibm.com/support/docview.wss?rs=663&context=SSGSG7&q1=linux+start+dsmcad&uid=swg21240599&loc=en_US&cs=utf-8&lang=en

Restore

Although Tivoli Storage Manager provides good facilities for restoring individual files, performing a “bare metal” restore of an entire server is more problematic. Because Tivoli Storage Manager does not provided any way of booting the server from a backup image, alternative methods have to be used. One option is to create a custom boot CD which includes the Tivoli Storage Manager client, and use it to perform the restore.

Although a full explanation of this activity is beyond the scope of this book, here are a few considerations:

- ▶ Ensure that you have the volume group, logical volume, and file system layout documented in case you need to recreate it.
- ▶ Choose a Linux distribution that allows an easy way to create a custom Live-CD so that you can include the Tivoli Storage Manager client. A guide is available at this address:
<http://www.ibm.com/developerworks/linux/library/l-fedora-livecd/>
- ▶ After you have booted from your live CD, mount your recreated file systems at temporary mount points (for example, under /mnt) and restore each one from Tivoli Storage Manager.
- ▶ If you are recreating your file systems, you must ensure that mount point directories are created at the appropriate places in the restored file systems. This needs to include mount point directories for dynamic file systems such as /proc, /sys, /dev, /dev/pts, because these will not be recreated by the Tivoli Storage Manager restore. You can check the /etc/fstab file which you restore from Tivoli Storage Manager to ensure that mount points have been created for all file systems.
- ▶ Mount the /dev directory from your live CD onto the /dev mount point in the restored root file system (**mount -bind /dev /mnt/dev**).
- ▶ Reinstall the grub boot loader by chroot-ing into the restored file systems (**chroot /target grub-install /dev/sda**).

If considering this as a restore process, the steps must be tested thoroughly and expanded into a full recovery procedure.

Cristie Bare Machine Recovery

Cristie Data Products, which offers data storage and backup solutions, integrates with Tivoli Storage Manager to provide a Bare Machine Recovery (BMR) solution for Linux. The Tivoli BMR product is available for resale through IBM.

All files are backed up to Tivoli Storage Manager, and configuration files are created to reflect the disk layout (partitioning, volume groups, logical volumes, file systems) of the server.

Recovery is performed by booting to a recovery environment provided either on a bootable CD ISO image, or as a network bootable PXE image. Recreation of the disk layout and recovery of the file system data is automated by the Tivoli BMR product. For further information about Tivoli BMR, see this IBM web address:

<http://www-01.ibm.com/software/tivoli/products/storage-mgr/cristie-bmr.html>

Or, go to the Cristie website at this address:

<http://www.cristie.com/products/tbmr/>

4.6.2 Database backup and recovery

The challenge in planning database backup and recovery strategies is to ensure continuous data availability and, in the meantime, secure the data to be ready for recovery in a data loss or corruption situation. The backup and recovery planning must start from the beginning of the project and must consider both the recovery point the recovery time objectives. These objectives must be documented and used as a starting point to design the backup and recovery strategies.

The *recovery time objective (RTO)* is the time expected to recover the database from a data loss or corruption situation. It is set from the beginning of the project. Based on the data volume and the desired recovery time, the infrastructure must be architected to be able to achieve this objective.

The *recovery point objective (RPO)* is the minimum point in time to which data must be recovered. Lost data beyond this point can be re-loaded from source files or be deemed acceptable data loss. This event must be discussed and planned with the application owners and business users. The recovery point objective affects the database backup plan, database transactional log management, and table space design.

Always have the recovery objectives aligned with the infrastructure to ensure that the objectives are reachable. The recovery strategy ties with the data availability and drives the backup plan. Online backup is preferable rather than offline backup because it provides concurrency with other data warehouse operations. You must also consider the granularity of the backup, such as, taking advantage of table space level backup to increase the backup and recovery speed.

To take online backup, the database must have archive logging. When performing major system or database upgrades, it is advisable to do a full offline backup. Unlike a data recovery scenario that might involve the recovery of individual table spaces, a full database recovery is more efficient from a full offline database backup.

You can benefit from IBM Tivoli Storage Manager in managing the database and table space backup images and the transaction logs associated with backup in an IBM Smart Analytics System environment. When using IBM Tivoli Storage Manager, plan the storage (disk) pool size and the number of tape drives per IBM Smart Analytics System data modules according to the backup and recovery time objectives to meet the desired performance and service level agreement.

Tapes: To make the backup and recovery process faster, use two tape drives for each IBM Smart Analytics System data module.

The IBM Smart Analytics System has an option to have extra host bus adapter (HBA) cards to support LAN-free based backup technologies. The LAN-free backup process sends the data directly to a centralized storage device, eliminating the traffic created by backup from the corporate network. The LAN-free backup is conducted through the corporate Storage Attached Network (SAN) to delivery the data directly to the storage and tapes devices.

Use the LAN-free backup technology for IBM Smart Analytics System to optimize the backup and recovery speed, especially when the network is an issue in meeting the backup and recovery objectives. This technology is efficient in handling large data volume such as database backup but not the small file transfer like transaction logs. The transaction log backup can go through the corporate network or have a dedicated network. IBM Tivoli Storage Manager can manage the backup transferred through either LAN-free devices or through the corporate network.

When backing up transaction log files through Tivoli Storage Manager, the storage size must be large enough to accommodate, minimally, the log files from the last backup until the next backup.

More information about planning a backup and recovery strategy, see the following documentation:

- ▶ The DB2 Information Center, at this address:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/c0005945.htm>
- ▶ *DB2 best practices: Building a recovery strategy for an IBM Smart Analytics System data warehouse* at developerWorks:
<http://www.ibm.com/developerworks/data/bestpractices/isasrecovery/index.html>
- ▶ *DB2 best practices: DB2 instance recovery for IBM Smart Analytics System* at developerWorks:
<http://www.ibm.com/developerworks/data/library/techarticle/dm-1010db2instancerecovery/index.html>

Data warehouse database design considerations for recovery

Database design can affect backup performance and recovery efficiency. How data is placed and spread across the table spaces can impact performance of applications loading data, as well as the table space level backup and recovery.

Create only the table spaces required. An excessive number of table spaces can increase complexity and administration tasks. The size of the recovery history files can grow such that starting and stopping the database and creating and dropping existing table spaces becomes very slow.

Consider these possibilities when designing a data warehouse database:

► Classify the data:

One characteristic of data warehouse environment is the large data volume. The update frequency can vary among the data. You can classify your data based on the update frequency into *active*, for frequently updated data; *warm*, for less frequently updated data; and *cold*, for static or historical data. The data is then placed on unique table spaces by these categories. This approach allows you to apply particular backup strategies for each group of table spaces, for example, back up the table spaces with active data more frequently.

► Partition data by range:

You also can distribute data using the DB2 range partition feature and place range partitioned data in unique table spaces. Evenly distributing the data among the database partitions and table partitions improves both data load performance and optimizes recovery capabilities.

When loading data into a range partitioned table, load the data by range to facilitates adding and purging data in the data warehouse and take advantage of the range partitioned table resources.

► Staging tables:

Avoid non-logged data load (by the **load** utility, or non-logged inserts) to the production tables directly. The non-logged load operation places the table space in a backup pending state. Consider performing non-logged load into staging tables, then perform a logged insert to the final table, reading from the staging tables. It will log the transaction (to allow rollforward recovery) and avoid placing the table space in a backup pending state.

The staging tables must be placed in their own table space because the stage tables do not need to be backed up.

► Referential integrity:

Consider placing the tables that have referential integrity relationship in the same table space. All the parent and child tables will be backed up and restored along with the table space. This method can reduce the number of table spaces to be recovered compared to spreading the child tables in unique table spaces.

Backup guidelines

The purpose of backing up data is to ensure data availability including the time taking the backup. Online database and table space backup allows you to perform backup without stopping other database activities. Utility throttling provides the automatic capability to regulate the resource used by the backup job thus minimizing the performance impact on the database.

Here we list guidelines for the database backup of the IBM Smart Analytics System offerings:

- ▶ Perform full database backup on a quarterly basis. Take a full database backup before and after the IBM Smart Analytics System expansion and DB2 software upgrades.
- ▶ Table space level backup takes less time than full database backups. If time is a concern, take table space level backup. Consider archiving inactive data.
- ▶ Perform full online table space backups whenever it is possible. If the active data in the hot table spaces is often updated, take incremental backup. Perform full table space backup (including logs) for the active data at least twice a week.
- ▶ The catalog partition must be backed up on a daily basis to ensure that any DDL issued is synchronized across data node and administration node backups. If the catalog database partition holds a large amount of data, perform incremental backups, but perform full table space backup on a daily basis for the catalog table space.
- ▶ Back up the catalog partition on a daily basis to ensure that any DDL issued is synchronized across data node and administration node backups.
- ▶ Perform a full table space backup when a new table is added. A new table has no data, saving time during recover.
- ▶ The point-in-time recovery of a single table space in an individual database partition is not possible in a partitioned database environment (DPF). To recover a table space to the same point-in-time across all database partitions requires rolling forward to the end of log for all database partitions. If you have to perform a point-in-time recovery, run a full database recovery instead of a table space recovery.
- ▶ When performing database or table space backups on an IBM Smart Analytics System environment, run the backup job in parallel across the data nodes. For example, perform the backup task for the first database partition on each data node in parallel, then when it is finished, start for the second database partition and so on. When you have completed the backup of the first table space, back up the second table space, then the third, until you have backed up all table spaces, and always do one database partition per data node at a time, running in parallel across all data nodes.

- ▶ Back up the sqllib directory on the DB2 instance user home directory using the operating system backup mechanism.
- ▶ Back up the data definition language (DDL) of the production database structure (DDL). Use **db2look** to generate the DDL file after a structure change and save the file. Another good practice is to keep the change history of the instance and database configuration.
- ▶ On the IBM Smart Analytics System environment, there are other databases besides the user database:
 - The IBM InfoSphere Warehouse metadata database (ISWMETA) hosted on the warehouse applications nodes
 - The IBM Cognos content store hosted on the BI nodes

These databases must also be backed up. See the *IBM Smart Analytics System User's Guide* for your respective version for the backup procedures for these metadata databases.

Example 4-16 shows a command for performing an online backup of the catalog database partition.

Example 4-16 Catalog database partition backup

```
# IBM Smart Analytics System 7600 - Full online catalog partition Backup - Partitions 0
db2 backup database edwp on dbpartitionnum(0) online use tsm
```

Example 4-17 shows how to take online table space backup by database partitions in a two-data-node IBM Smart Analytics System 7600 environment. In the example, a backup operation is issued to each node in parallel to help ensure that there is no skew in performance, which will occur if a backup was issued to one node only at a time.

Example 4-17 Table space level online backup

```
# IBM Smart Analytics System 7600 - Tablesapces Backup - Partitions 1 and 5
backup database edwp on dbpartitionnums (1,5) tablespace(pd_ts1, pd_tsl1x)
online use tsm open 2 sessions util_impact_priority 33 include logs
```

```
# IBM Smart Analytics System 7600 - Tablesapces Backup - Partitions 2 and 6
backup database edwp on dbpartitionnums (2,6) tablespace(pd_ts1, pd_tsl1x)
online use tsm open 2 sessions util_impact_priority 33 include logs
```

```
# IBM Smart Analytics System 7600 - Tablesapces Backup - Partitions 3 and 7
backup database edwp on dbpartitionnums (3,7) tablespace(pd_ts1, pd_tsl1x)
online use tsm open 2 sessions util_impact_priority 33 include logs
```

```
# IBM Smart Analytics System 7600 - Tablesapces Backup - Partitions 4 and 8
backup database edwp on dbpartitionnums (4,8) tablespace(pd_ts1, pd_tsl1x)
online use tsm open 2 sessions util_impact_priority 33 include logs
```

Example 4-18 shows how to perform online database backup by database partitions in a two-data-node IBM Smart Analytics System 7600 environment.

Example 4-18 Database level online backup example

```
# IBM Smart Analytics System 7600 - Full online database Backup - Partitions 1 and 5
db2 backup database edwp on dbpartitionnums(1,5) online use tsm

# IBM Smart Analytics System 7600 - Full online database Backup - Partitions 2 and 6
db2 backup database edwp on dbpartitionnums(2,6) online use tsm

# IBM Smart Analytics System 7600 - Full online database Backup - Partitions 3 and 7
db2 backup database edwp on dbpartitionnums(3,7) online use tsm

# IBM Smart Analytics System 7600 - Full online database Backup - Partitions 4 and 8
db2 backup database edwp on dbpartitionnums(4,8) online use tsm
```

When backup with the IBM Tivoli Storage Manager, use the **db2adut1** command to query the backup images and log archive. You also can use **db2adut1** to retrieve log files for restore.

When taking a backup, use the **DB2 list utilities** command to track the progress.

To check the consistency of a backup image, use the **db2ckbkp** command to verify the backup image integrity and to retrieve the backup image information stored in the backup header.

For more information about DB2 backup command and utilities, see the following DB2 documentation:

- ▶ DB2 backup overview:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/c0006150.html>
- ▶ DB2 list utilities command:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011550.html>
- ▶ Checking DB2 backup image:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0002585.html>
- ▶ Managing DB2 objects with Tivoli Storage Manager:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0002077.html>

Recovery guidelines

To have an effective and time saving recovery, it is essential to analyze what is required for recovery, then decide on the recovery scope and steps. A disaster recovery strategy and data recovery strategy must be implemented as separate processes. A database backup is best suited to disaster recovery, whereas a table space backup strategy is best suited to data recovery.

Starting from DB2 version 9.5, you can rebuild the database from table space level backups. No full database backup is required to rebuild the database. Each table space backup image has the entire structure for the database that can be used to restore the entire database structure. After the database structure is restored, you can restore the table spaces in sequence and roll forward to the point of recovery desired.

When performing a full database restore, start from restoring the catalog partition from the latest catalog partition backup, then restore the rest of the database partitions.

Table space recovery works on the individual table space level; it is more granular, flexible, and faster than a full database restore.

DB2 provides tools to help identify the consistency of the data and the database structure. You can use these tools to analyze the database and design a suitable recovery plan when a recovery is required.

Use the **inspect** command to identify where is the corrupted data. The **inspect** command can be performed without deactivate the database. Use the **inspect** command to verify the database architectural integrity and to check the database pages for data consistency. You can save the output to a file and format the results using the **db2inspf** command.

The **db2dart** command is suitable to verify the architectural correctness of the database and the objects within them. **db2dart** accesses the data and metadata in a database by reading them directly from disk, therefore, run this command only when there are no active connections on the database.

After the data corruption scope is identified, you can take a proper recovery action:

- ▶ If the corruption occurred on temporary objects such as the stage tables, no recovery is required.
- ▶ If the corruption was on a table level, recover only the dropped table from the backup.
- ▶ If the corruption occurs on the table space level, a table space restore is sufficient.

- ▶ If the corruption occurs on the database partition level, a full database partition restore is required.
- ▶ If the data error is caused by an application, consider the possibility of unloading and reloading the data using the application.

Example 4-19 shows how to restore a table space across all database partitions.

Example 4-19 Tablespace restore across all database partitions

```
# IBM Smart Analytics System 7600 - Table space restore across all database
partitions

db2_a11 "<<+1<|| db2 \"restore database edwp tablespace (pd_ts1, pd_tslix)
online use TSM taken at <bkp_timestamp> replace existing\"

# Once the restore is completed, perform the roll forward

db2 \"rollforward database edwp to end of logs on dbpartitionnum (1 to 8)
tablespace
(pd_ts1, pd_tslix) online\"
```

For more examples and information about DB2 restore command and utilities, see the following DB2 documentation:

- ▶ DB2 restore overview:
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/c0006237.html>
- ▶ DB2 **inspect** utility:
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0008633.html>
- ▶ DB2 database analysis and report tool (db2dart):
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.trb.doc/doc/c0020760.html>
- ▶ DB2 recover overview:
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0011800.html>
- ▶ Recovering dropped table:
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/t0006318.html>
- ▶ Managing objects with Tivoli Storage Manager (**db2adut1**):
 - <https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0002077.html>



Monitoring tools

In this chapter we discuss methods for providing proactive monitoring for an IBM Smart Analytics System, in particular, how to integrate into an existing IBM Tivoli Monitoring infrastructure.

Proactive monitoring is important for the IBM Smart Analytics System, just as it is for any other critical piece of IT infrastructure, because it helps to provides early notification of any problems, which will allow you to minimize or avoid any outages that will affect your end users.

5.1 Cluster and operating system monitoring

An important aspect in the availability and health of the IBM Smart Analytics System is monitoring the operating system and hardware of the database and application modules in the system. This includes monitoring such items as these:

- ▶ File system utilization
- ▶ Hardware errors on the servers
- ▶ Critical resource utilization problems (for example, CPU, memory)

5.1.1 AIX and Linux

The AIX and Linux based IBM Smart Analytics System offerings vary slightly in monitoring requirements based on their underlying operating systems. For AIX, hardware monitoring can be done both from within the OS, and also from the Hardware Management Console supplied as part of the IBM Smart Analytics System. For Linux, hardware monitoring is done by the service processor included in each System x server.

5.1.2 IBM Systems Director

The IBM Smart Analytics System comes configured with an IBM Systems Director environment on Linux-based offerings that provides monitoring capabilities for the cluster. The default configuration consists of an IBM Systems Director server running on the Management node, and IBM Systems Director agents installed on all other nodes in the cluster.

The IBM Systems Director server communicates directly with the service processor on each of the servers in the IBM Smart Analytics System. This allows monitoring and reporting of any hardware faults that might occur on the servers.

Integrating IBM Systems Director with enterprise monitoring

The IBM Systems Director server provides a number of ways to integrate the hardware monitoring it provides for the IBM Smart Analytics System into your enterprise monitoring system. You can create an Event Automation Plan for your servers that allows you to specify any number of actions when the events of the type and/or severity you specify occur. The actions you can specify include:

- ▶ Send a Tivoli Enterprise Console® event
- ▶ Send an SNMP trap
- ▶ Send an email
- ▶ Update a log file

The most straightforward method for integrating into an IBM Tivoli Monitoring environment is to configure the IBM Systems Director Server to forward a Tivoli Event Console event directly. Other alternatives such as sending an SNMP trap to an IBM Tivoli Netcool® OMNIBus server, or simply writing the event to a log file which you have configured a Tivoli log file adapter to monitor, will also work.

Example of configuring IBM Systems Director

Configuration of the IBM Systems Director server is done by accessing the product's web interface. By default, the web interface uses ports 8421 (for http) or 8422 (for https). Use https if possible to ensure that passwords are not transmitted over the network without encryption. Use the following URL for accessing the IBM Systems Director web interface:

https://management_node_hostname:8422/ibm/console

When you are prompted, login with a user who has been authorized to administer IBM Systems Director by being a member of the *smadmin* group. This is only the root user initially, but it is best to add individual administrators to this group so that you do not need to login directly with the root user.

After you have successfully logged in, you are presented with a panel similar to the one in Figure 5-1.

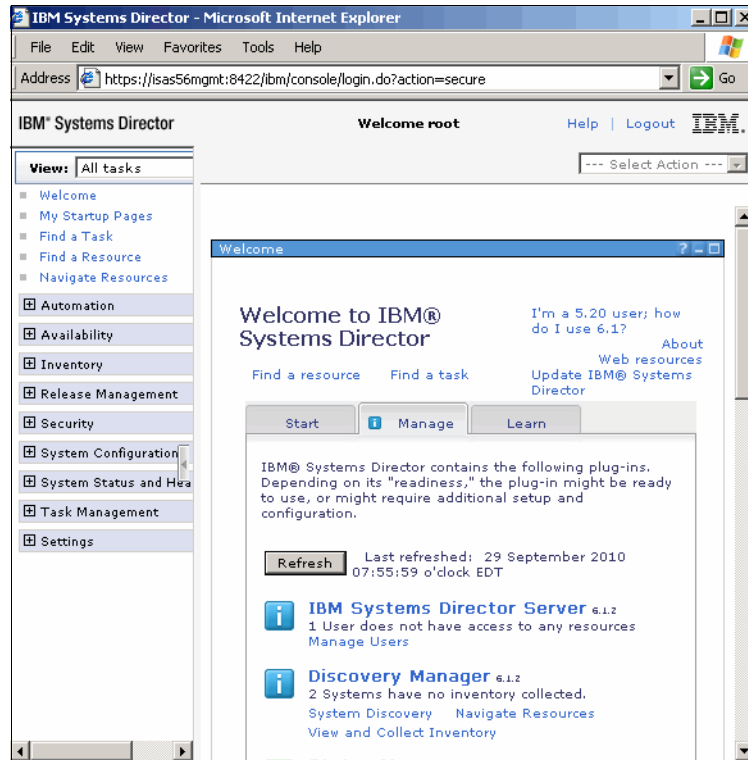


Figure 5-1 Initial IBM Systems Director panel

You can use the *Navigate Resources* link from the task list in the left panel, then the *All Systems* group to explore the resources which have been configured in your environment, and to check that all the servers in your IBM Smart Analytics System are present. You can optionally create custom groups for your servers from this panel if you want to configure unique alerting for various server types. Alternatively, you can just use the pre-configured dynamic *All Systems* group.

You can now create an Event Automation Plan that allows you to forward alerts generated by IBM Systems Director for problems with the servers in the IBM Smart Analytics System to your enterprise monitoring environment. From the task list in the left panel, select the **Automation** menu, then select **Event Automation Plans**. In the main panel, click **Create** to launch the wizard.

Complete the wizard panels as follows:

► Name and Description:

Enter a name for the Event Action Plan, such as `Forward_Alerts`.

► Targets:

Select either one of the custom groups you created for your systems or the default *All Systems* dynamic group, and add it to the Selected Systems list on the right.

► Events:

Configure the type of events you need to forward to your enterprise monitoring environment. One approach is to use the *Event Severity* filter, and include all *Fatal* and *Critical* events. You can also set up CPU, memory, and disk utilization thresholds at which you want to be alerted.

► Event Actions:

Select **Create** to create a new event action. Select from the list the appropriate action to forward the event to your enterprise monitoring. This might be sending a Tivoli Enterprise Console event, sending an SNMP trap or inform request, or even running a custom program on the management node to forward the alert for you.

After you have created your new event action, be sure to use the *Test* button to test that the action works correctly and that your enterprise monitoring receives the alert. Testing an SNMP trap event action will result in an SNMP trap being received on your specified SNMP server, similar to this example:

```
2010-09-29 14:26:33 172.16.10.10(via TCP: [9.26.120.212]:-19469) TRAP, SNMP
v1, community public
  SNMPv2-SMI::enterprises.2.6.159.201.1.3.1 Enterprise Specific Trap (1)
Uptime: 1 day, 12:45:13.43
  SNMPv2-SMI::enterprises.2.6.159.202.1 = STRING: "Director.Test.Action"
SNMPv2-SMI::enterprises.2.6.159.202.2 = STRING: "Informational"
SNMPv2-SMI::enterprises.2.6.159.202.5 = STRING: "An internally generated
event for the purpose of testing the 'Forward alert to central monitoring -
9/29/10 10:26 AM' action configuration."
SNMPv2-SMI::enterprises.2.6.159.202.6 = STRING: "Alert"
```

► Time Range:

Select the time range over which you want the Event Automation Plan to be effective. This is normally the default of *All the time*.

► Summary:

This panel displays a summary of the options configured previously, similar to Figure 5-2.

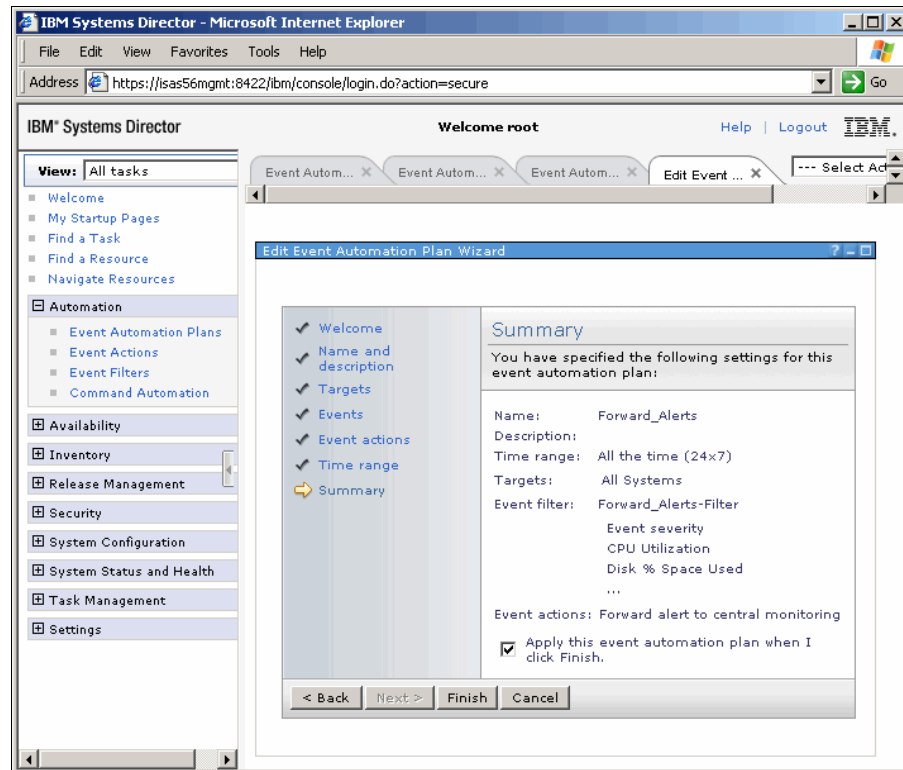


Figure 5-2 Event Automation Plan summary

5.2 DB2 monitoring

The IBM Smart Analytics System consists of many DB2 database partitions and nodes. In this section, we introduce the monitoring utilities DB2 provided and DB2 Performance Expert. These monitoring tools provides the capability to gather information for problem determination, performance tuning, and trend analysis.

5.2.1 DB2 monitoring utilities

The DB2 Database Server provides a comprehensive set of monitor tools to help database administrators manage DB2 instance and both single and multi partition databases.

DB2 snapshot monitor and event monitors

The DB2 snapshot monitor and event monitor are good for performance monitoring. The database administrator can use these two monitor tools to find out why an application receives poor response time or to track an on-going event.

The DB2 snapshot monitor gathers information about the system activity for a specific time. It takes a “picture” of the usage of the database resource usages such as buffer pools, memory, connection activities, statements, and others. You can analyze the snapshots taken for a period time to understand the application behavior and system resource usage trends and take proactive action to maintain a healthy DB2 system.

The database administrator can use the DB2 event monitors to track an event in the database for a period of time. The event monitor records the complete transaction activity and store the information into a file or a table. DB2 provides a set of predefined events for monitoring the server activities, for example, a CONNECTIONS event tracks database connections. You also can generate your own events. An event can be started and stopped anytime. When using an event monitor, limit the information collected to the level needed because event monitors also consume resources.

For further references about DB2 snapshot monitor and event monitors, see the documentation available at the following website:

- ▶ DB2 snapshot monitor:

<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/c0006003.html>

- ▶ DB2 event monitors:

<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/r0005993.html>

DB2 administrative views and table functions

Analyzing information about the DB2 snapshot monitor output for a partitioned database environment can be challenging. DB2 administrative views and table functions provide simple means to gather specific or globally aggregated snapshot data from a specific database partition or all database partitions. DB2 administrative views provide an easy-to-use application programming interface to DB2 administrative functions through SQL.

Example 5-1 shows a sample output of the SNAPAPPL administrative view.

Example 5-1 Gathering snapshot information using SNAPAPPL administrative view

C:\>db2 select AGENT_ID, ROWS_READ, ROWS_WRITTEN, UOW_ELAPSED_TIME_S from sysibmadm.snapappl

AGENT_ID	ROWS_READ	ROWS_WRITTEN	UOW_ELAPSED_TIME_S
131128	0	0	0
75	0	0	0
81	0	0	0
131127	0	0	0
78	0	0	0
77	3	0	0
65590	4	0	0
83	0	0	0
75	5922	0	0
81	0	0	0
80	0	0	0
79	0	0	0
65592	0	0	0
65591	0	0	0
65590	0	0	0
75	0	0	0
81	0	0	0

17 record(s) selected.

Example 5-2 shows a list of the administrative views available in DB2 version 9.7. The views, routines, and procedures with SNAP in their name actually call snapshot under the covers, whereas the MON and WLM routines and procedures do not.

Example 5-2 DB2 version 9.7 administrative views

C:\>db2 list tables for schema SYSIBMADM

Table/View	Schema	Type
ADMINTABCOMPRESSINFO	SYSIBMADM	V
ADMINTABINFO	SYSIBMADM	V
ADMINTEMPCOLUMNS	SYSIBMADM	V
ADMINTEMPTABLES	SYSIBMADM	V
APPLICATIONS	SYSIBMADM	V
APPL_PERFORMANCE	SYSIBMADM	V
AUTHORIZATIONIDS	SYSIBMADM	V
BP_HITRATIO	SYSIBMADM	V
BP_READ_IO	SYSIBMADM	V
BP_WRITE_IO	SYSIBMADM	V
CONTACTGROUPS	SYSIBMADM	V
CONTACTS	SYSIBMADM	V
CONTAINER_UTILIZATION	SYSIBMADM	V
DBCFCG	SYSIBMADM	V
DBMCFG	SYSIBMADM	V
DBPATHS	SYSIBMADM	V
DB_HISTORY	SYSIBMADM	V
ENV_FEATURE_INFO	SYSIBMADM	V
ENV_INST_INFO	SYSIBMADM	V

ENV_PROD_INFO	SYSIBMADM	V
ENV_SYS_INFO	SYSIBMADM	V
ENV_SYS_RESOURCES	SYSIBMADM	V
LOCKS_HELD	SYSIBMADM	V
LOCKWAITS	SYSIBMADM	V
LOG_UTILIZATION	SYSIBMADM	V
LONG_RUNNING_SQL	SYSIBMADM	V
NOTIFICATIONLIST	SYSIBMADM	V
OBJECTOWNERS	SYSIBMADM	V
PDLOGMSG_LAST24HOURS	SYSIBMADM	V
PRIVILEGES	SYSIBMADM	V
QUERY_PREP_COST	SYSIBMADM	V
REG_VARIABLES	SYSIBMADM	V
SNAPAGENT	SYSIBMADM	V
SNAPAGENT_MEMORY_POOL	SYSIBMADM	V
SNAPAPPL	SYSIBMADM	V
SNAPAPPL_INFO	SYSIBMADM	V
SNAPBP	SYSIBMADM	V
SNAPBP_PART	SYSIBMADM	V
SNAPCONTAINER	SYSIBMADM	V
SNAPDB	SYSIBMADM	V
SNAPDBM	SYSIBMADM	V
SNAPDBM_MEMORY_POOL	SYSIBMADM	V
SNAPDB_MEMORY_POOL	SYSIBMADM	V
SNAPDETAILLOG	SYSIBMADM	V
SNAPDYN_SQL	SYSIBMADM	V
SNAPFCM	SYSIBMADM	V
SNAPFCM_PART	SYSIBMADM	V
SNAPHADR	SYSIBMADM	V
SNAPLOCK	SYSIBMADM	V
SNAPLOCKWAIT	SYSIBMADM	V
SNAPSTMT	SYSIBMADM	V
SNAPSTORAGE_PATHS	SYSIBMADM	V
SNAPSUBSECTION	SYSIBMADM	V
SNAPSWITCHES	SYSIBMADM	V
SNAPTAB	SYSIBMADM	V
SNAPTAB_REORG	SYSIBMADM	V
SNAPTbsp	SYSIBMADM	V
SNAPTbsp_PART	SYSIBMADM	V
SNAPTbsp QUIESCER	SYSIBMADM	V
SNAPTbsp_RANGE	SYSIBMADM	V
SNAPUTIL	SYSIBMADM	V
SNAPUTIL_PROGRESS	SYSIBMADM	V
Tbsp_UTILIZATION	SYSIBMADM	V
TOP_DYNAMIC_SQL	SYSIBMADM	V

DB2 relational monitoring interfaces, introduced in DB2 9.7, is an enhanced reporting and monitoring tool that can capture information about database system, data objects, and package cache. DBA can access the interfaces by SQL to quickly identify issues during performance monitoring and problem determination situations. The DB2 relational monitoring interfaces are light-weight, efficient, and have low impact on the system.

The functions available to gather information about system activity, data object level monitoring are as follows:

- ▶ System level:
 - MON_GET_CONNECTION
 - MON_GET_CONNECTION_DETAILS
 - MON_GET_SERVICE_SUBCLASS
 - MON_GET_SERVICE_SUBCLASS_DETAILS
 - MON_GET_UNIT_OF_WORK
 - MON_GET_UNIT_OF_WORK_DETAILS
 - MON_GET_WORKLOAD
 - MON_GET_WORKLOAD_DETAILS
- ▶ Activity level:
 - MON_GET_ACTIVITY_DETAILS
 - MON_GET_PKG_CACHE_STMT
 - MON_GET_PKG_CACHE_STMT_DETAILS¹
- ▶ Data object level:
 - MON_GET_BUFFERPOOL
 - MON_GET_CONTAINER
 - MON_GET_EXTENT_MOVEMENT_STATUS
 - MON_GET_INDEX
 - MON_GET_TABLE
 - MON_GET_TABLESPACE

In 6.3, “DB2 Performance troubleshooting” on page 152, we show examples of using the relational monitoring interfaces to gather database performance metrics.

The following websites provide further information about DB2 administrative views, table functions, and DB2 relational monitoring interfaces:

- ▶ DB2 administrative views and table functions:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.mon.doc/doc/t0010418.html>

<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.rtn.doc/doc/c0022652.html>
- ▶ DB2 relational monitoring interfaces:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.wn.doc/doc/c0055021.html>

¹ Available with DB2 Version 9.7 FixPack 1 and after

DB2 monitor utility: db2top

The **db2top** utility is a monitoring tool distributed with DB2 for Linux and UNIX environments. **db2top** collects DB2 snapshot monitor information cumulatively and gives real time delta values of snapshot metrics. **db2top** is a user friendly tool with interactive text-based GUI interface that provides DBA a better understanding about the metrics. The utility consolidates multiple snapshot options and categorizes the information to make the outputs easy to interpret.

You can access **db2top** in either the interactive mode or batch mode. Using the interactive mode, users can browse between the snapshots options. When running **db2top** in batch mode, you can store the performance information output (in CSV format, for example) and used later for further analysis.

db2top monitors the following snapshot subjects:

- ▶ Database
- ▶ Table space
- ▶ Dynamic SQL
- ▶ Session
- ▶ Buffer pool
- ▶ Lock
- ▶ Table
- ▶ Bottlenecks
- ▶ Utilities
- ▶ Skew monitor (for database partitioned environments)

In Example 5-3 we start **db2top** in interactive mode to monitor database edwp.

Example 5-3 Starting db2top in interactive mode

```
db2top -d edwp
```

In 6.3, “DB2 Performance troubleshooting” on page 152 we show how to monitor performance on IBM Smart Analytics System databases using **db2top**.

Documentation about the **db2top** utility is available at the following websites:

- ▶ DB2 utility tool, **db2top**:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0025222.html>
- ▶ IBM Redbooks publication, *Up and Running with DB2 on Linux*, SG24-6899:
<http://www.redbooks.ibm.com/abstracts/SG246899.html>
- ▶ IBM developerWorks: DB2 problem determination using **db2top** utility:
<http://www.ibm.com/developerworks/data/library/techarticle/dm-0812wang/>

DB2 problem determination command: db2pd

The **db2pd** command is a problem determination tool that collects information without acquiring any latches or using any DB2 engine resources. **db2pd** reads information directly from the memory sets. The **db2pd** command supports partitioned databases and can be used for IBM Smart Analytics System.

Documentation about the **db2top** utility is available at the following websites:

- ▶ DB2 Monitoring and troubleshooting using the **db2pd** command:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.trb.doc/doc/c0054595.html>
- ▶ DB2 **db2pd** command reference:
<https://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011729.html>
- ▶ IBM Redbooks publication, *Up and Running with DB2 on Linux*, sg24-6899:
<http://www.redbooks.ibm.com/abstracts/SG246899.html?Open>

DB2 memory tracker command: db2mtrk

The **db2mtrk** command reports the memory usage and memory pool allocation for DB2 instance and databases. **db2mtrk** is partition-based command, and you can invoke **db2mtrk** from any database partition defined on the db2nodes.cfg file. When the instance level information is returned, the command returns information about the attached database partition only.

Example 5-4 t shows sample **db2mtrk** output about the DB2 instance and single-partition database memory information.

Example 5-4 db2mtrk output for DB2 instance and single partition database

```
C:\Documents and Settings\Administrator>db2mtrk -i -d
Tracking Memory on: 2010/10/18 at 14:22:36
```

Memory for instance

other	monh	fcmbp
37,9M	320,0K	52,8M

Memory for database: SAMPLE

utilh	pckcacheh	other	catcacheh	bph (1)	bph (S32K)
64,0K	192,0K	128,0K	64,0K	2,3M	832,0K
bph (S16K)	bph (S8K)	bph (S4K)	shsorth	lockh	dbh
576,0K	448,0K	384,0K	0	16,6M	22,1M

apph (57) 64,0K	apph (56) 64,0K	apph (55) 64,0K	apph (54) 64,0K	apph (53) 64,0K	apph (51) 64,0K
appshrh 256,0K					

For further information about **db2mtrk**, see the documentation available at the following website:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0008712.html>

5.2.2 DB2 Performance Expert for Linux, UNIX, and Windows

DB2 Performance Expert is part of the IBM Information Management integrated tooling for DB2 on Linux, UNIX, and Windows platforms, and is delivered with IBM Smart Analytics System versions 5600, 7600, and 7700. DB2 Performance Expert tracks database and operating system activities and stores information on its own database to be used later for analysis. You can use DB2 Performance Expert to monitor DB2 on the IBM Smart Analytics System.

DB2 Performance Expert provides four levels of performance monitoring:

- ▶ Online monitoring
- ▶ Short-term history monitoring
- ▶ Long-term history monitoring
- ▶ Exception processing

IBM Optim Performance Manager, a successor of DB2 Performance Expert, significantly extends the database monitoring capabilities provided in DB2 Performance Expert. It introduced a new web-based interface, which significantly simplifies the deployment of the product. The legacy Performance Expert client component is still available with this version, to allow for smoother migration of existing DB2 Performance Expert users. At the time of writing, Optim Performance Manager is not shipped with IBM Smart Analytics System yet. For detail description about IBM Optim Performance Manager, see the IBM Redbooks publication, *IBM Optim Performance Manager for DB2 for Linux, UNIX, and Windows*, SG24-7925. Contact IBM support for migration details.

Online monitoring

Online monitoring is used to monitor the current operation of your DB2 system and the underlying operating system at a point in time when the DB2 instance is being monitored by the DBA sitting in front of the Performance Expert. This can help you gather current DB2 system information, current applications, and SQL workload, and because certain DB2 performance problems are caused by

bottlenecks in the underlying operating system, this information is gathered as well. PE online monitoring features and functions can help you detect problems such as long waits and timeouts, deadlocks, and long running SQL statements.

These features and functions provide the DBA with the ability to drill down to get more detailed information, such as set filters to isolate the problem, customize graphical health charts to visualize how activity and performance evolves over time, trace SQL activities for a single application or the whole database, and view and analyze the trace to identify, for example, heavy hitter SQL statements that need further tuning.

Short-term history monitoring

Short-term history data can provide information to help a DBA look at specific events that occurred in a short interval of time. PE allows the user to configure the number of hours PE stores short-term history information. Using PE short-term history monitoring mode can help a DBA diagnose deadlocks, long running SQL, time-outs, and lock escalations that happened minutes, hours, or days ago without the need to reproduce the problems, and monitor other aspects, such as UOW or buffer pool, table space, and file system usage.

Also, for short-term history data, the graphical health charts can be used to visualize performance metrics over time in history either to diagnose problems or identify trends. For online and short-term monitoring, PE provides the users the ability to see detailed information for the following items:

- ▶ Application Summary/Details:
 - Times
 - Locking
 - SQL activity
 - SQL statements
 - Buffer pools
 - Caching
 - Sorting
 - Memory pools
 - Agents
- ▶ Statistic Details:
 - Instance information
 - Database (usage, caches, high water marks, locks, reads, and writes)
 - Table spaces (Space management, read/write and I/O, and containers)
 - Tables
 - Buffer pool (read, write, I/O, and so on)
 - Memory pools
 - Dynamic SQL statement cache details
 - Utility Information

- ▶ Applications in Lock Conflicts/Locking Conflicts
- ▶ Locking Conflicts
- ▶ System Health: View DB2 performance information in a graphical format
- ▶ System Parameters - Instance
- ▶ System Parameters - Database
- ▶ Operating System Information:
 - Memory and process configuration, processor status
 - File systems
- ▶ Operating System Status:
 - Memory and CPU usage
 - Running processes
 - Disk utilization

Long-term history monitoring

Long-term history data is collected over a period of time. The collected data is used for trend analysis. PE can help you collect trend analysis data that can be used to develop a performance baseline for your system. Using trend analysis data can also help you understand how your system will perform as follows:

- ▶ React during normal and peak periods to help you set realistic performance goals.
- ▶ Resolve potential performance problems before they become an issue.
- ▶ Grow over a period of time.

DB2 PE provides long-term monitoring capability under the following functions:

- ▶ Performance Warehouse and Rules of Thumb:

PE includes Performance Warehouse, which allows you to quickly and easily identify potential performance problems. Performance Warehouse collects performance data for SQL, database, buffer pool activity, and the operating system. This performance data is used for generating reports. These reports can be used for further investigation and trend analysis. Performance Warehouse data can also be used for Rules of Thumb (RoT), which is included in Performance Warehouse.

RoT can help a DBA by being proactive in making suggestions on how to improve performance. Performance Warehouse provides RoT queries for SQL, database, table space, and buffer pool activity.
- ▶ Buffer Pool Analysis:

Buffer pools are one of the most important aspects for tuning. PE Buffer Pool Analysis gathers detailed information regarding current buffer pool activity

using snapshots. Buffer Pool Analysis allows the database administrator to view buffer pool information in a variety of formats, including tables, pie charts, and diagrams. Providing these particular formats to view buffer pool information will enable the database administrator to quickly identify potential problems and do trend analysis.

Exception processing

Exception process monitoring is another PE feature that allows DBA to monitor a database server proactively. DBAs can use the exception processing function to activate predefined alert sets for OLTP or BI workloads or to configure their own alerts both to notify them when a particular situation has occurred. PE provides two types of alerts: deadlock and periodic. The alert message can be sent to specified email addresses or a user exit can be called that allows you to exchange the alert message and details with other applications or to execute actions. The user exits can be used, for example, to send SNMP traps to IBM Tivoli Enterprise Console when a threshold is reached. That way the PE can integrate IBM Smart Analytics System with the existing enterprise monitoring environment.

Additionally, signals on the PE client indicate the occurrence of an exception together with drill down options.

DB2 PE high level architecture overview

The DB2 Performance Expert version 3.2 has two main components: PE Server and PE Client.

- ▶ **PE Server:**

PE Server collects and stores the performance data of the monitored DB2 instance. On the IBM Smart Analytics System environment, PE server is installed on the management node, and it monitors the production instance remotely. The DB2 Performance Expert stores its metadata information and the monitored DB2 instance information collected in the PE database DB2PE. This database is hosted at the management node under bcupe instance.

The PE Server uses DB2 snapshot and event monitors to collect DB2 performance data for the online monitoring, short-term history, long-term history, and exception processing. To reduce overhead on the monitored DB2 instance, PE Server uses DB2 snapshots instead of event monitoring whenever possible.

- ▶ **PE Client:**

PE Client is the user interface of DB2 Performance Expert. It allows you to view the performance data collected by PE Server. PE Client does not communicate with the monitored instance, it always gathers information from the PE Server. You can use the PE Client to configure the PE Server.

The PE Client must be installed on a workstation apart from the IBM Smart Analytics System servers.

For further reference about how to install and configure the PE Client, see the manual *IBM DB2 Performance Expert for Linux, UNIX, and Windows Installation and Configuration*, GC19-2503-02.

Figure 5-3 illustrates a DB2 Performance Expert architecture on the IBM Smart Analytics System.

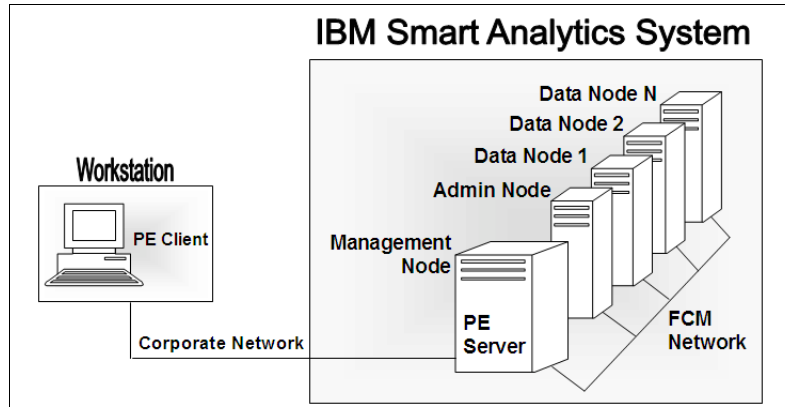


Figure 5-3 DB2 Performance Expert architecture on IBM Smart Analytics System

To start, stop, and check the status of the PE Server, use the following procedures from the management node:

- ▶ To start the DB2 Performance Expert server:
 - a. Log in to the management node as the DB2 Performance Expert user *bcupe*.
 - b. Start the Performance Expert instance:

```
db2start
```
 - c. Issue the following command to start DB2 Performance Expert:

```
pestart
```
- ▶ To stop the DB2 Performance Expert server:
 - a. Log in to the management node as the DB2 Performance Expert user *bcupe*.
 - b. Issue the following command:

```
pestop
```
 - c. Stop the DB2 Performance Expert instance:

```
db2stop
```

- ▶ To determine the status of the DB2 Performance Expert server:

Log in to the management node as the DB2 Performance Expert user *bcupe* and issue the following command:

```
pestatus
```

For further documentation and references about DB2 Performance Expert, check this website:

<http://www-01.ibm.com/software/data/db2imstools/db2tools-library.html#expert>

Naming: The latest version of DB2 Performance Expert is now named IBM Optim Performance Manager. The current versions of the IBM Smart Analytics System are delivered with IBM DB2 Performance Expert V3.2. There are no restrictions to upgrade DB2 PE V3.2 on the IBM Smart Analytics System to IBM Optim Performance Manager.

For further references about IBM Optim Performance Manager, see the website:

<http://www-01.ibm.com/software/data/optim/performance-manager-extended-edition/>

5.3 Storage monitoring

A number of storage devices which need to be considered for monitoring are present in the IBM Smart Analytics System. These include the internal disks on each node used for the OS and the SAN switches used to connect the nodes to the storage subsystems such as DS3400, DS3500, and DS5300.

5.3.1 IBM Remote Support Manager

The most effective method of monitoring for the storage subsystems is to use the IBM Remote Support Manager (RSM) for Storage product. This runs on a dedicated server, and monitors all storage subsystems in the cluster. Any problems are automatically logged with IBM's call management system, with details optionally being emailed to a user configurable address.

The configuration of the Remote Storage Manager can be examined by logging on to the server using its web interface. You need to login with one of the supplied user IDs such as *admin*. Passwords for these IDs can be reset as root from the command line on the RSM server itself using the `rsm-passwd` command.

Figure 5-4 shows the RSM initial panel after being logged in.



Figure 5-4 Initial Remote Support Manager panel

Although the Remote Support Manager must already be configured to report problems to IBM Support, it is best to also configure it to send notification to your system administrators to ensure that you are aware of any problems as soon as they occur.

You can configured this by using, from the main page, **System Configuration** → **Contact Information** to configure your primary contact for the system. You must also configure the "SMTP Server" field in the "Connection Information" page with your local mail relay server, to ensure that the Remote Support Manager is able to notifications through mail to the address you have specified.

5.3.2 Internal disks

Also, you need to monitor internal SAS and SATA disks.

Linux

The internal disks on Linux-based IBM Smart Analytics System environments are monitored by IBM Systems Director, and hardware problems with the disks are logged with IBM Systems Director

AIX

The internal disks on AIX based IBM Smart Analytics System are standard Power Systems disks, and are monitored by the operating system. Any errors with the disks must be logged in the system error log, and can be picked up from there by a standard Tivoli AIX log adapter.

5.3.3 SAN switches

The IBM Smart Analytics System contains a number of SAN switches which are used to connect the individual servers to the storage subsystems. Preferably, monitor these switches.

Although the switches support sending SNMP traps natively, by default they are only connected to the internal cluster network, therefore, it is not possible to send SNMP traps directly to your enterprise monitoring environment. By default, the SAN switches must be configured to send SNMP traps to the IP address of the management node, where they must be captured by IBM Systems Director.

You can confirm this by logging on to the web administration interface for the switch, selecting **Switch Admin** from the menu at the left, selecting **Show Advanced Mode** at the top right, and selecting the **SNMP** tab. The default configuration is similar to that shown in Figure 5-5.

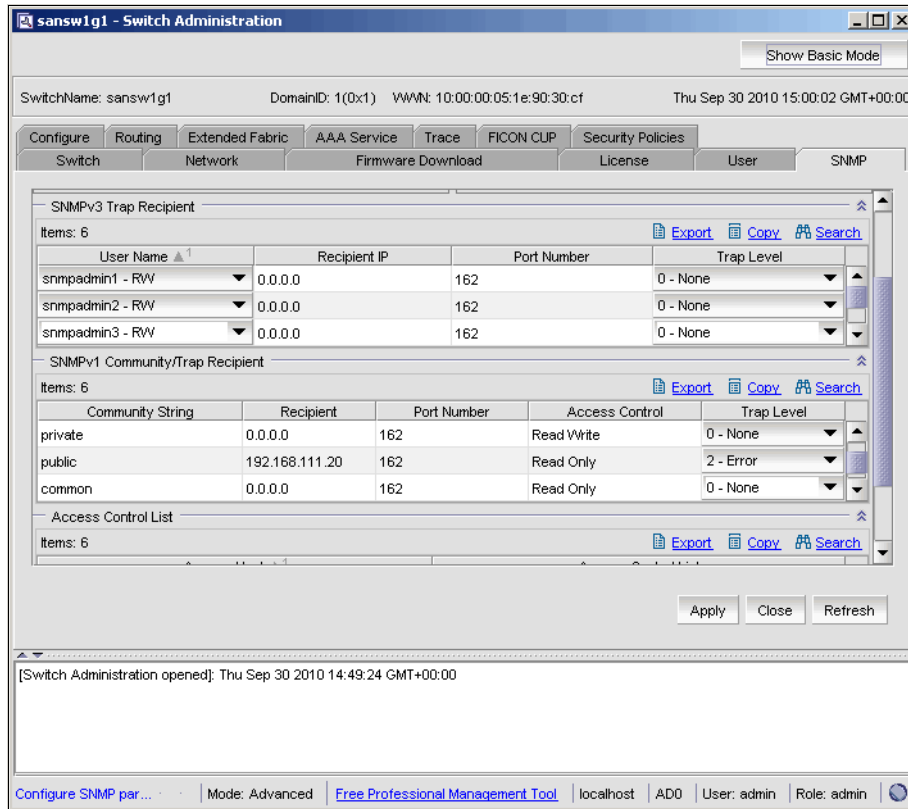


Figure 5-5 Default SNMP configuration for SAN switch

IBM Systems Director can be configured to forward SNMP traps, which will allow the SNMP traps generated by the SAN switches to be picked up by your enterprise alerting infrastructure.

SNMP traps can be forwarded in two ways:

- ▶ Through an Event Action Plan
- ▶ By configuring the SNMPServer.properties file

To configure using an Event Action Plan, go to the Event Action Plan Builder, and select one of the following events, then right-click and select **Customize**:

- ▶ Send an SNMP Trap to a NetView® Host
- ▶ Send an SNMP Trap to an IP Host

5.4 Network monitoring

The IBM Smart Analytics System comes with a number of Ethernet switches that are used to provide the interconnects between the various modules which make up the system. The switches support SNMP, but similar to the SAN switches, their administrative IP interfaces are configured on the internal networks by default. In the same way, this can be worked around by configuring the Ethernet switches to send SNMP traps to the management node, where they can be forwarded to your enterprise monitoring environment.

The Ethernet switches can be configured either using a web interface or using a telnet session. To configure using telnet, perform these steps:

1. Telnet to the switch IP address from one of the IBM Smart Analytics System nodes, and log on (the default user and password for the switches are “admin” and “admin,” though this must be changed).

2. Enter configuration mode using this command:

```
configure
```

3. You can now use the **snmp-server** command to configure SNMP traps.

To get a full list of sub commands, type:

```
snmp-server ?
```

For example, to configure SNMP traps to be sent to the management node at IP address 192.168.111.8 using community string public, use the following commands:

```
snmp-server host 192.168.111.20 public
snmp-server enable traps
```

As for SAN switch monitoring using IBM Systems Director, SNMP traps can be forwarded in two ways:

- ▶ Through an Event Action Plan
- ▶ By configuring the `SNMPServer.properties` file

To configure using an Event Action Plan, go to the Event Action Plan Builder, and select one of the following events, then right-click and select **Customize**:

- ▶ Send an SNMP Trap to a NetView Host
- ▶ Send an SNMP Trap to an IP Host



Performance troubleshooting

In this chapter we introduce performance monitoring on a system with nodes running in parallel, monitoring the system both globally and at the individual node level. We discuss troubleshooting when a performance issue is suspected by using operating system level and database level tools and metrics.

6.1 Global versus local server performance troubleshooting

The IBM Smart Analytics System offering is a DB2 database system, running a shared-nothing set of database nodes in parallel. Any SQL workload is spread across all nodes simultaneously using the “divide and conquer” approach. The resulting work is accomplished in parallel with the results from all of the individual nodes compiled together at the end on the administration node and shipped back to the SQL client.

All SQL workload submitted to the IBM Smart Analytics System is dependent both on the efficient orchestration of the work across all the nodes, as well as on effective execution at the individual node level.

Hence, it is critical to observe performance issues from both a “global” view of the entire set of nodes (servers), as well as from the more traditional view of the individual server’s performance and resource use. Start with the “global” view of system performance and resource use, and drill down, if necessary to the individual node level.

Another important fact to remember when working on performance troubleshooting is that problems and symptoms can be “layered”. You might often notice a performance problem from the “outer layer” of the operating system (OS), and that after identifying which resource is a problem (such as CPU, I/O, and paging), you can drill down a layer for more specifics at the relational database management system (RDBMS) layer (and sometimes further to the application layer).

So a natural progression in the performance troubleshooting of the IBM Smart Analytics System is to incorporate both notions: start viewing global OS resources (looking for type of resource overuse or shortage and comparing “global” verse local use of OS resources); then drill down to the DB2 database layer and try to isolate a specific cause (such as configuration, bad SQL, and too much regular workload).

It is also important to note that, although we are highlighting the notion of a “global” perspective of performance and resource monitoring, all the traditional methods and tools (such as **vmstat**, **iostat**, **top**, **topaz**, **sar**, **uptime**, **db2top**, **db2pd**, **db2mtrk**, DB2 snapshot monitor, and DB2 event monitor) to view performance and monitor resource at the individual server level work. This operates just as on other stand-alone servers and still applies to the IBM Smart Analytics System.

6.1.1 Running performance troubleshooting commands

There are various ways that you can run performance troubleshooting commands on the IBM Smart Analytics System:

- ▶ Running the stand-alone commands directly on specific nodes
- ▶ Running commands in parallel using various utilities built-in with the IBM Smart Analytics System
- ▶ Custom methods in more complex situations

Running commands directly on each physical node

The simplest method is to run any of the standard performance commands directly on each node of interest. To obtain a complete picture of the whole system, you need to run the command separately, directly on each node one at a time. This method can be impractical, especially if you have many nodes.

Example 6-1 shows this direct “one-node-at-a-time” method of executing the **uptime** command on three nodes. It is easy to see that this method might be too cumbersome on any greater number of nodes.

Example 6-1 Running uptime separately on multiple nodes

```
ISAS56MGMT:~ # ssh ISAS56R1D1
Last login: Wed Oct 6 12:11:26 2010 from isas56mgmt
ISAS56R1D1:~ # uptime
 12:13pm up 50 days 2:57,  2 users,  load average: 0.96, 0.47, 0.17
ISAS56R1D1:~ # exit
logout
Connection to ISAS56R1D1 closed.
ISAS56MGMT:~ # ssh ISAS56R1D2
Last login: Wed Oct 6 12:12:48 2010 from isas56mgmt
ISAS56R1D2:~ # uptime
 12:13pm up 50 days 3:06,  1 user,  load average: 0.97, 0.51, 0.20
ISAS56R1D2:~ # exit
logout
Connection to ISAS56R1D2 closed.
ISAS56MGMT:~ # ssh ISAS56R1D3
Last login: Wed Oct 6 05:16:38 2010 from isas56mgmt
ISAS56R1D3:~ # uptime
 12:13pm up 50 days 3:06,  1 user,  load average: 0.90, 0.57, 0.24
ISAS56R1D3:~ # exit
logout
Connection to ISAS56R1D3 closed.
```

Running commands across multiple physical nodes in parallel using dsh

In addition to using the commands directly on each node in a serial fashion, you can also use the **dsh** utility on the management node to run one or more

commands across all (or a chosen subset) of the physical nodes. You must be the root system administration user, and **dsh** is available on AIX based IBM Smart Analytics system environments 7600 and 7700. The IBM Smart Analytics System 5600 does not include **dsh** by default. However, you can choose to download and install it.

Example 6-2 on page 130 shows that **dsh** executes commands once per physical node for all physical nodes. Figure 6-2 also shows that the command executes in parallel (not serially) as the **date** command returns timestamps that are identical, unlike the case if they had been launched serially.

Figure 6-2 also shows that the output returned by **dsh** is not in the order launched, but rather in the order that the commands completed on the nodes (output for node ISAS56R1D5 comes before node ISAS56R1D1, and output for nodes ISAS56R1D3 and ISAS56R1D4 come before node ISAS56R1D2).

Example 6-2 dsh launches command in parallel across all nodes chosen

```
ISAS56MGMT:~ # dsh -a "sleep 5;echo `date`" : IP addr ==> `hostname -i`"
ISAS56R1D5: Sat Oct 9 18:21:28 EST 2010 : IP addr ==> 172.16.10.10
ISAS56R1D1: Sat Oct 9 18:21:28 EST 2010 : IP addr ==> 172.16.10.10
ISAS56R1D3: Sat Oct 9 18:21:28 EST 2010 : IP addr ==> 172.16.10.10
ISAS56R1D4: Sat Oct 9 18:21:28 EST 2010 : IP addr ==> 172.16.10.10
ISAS56R1D2: Sat Oct 9 18:21:28 EST 2010 : IP addr ==> 172.16.10.10
```

Example 6-3 shows how to use **dsh** to run the performance command **uptime** across all physical database nodes and display the output in node name order using the UNIX **sort** command.

Example 6-3 Using dsh to run the 'uptime' command across all nodes.

```
ISAS56MGMT:~ # dsh -a uptime | sort
ISAS56R1D1: 11:24pm up 52 days 14:09, 3 users, load average: 0.01, 0.02, 0.00
ISAS56R1D2: 11:24pm up 52 days 14:17, 0 users, load average: 0.00, 0.00, 0.00
ISAS56R1D3: 11:24pm up 52 days 14:17, 1 user, load average: 0.00, 0.00, 0.82
ISAS56R1D4: 11:24pm up 52 days 14:17, 0 users, load average: 0.00, 0.00, 0.00
ISAS56R1D5: 11:24pm up 52 days 14:17, 0 users, load average: 0.11, 0.03, 0.01
```

Running commands across multiple physical nodes serially using **rah**

The **rah** utility executes your command across all physical database nodes serially one at a time, and returns the results in order. The **rah** utility is perfect for obtaining information that is “physical-node” oriented. You can run this utility from the administration node as the DB2 instance owner user ID.

Example 6-4 demonstrates how to use the **rah** utility to check the **db2sysc** UNIX processes running on all physical nodes. because UNIX processes are created and managed at the physical UNIX node level, **rah** is the proper tool to use to check physical-level UNIX process information across all physical UNIX nodes.

Example 6-4 Using rah to check UNIX process db2sysc

```
bculinux@ISAS56R1D1:~> rah 'ps aux | grep db2sysc | grep -v grep'

bculinux 5999 0.7 0.9 7098148 639640 ? S1 23:42 0:00 db2sysc 0
ISAS56R1D1: ps aux | grep db2sysc ... completed ok

bculinux 31800 0.7 1.0 9193208 683176 ? S1 23:42 0:00 db2sysc 1
bculinux 31813 0.6 0.8 9193212 581936 ? S1 23:42 0:00 db2sysc 2
bculinux 31836 0.6 0.8 9193208 581828 ? S1 23:42 0:00 db2sysc 3
bculinux 31846 0.6 0.8 9193212 581908 ? S1 23:42 0:00 db2sysc 4
ISAS56R1D2: ps aux | grep db2sysc ... completed ok

bculinux 28802 0.7 1.0 9193208 683180 ? S1 23:42 0:00 db2sysc 5
bculinux 28812 0.6 0.8 9193212 581932 ? S1 23:42 0:00 db2sysc 6
bculinux 28822 0.6 0.8 9193212 581836 ? S1 23:42 0:00 db2sysc 7
bculinux 28845 0.6 0.8 9193208 581904 ? S1 23:42 0:00 db2sysc 8
ISAS56R1D3: ps aux | grep db2sysc ... completed ok
```

Because **rah** is a physical node oriented utility, running commands across all physical database nodes as opposed to across all logical database partitions, using this utility with commands meant to be used on a logical database partition level will yield incomplete results. The command **rah** only executes on the first logical database partition of a given physical node.

Example 6-5 shows that **rah** is not intended for running the logical database partition level commands such as **db2 list active databases**. The IBM Smart Analytics System 5600 has four database partitions (logical database partitions) per physical node. When checking on all active partitions, the expected output is four partitions on each regular database partition plus one active one for the administration node. However, the output shows only one active database per physical node. Node ISAS56R1D2 shows only one out of the expected four active database partitions.

Example 6-5 Improper use of rah: checking logical node database information

```
bculinux@ISAS56R1D1:~> rah 'db2 list active databases'

list active databases

Active Databases

Database name = BCUKIT
Applications connected currently = 0
Database path = /db2fs/bculinux/NODE0000/SQL00001/

ISAS56R1D1: db2 list active databases completed ok

list active databases

Active Databases

Database name = BCUKIT
Applications connected currently = 0
Database path = /db2fs/bculinux/NODE0001/SQL00001/
```

```
ISAS56R1D2: db2 list active databases completed ok
```

```
list active databases
```

```
Active Databases
```

```
Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0005/SQL00001/
```

```
ISAS56R1D3: db2 list active databases completed ok
```

Running commands across multiple logical database partitions serially using db2_all

The **db2_all** utility executes your command across all logical database partitions serially one at a time, and returns the results in order. The **db2_all** utility is perfect for obtaining information that is “logical-node” oriented. You can run this utility from the administration node as the DB2 instance owner user ID.

Example 6-6 demonstrates using **db2_all** to run the logical database partition level commands **db2 list active databases**. The output shows the expected four database partitions per physical node.

Example 6-6 Proper use of db2_all: checking logical node db information

```
bcunix@ISAS56R1D1:~> db2_all 'db2 list active databases'
```

```
list active databases
```

```
Active Databases
```

```
Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0000/SQL00001/
```

```
ISAS56R1D1: db2 list active databases completed ok
```

```
list active databases
```

```
Active Databases
```

```
Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0001/SQL00001/
```

```
ISAS56R1D2: db2 list active databases completed ok
```

```
list active databases
```

```
Active Databases
```

```
Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0002/SQL00001/
```

```

ISAS56R1D2: db2 list active databases completed ok
list active databases

      Active Databases

Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0003/SQL00001/

ISAS56R1D2: db2 list active databases completed ok
list active databases

      Active Databases

Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0004/SQL00001/

ISAS56R1D2: db2 list active databases completed ok
list active databases

      Active Databases

Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0005/SQL00001/

ISAS56R1D3: db2 list active databases completed ok
list active databases

      Active Databases

Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0006/SQL00001/

ISAS56R1D3: db2 list active databases completed ok
list active databases

      Active Databases

Database name           = BCUKIT
Applications connected currently = 0
Database path           = /db2fs/bcunix/NODE0007/SQL00001/

ISAS56R1D3: db2 list active databases completed ok

```

Because **db2_a11** runs on all logical nodes and there can be multiple logical database partitions per physical node, when you use **db2_a11** to check any physical node level information, the result can be misleading.

Example 6-7 shows the output when using **db2_a11** to check the **db2sysc** UNIX processes on all the nodes. The output has duplicate process information, for example, the PID 27947 information is repeated four times. This is because **db2_a11** ran the command four times on the same physical node, once for each of the four logical database partitions.

Example 6-7 Improper use of db2_a11: checking UNIX processes db2sysc

```

bcu linux@ISAS56R1D1:~> db2_a11 'ps -ef | grep db2sysc | grep -v grep'

bcu linux 9553 9550 0 Oct05 ?          00:00:30 db2sysc 0
ISAS56R1D1: ps -ef | grep db2sysc ... completed ok

bcu linux 27947 27942 78 Oct05 ?       12:11:37 db2sysc 1
bcu linux 27961 27955 76 Oct05 ?       11:56:54 db2sysc 2
bcu linux 28391 28282 75 Oct05 ?       11:41:54 db2sysc 3
bcu linux 28401 28399 74 Oct05 ?       11:33:56 db2sysc 4
ISAS56R1D2: ps -ef | grep db2sysc ... completed ok

bcu linux 27947 27942 78 Oct05 ?       12:11:37 db2sysc 1
bcu linux 27961 27955 76 Oct05 ?       11:56:54 db2sysc 2
bcu linux 28391 28282 75 Oct05 ?       11:41:54 db2sysc 3
bcu linux 28401 28399 74 Oct05 ?       11:33:56 db2sysc 4
ISAS56R1D2: ps -ef | grep db2sysc ... completed ok

bcu linux 27947 27942 78 Oct05 ?       12:11:37 db2sysc 1
bcu linux 27961 27955 76 Oct05 ?       11:56:54 db2sysc 2
bcu linux 28391 28282 75 Oct05 ?       11:41:54 db2sysc 3
bcu linux 28401 28399 74 Oct05 ?       11:33:56 db2sysc 4
ISAS56R1D2: ps -ef | grep db2sysc ... completed ok

bcu linux 27947 27942 78 Oct05 ?       12:11:37 db2sysc 1
bcu linux 27961 27955 76 Oct05 ?       11:56:54 db2sysc 2
bcu linux 28391 28282 75 Oct05 ?       11:41:54 db2sysc 3
bcu linux 28401 28399 74 Oct05 ?       11:33:56 db2sysc 4
ISAS56R1D2: ps -ef | grep db2sysc ... completed ok

bcu linux 1378 1374 70 Oct05 ?          11:03:13 db2sysc 5
bcu linux 1383 1376 73 Oct05 ?          11:26:23 db2sysc 6
bcu linux 1394 1392 73 Oct05 ?          11:24:53 db2sysc 7
bcu linux 1417 1415 72 Oct05 ?          11:17:53 db2sysc 8
ISAS56R1D3: ps -ef | grep db2sysc ... completed ok

bcu linux 1378 1374 70 Oct05 ?          11:03:13 db2sysc 5
bcu linux 1383 1376 73 Oct05 ?          11:26:23 db2sysc 6
bcu linux 1394 1392 73 Oct05 ?          11:24:53 db2sysc 7
bcu linux 1417 1415 72 Oct05 ?          11:17:53 db2sysc 8
ISAS56R1D3: ps -ef | grep db2sysc ... completed ok

bcu linux 1378 1374 70 Oct05 ?          11:03:13 db2sysc 5
bcu linux 1383 1376 73 Oct05 ?          11:26:23 db2sysc 6
bcu linux 1394 1392 73 Oct05 ?          11:24:53 db2sysc 7
bcu linux 1417 1415 72 Oct05 ?          11:17:53 db2sysc 8
ISAS56R1D3: ps -ef | grep db2sysc ... completed ok

```

6.1.2 Formatting the command output

When there are multiple lines of output per node involved, the output displayed can be “busy” and cannot be sorted easily in alphanumeric order by node name.

Figure 6-8 shows a standard `vmstat` output using the `dsh` utility. The `vmstat` command reports on each node but not in sequence. For example, node `ISAS56R1D5` is reported on before `ISAS56R1D4`, and `ISAS56R1D2` is reported between `ISAS56R1D4` and `ISAS56RD3`. The individual `vmstat` report fields do not line up well with one another and the headers are repeated for every two lines of output, which adds to the amount of unnecessary characters on the screen. Most important of all, for every useful line per node of information, `dsh` displays four total lines per node on the screen. This prevents you from seeing all the nodes on the screen when you have many nodes on your system.

Example 6-8 Standard `vmstat` output using the `dsh` utility

```
ISAS56MGMT: # dsh -a 'vmstat 1 2'
ISAS56R1D5: procs -----memory----- --swap-- ----io---- -system-- ----cpu-----
ISAS56R1D5: r b swpd free buff cache si so bi bo in cs us sy id wa st
ISAS56R1D5: 0 0 148 61028836 3272800 403180 0 0 367 106 0 0 1 0 98 1 0
ISAS56R1D5: 0 0 148 61029012 3272800 403180 0 0 0 0 284 670 0 0 100 0 0
ISAS56R1D4: procs -----memory----- --swap-- ----io---- -system-- ----cpu-----
ISAS56R1D4: r b swpd free buff cache si so bi bo in cs us sy id wa st
ISAS56R1D4: 0 0 0 61516076 2933116 352372 0 0 369 105 0 0 1 0 98 1 0
ISAS56R1D4: 0 0 0 61516240 2933116 352372 0 0 0 12 285 678 0 0 100 0 0
ISAS56R1D2: procs -----memory----- --swap-- ----io---- -system-- ----cpu-----
ISAS56R1D2: r b swpd free buff cache si so bi bo in cs us sy id wa st
ISAS56R1D2: 9 69 0 16172928 1804884 41932404 0 0 1241 246 0 1 2 0 95 3 0
ISAS56R1D2: 13 61 0 16174512 1804884 41932404 0 0 487232 29264 35347 155873 52 9 1 38 0
ISAS56R1D3: procs -----memory----- --swap-- ----io---- -system-- ----cpu-----
ISAS56R1D3: r b swpd free buff cache si so bi bo in cs us sy id wa st
ISAS56R1D3: 3 74 0 43792620 1634324 15122076 0 0 1216 244 0 0 2 0 95 3 0
ISAS56R1D3: 17 57 0 43794452 1634324 15122076 0 0 420624 14416 31996 149589 38 6 1 55 0
```

Formatting the output can help spot the problem when running performance troubleshooting commands in parallel. Example 6-9 shows an example of formatting the `vmstat` command output. The command is saved as an alias `savmstat` for convenience which can be rerun later.

Example 6-9 Saving a more complex command as a reusable alias command

```
ISAS56MGMT:~ # vi .bash_profile (add alias command to end of file and save)
...
alias savmstat="echo '          '\`vmstat 1 | head -1`;dsh -a 'vmstat 1 2 | tail -1' | sort"

ISAS56MGMT:~ # savmstat
procs -----memory----- --swap-- ----io---- -system-- ----cpu-----
ISAS56R1D2: 1 0 0 20396720 1869112 41451836 0 0 0 36 292 747 0 0 100 0 0
ISAS56R1D3: 0 0 0 47966484 1697744 14776984 0 0 0 0 273 748 0 0 100 0 0
ISAS56R1D4: 0 0 0 61512632 2934180 353364 0 0 0 0 284 661 0 0 100 0 0
ISAS56R1D5: 0 0 148 61029704 3273744 402236 0 0 0 0 261 603 0 0 100 0 0
```

For more complex output formatting, filtering, reordering, summarizing the information at top of screen, and merging output of two or more commands, you can use custom script. As an example, we provide a Perl script, `sa_cpu_mon.pl`, that formats the `vmstat` output for monitoring CPU resource. This script combines the relevant CPU-related elements from both the `vmstat` and `uptime` commands, computes system-wide averages, and displays the results in node name order for all physical nodes. You can use this Perl script for both Linux- and AIX-based IBM Smart Analytics System offerings.

Example 6-10 shows a sample output of `sa_cpu_mon` with CPU performance from a system-wide glance at all nodes in parallel and system average statistics summarized at the top. The information is parsed and reformatted into an easy-to-view display.

Example 6-10 Script formatted `vmstat` with load averages

```
ISAS56MGMT:~ ./sa_cpu_mon.pl
sa_cpu_mon
```

	Run Queue	Block Queue	CPU				Load Average		
			usr	sys	idle	wio	1min	5mins	15mins
System Avg:	1.2	2.2	4.6	0.4	88.4	6.6	4.39	7.46	9.39
ISAS56R1D1:	0.0	0.0	0.0	0.0	100.0	0.0	0.08	0.07	0.01
ISAS56R1D2:	4.0	5.0	11.0	1.0	78.0	10.0	10.86	19.25	23.92
ISAS56R1D3:	2.0	6.0	12.0	1.0	64.0	23.0	10.98	17.96	23.00
ISAS56R1D4:	0.0	0.0	0.0	0.0	100.0	0.0	0.02	0.01	0.00
ISAS56R1D5:	0.0	0.0	0.0	0.0	100.0	0.0	0.00	0.00	0.00

The following columns are shown:

- ▶ Run Queue: The count of all processes ready to run, running, or waiting to run on an available CPU.
- ▶ Block Queue: All processes waiting for data to be returned from I/O before they can resume work.
- ▶ CPU: The standard four `vmstat` CPU columns:
 - `usr` - The percentage of user (application) CPU usage.
 - `sys` - The percentage of system (running UNIX kernel code) CPU usage.
 - `idle` - The percentage of unused CPU.
 - `wio` - The percentage waiting on an IO operation to complete.
- ▶ Load Average: The run queue plus block queue columns of the `uptime` command (the running average load for the past 1 minute, 5 minutes and 15 minutes). That is, the count of all running or runnable tasks plus the count of all tasks waiting on I/O.

To have a parallel view of I/O resource usage across all the nodes of the IBM Smart Analytics System, we provide a Perl script, `a_cpu_mon.pl`. This script pulls relevant I/O-related elements from both the `iostat` and the `/proc/stat` system file, computes system-wide averages, and displays the results in node name order for all physical nodes.

Another Perl script that we provide for monitoring the memory paging activity is `sa_paging_mon.pl`. This script combines all relevant memory and swap space resources and activities.

For the source code of these scripts, see Appendix A, “Smart Analytics global performance monitoring scripts” on page 281.

6.2 Performance troubleshooting at the operating system level

When conducting performance troubleshooting, you must first take an overall glance at the system resources being used across the data nodes and administrations that carry the SQL workload.

In this section we discuss the performance troubleshooting methods and tools to analyze resource issues at the operating system layer.

6.2.1 CPU, run queue, and load average monitoring

In this section we go through the process of troubleshooting the CPU resource issues across the system, such as high CPU usage, high number of processes in the run queue, and the higher-than-normal uptime load averages. To troubleshoot CPU resources issues on the IBM Smart Analytics System, check the resource from the entire system level, then on the node level, and drill down to the process level.

Checking on the global system level

Start off by checking the CPU-related resources across the entire IBM Smart Analytics System to find the nodes that consume the most CPU resources.

Here we use the custom tool `sa_cpu_mon` to check CPU-related resource consumption on all nodes:

```
$ ./sa_cpu_mon.pl
```

Example 6-11 shows a snapshot of the CPU usage across all the nodes of an IBM Smart Analytics System.

Example 6-11 identifying CPU “hot spots” at the node level

```
# ./sa_cpu_mon.pl
sa_cpu_mon
```

	Run		Block		----- CPU -----				----- Load Average -----		
	Queue	Queue	usr	sys	idle	wio	1min	5mins	15mins		
System Avg:	26.6	3.2	20.0	0.2	79.4	0.4	22.84	9.62	6.03		
ISAS56R1D1:	0.0	0.0	0.0	0.0	100.0	0.0	0.24	0.53	0.45		
ISAS56R1D2:	133.0	16.0	99.0	1.0	0.0	0.0	112.23	45.38	27.56		
ISAS56R1D3:	0.0	0.0	1.0	0.0	97.0	2.0	1.70	2.16	2.14		
ISAS56R1D4:	0.0	0.0	0.0	0.0	100.0	0.0	0.00	0.00	0.00		
ISAS56R1D5:	0.0	0.0	0.0	0.0	100.0	0.0	0.01	0.01	0.00		

In this example, the node ISAS56R1D2 stands out amongst all the nodes as the greatest consumer of CPU resources in the entire system. The user application takes up 99% CPU time. Because there are 16 CPUs per node, the high run queue number means that 117 processes were waiting on an available CPU to run on. The high load averages tell us that this is not just a “spike” in the run queue, but rather that it has been very high for at least the past minute, and also appears to have been higher than the other nodes for at least the past 15 minutes.

Also, the load average appears to have been higher in average for the past 15 minutes, higher in average of the past 5 minutes, and higher still in average of the past minute. This information seems to indicate that the workload on the system has been rising, and the trend indicates that it might continue to do so, so we might want to check this out further before it becomes a bigger issue.

Checking the node level

After identifying the nodes that have a CPU usage problem, try to isolate which process cause the problem using the **ps** command. The following Linux and AIX **ps** command shows the current top 10 CPU consumers:

```
$ ps aux | head -1;ps aux | sort -nr -k3 | head -10
```

In our example, the node ISAS56R1D2 appears to be a “hot node” with higher workload and higher CPU resource usage. Example 6-12 shows the top 10 CPU consumers on the ISAS56R1D2 node.

Example 6-12 Identify the current top 10 CPU consumers using ps

```
ISAS56R1D2:~ # ps aux | head -1;ps aux | sort -nr -k3 | head -10
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
bculinux	28068	643	5.7	13931224	3820832	?	S1	04:34	17:03	db2sysc 2
bculinux	28101	607	5.8	13932380	3830956	?	S1	04:34	16:00	db2sysc 4
bculinux	28078	3.8	5.6	13652636	3695396	?	S1	04:34	0:06	db2sysc 3
bculinux	28058	3.8	5.6	13655708	3743236	?	S1	04:34	0:06	db2sysc 1
root	8524	2.0	0.7	503004	466864	?	Ss1	Aug17	1526:15	/opt/ibm/director/agent/...
root	2438	0.8	0.0	0	0	?	S<	Aug17	631:57	[mpp_dcr]
root	30347	0.6	0.0	9300	1732	?	Ss	04:37	0:00	bash -c echo `hostname`: `vmstat 5 2

```
| tail -1 `uptime`
```



```

root      7849  0.5  0.0      0      0 ?      RN   Aug17 441:24 [kipmi0]
root      30345  0.3  0.0 41872 2920 ?      Ss   04:37  0:00 sshd: root@notty
root      8054  0.2  0.0 224260 20392 ?      S1   Aug17 221:12 ./jre/bin/java -Djava.compiler=NONE
-cp /usr/RaidMan/RaidMsgExt.jar:/usr/RaidMan/RaidMan.jar com.ibm.sysmgmt.raidmgr.agent.ManagementAgent

```

The UNIX process ID # 28068 is the highest current consumer of CPU resources at 643% CPU, the equivalent of 6.43 CPUs (100% CPU = 1 CPU) out of a total of 16 CPUs available on this node. The process ID 28101 is a close second at 607% CPU, equivalent of 6.07 CPUs.

Process ID 28068 shows a command of **db2sysc 2**. In this command, **db2sysc** is the main DB2 engine process, and the number **2** next to it tells us that it is the main DB2 process for DB2 logical database partition #2. The command for the other high-CPU consuming PID# 28101 is **db2sysc 4**, indicating it is the main DB2 engine process for DB2 logical database partition #4. Because the CPU usage of **db2sysc 3** (PID# 28078) and **db2sysc 1** (PID# 28058) is very low and there is no CPU usage on other physical nodes, this seems to indicate that all the SQL activity is concentrated on logical partitions 2 and 4 exclusively. This situation might potentially indicate a data skew issue, with the data for a specific table being concentrated on very few database partitions instead of being spread out evenly across all the database partitions.

Alternatively, you can use the **top** (Linux) or **topaz** (AIX) commands to list the top current CPU consumers.

Example 6-13 show the output of the **top** command that confirms what we discovered in the Example 6-12 on page 138, that is, the process ID 28068 and 28101 represent the lion's share of the CPU resource consumption.

Note that the **COMMAND** field of the **top** output does not provide the logical partition number. You cannot tell which logical database partition these **db2sysc** DB2 engine processes are associated with. In this case, use the **ps** command.

Example 6-13 Using top to identify the top CPU consuming processes

```

top - 04:39:36 up 52 days, 19:32,  4 users,  load average: 104.22, 59.18, 24.40
Tasks: 260 total,  11 running, 249 sleeping,   0 stopped,   0 zombie
Cpu(s): 99.3%us,  0.3%sy,  0.0%ni,  0.0%id,  0.1%wa,  0.0%hi,  0.3%si,  0.0%st
Mem:  65981668k total, 37876324k used, 28105344k free, 1727960k buffers
Swap: 33559744k total,      0k used, 33559744k free, 34399044k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
28068	bcu	linux	20	0	13.3g	3.8g	3.8g	S	1302	6.1	38:15.31	db2sysc
28101	bcu	linux	25	0	13.3g	3.8g	3.8g	S	290	6.1	30:53.33	db2sysc
28058	bcu	linux	24	0	13.0g	3.6g	3.5g	S	2	5.7	0:08.61	db2sysc
28078	bcu	linux	25	0	13.0g	3.6g	3.5g	S	2	5.7	0:08.58	db2sysc
2438	root		0	-20	0	0	0	S	1	0.0	631:58.50	mpp_dcr
8054	root		15	0	219m	19m	8552	S	1	0.0	221:12.68	java
1	root		16	0	796	308	256	S	0	0.0	0:13.89	init
2	root		RT	0	0	0	0	S	0	0.0	0:00.29	migration/0
3	root		34	19	0	0	0	S	0	0.0	1:50.35	ksoftirqd/0
4	root		RT	0	0	0	0	S	0	0.0	0:00.21	migration/1

Another method for listing top 10 processes consuming CPU resources is **pidstat**. This command is for Linux only:

```
$ pidstat | head -3; pidstat 2 1 | egrep -v -i 'average|Linux' | sort -nr -k 5 | head
```

Example 6-14 shows how the alternative **pidstat** command can be used to list the top 10 CPU consuming processes. Similar to the **top** command, **pidstat** does not show the numeric logical database partition number next to the description of **db2sysc**.

Example 6-14 Determine top 10 CPU consuming processes pidstat

```
ISAS56R1D2:~ # pidstat | head -3; pidstat 2 1 | egrep -v -i 'average|Linux' | sort -nr -k 5 | head
Linux 2.6.16.60-0.21-smp (ISAS56R1D2) 10/10/10

06:15:43      PID  %user %system  %CPU  CPU  Command
06:15:45    30720 1254.23   3.48 1257.71  14  db2sysc
06:15:45    30687  330.35   3.98  334.33  15  db2sysc
06:15:45    30697   2.49   0.50   2.99  10  db2sysc
06:15:45    30650   2.49   0.50   2.99  10  db2sysc
06:15:45     7673   1.00   1.00   1.99  12  syslog-ng
06:15:45     8054   1.49   0.00   1.49   7  java
06:15:45     2438   0.00   1.49   1.49   0  mpp_dcr
06:15:45     7677   1.00   0.00   1.00  12  klogd
06:15:45     395   0.00   1.00   1.00   2  pidstat
06:15:43      PID  %user %system  %CPU  CPU  Command
ISAS56R1D2:~ #
```

In certain troubleshooting cases, it might be of interest to see which processes have been using up a lot of CPU resources over their “lifetime”. This information can tell you which processes are often consumers of a lot of CPU resources not just at this time but historically. The following commands show how to identify the top 10 cumulative CPU consumers:

► **Linux:**

```
ps aux | head -1; ps aux | sort -nr -k10 | head -10
```

► **AIX:**

```
ps aux | head -1; ps aux | sort -nr -k11 | head -10
```

Example 6-15 shows an output of the top 10 cumulative CPU consumers on our Linux system. Note that the **db2sysc 2** (process ID 28068) appears in the “historical” top 10 CPU consumer list, but it is not yet close to the top of the historical list. This might mean that it is a recently started process, and that it only made the historical top 10 list because its high current CPU usage. It was higher in CPU use compared to most other processes that have run on the system for a much longer time. The **START** column confirms that this process was started just today, at 04:34 AM. All the other processes in this top 10 historical list date back to August 17th.

Example 6-15 Linux top 10 cumulative CPU consumers using 'ps'

```
ISAS56R1D2:~ # ps aux | head -1;ps aux | sort -nr -k10 | head -10
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      8524  2.0  0.7 503004 466864 ?        Ssl  Aug17 1526:15 /opt/ibm/director/agent/...
root      2438  0.8  0.0      0     0 ?        S<   Aug17 631:59 [mpp_dcr]
root      7849  0.5  0.0      0     0 ?        SN  Aug17 441:24 [kipmi0]
root      8054  0.2  0.0 224260 20392 ?        S1  Aug17 221:12 ./jre/bin/java -Djava.compiler=NONE
-cp /usr/RaidMan/RaidMsgExt.jar:/usr/RaidMan/RaidMan.jar com.ibm.sysmgmt.raidmgr.agent.ManagementAgent
root       27  0.0  0.0      0     0 ?        SN  Aug17 54:54 [ksoftirqd/12]
root       13  0.0  0.0      0     0 ?        SN  Aug17 52:28 [ksoftirqd/5]
bcu1inux 28068 809 6.2 13936552 4109412 ? S1 04:34 49:46 db2sysc 2
root       11  0.0  0.0      0     0 ?        SN  Aug17 43:37 [ksoftirqd/4]
root       29  0.0  0.0      0     0 ?        SN  Aug17 41:24 [ksoftirqd/13]
root       17  0.0  0.0      0     0 ?        SN  Aug17 31:04 [ksoftirqd/7]
```

Checking the process level

You can further identify which thread is consuming the most CPU on the specific node by using the **ps** command. The following commands are used to list the top 10 CPU consuming threads of a given UNIX process ID:

► **Linux:**

```
$ ps -Lm -F | head -1;ps -Lm -F -p <PID> | sort -nr -k 5 | head -10
```

► **AIX:**

```
$ ps -mo THREAD | head -1; ps -mo THREAD -p <PID> | sort -rn -k 6 | head -10
```

THREAD is presented as a light weight process (LWP) on Linux and thread identification (TID) on AIX.

In the last section, we identified the DB2 **db2sysc 2** process (ID 28068) for logical database partition 2 has high CPU usage. Example 6-16 lists the top 10 CPU consuming threads of this process. The output shows that threads 29887, 29800, and 29746 in LWP column use 23% CPU each (equivalent of 0.23 of one CPU, out of 16 available CPUs on the system), with many others very close to the same CPU consumption. Because these threads are children of a DB2 process, you can then cross-reference them by thread ID in DB2 to determine what SQL or DB2 utility they are actually performing.

Example 6-16 List current top 10 CPU consuming threads using ps

```
ISAS56R1D2:~ # ps -Lm -F | head -1;ps -Lm -F -p 28068 | sort -nr -k 5 | head -10
UID      PID PPID  LWP  C NLWP   SZ  RSS PSR STIME TTY      TIME CMD
bcu1inux 28068 28065    - 99 138 3482806 3889452 - 04:34 ?      00:24:42 db2sysc 2
bcu1inux - - 29887 23 - - - 15 04:35 - 00:00:41 -
bcu1inux - - 29800 23 - - - 14 04:35 - 00:00:42 -
bcu1inux - - 29746 23 - - - 14 04:35 - 00:00:41 -
bcu1inux - - 29861 22 - - - 6 04:35 - 00:00:39 -
bcu1inux - - 29838 22 - - - 10 04:35 - 00:00:40 -
bcu1inux - - 29766 22 - - - 14 04:35 - 00:00:39 -
```

```
bcu1inux - - 29912 21 - - - 0 04:35 - 00:00:37 -
bcu1inux - - 29884 21 - - - 0 04:35 - 00:00:37 -
bcu1inux - - 29744 20 - - - 0 04:35 - 00:00:36 -
```

On Linux, an alternative method to determine the top 10 CPU consuming threads of a specific process is as follows:

```
pidstat -t | head -3;pidstat -p 10302 -t 2 1 | egrep -v -i
'average|Linux' | sort -nr -k 6 | head -10
```

Example 6-17 on page 142 shows this alternative method to list the top 10 threads of a specific process, PID 10302. Later we can track these threads (TID) in the DB2 tools to determine what they are actually doing.

Example 6-17 Using pidstat to show thread CPU usage

```
ISAS56R1D2:~ # pidstat -t | head -3;pidstat -p 10302 -t 2 1 | egrep -v -i 'average|Linux' | sort -nr
-k 6 | head -10
Linux 2.6.16.60-0.21-smp (ISAS56R1D2) 10/10/10
```

05:32:32	PID	TID	%user	%system	%CPU	CPU	Command
05:32:34	10302	-	1173.63	2.49	1176.12	10	db2sysc
05:32:34	-	12283	76.12	0.00	76.12	4	db2sysc
05:32:34	-	12183	66.17	0.00	66.17	7	db2sysc
05:32:34	-	12151	60.70	0.00	60.70	5	db2sysc
05:32:34	-	12282	57.71	0.00	57.71	2	db2sysc
05:32:34	-	12020	53.73	0.00	53.73	3	db2sysc
05:32:34	-	12106	51.24	0.00	51.24	11	db2sysc
05:32:34	-	12098	48.76	0.00	48.76	1	db2sysc
05:32:34	-	12185	45.77	0.00	45.77	10	db2sysc
05:32:34	-	12117	45.27	0.00	45.27	9	db2sysc

6.2.2 Disk I/O and block queue

In this section, we discuss how to troubleshoot the performance of disk I/O.

Identifying the most I/O consuming nodes

Start off by checking the I/O-related resources (I/O wait percentage, the block queue, and the percentage of the rolled-up bandwidth utilization of devices) across the entire system for all physical nodes.

Here we use the custom Perl script sa_io_mon.pl to check the I/O resource consumption on a global system level:

```
$ ./sa_io_mon.pl
```

Figure 6-1 shows a sample output of sa_io_mon.

```

ISAS56MGMT: # ./sa_io_mon.pl
sa_io_mon
-----CPU-----
Block  Tot  Tot  Tot  Tot
Queue usr  sys  idle  wio
-----
System Avg: 4.2  5.54  0.62  82.82  11.02
-----
ISAS56R1D1: 0.0  0.01  0.12  99.71  0.15
ISAS56R1D2: 0.0  0.14  0.05  99.16  0.65
ISAS56R1D3: 21.0  27.56  2.86  15.30  54.28
ISAS56R1D4: 0.0  0.00  0.01  99.99  0.00
ISAS56R1D5: 0.0  0.00  0.04  99.94  0.02
-----
Tot  Avg/dev  #Active  -- Nbr  IO Device Usage
#devices %util  devices  -- Nbr  devices in %util range --
0-30%  30-60%  60-90%  90-100%  tps  readKB  writeKB
-----
System Avg: 13  6.73  3.6  2.8  0.0  0.4  0.4  1318.0  815760.0  379.0
-----
ISAS56R1D1: 25  0.15  4.0  4.0  0.0  0.0  0.0  12.8  0.0  488.0
ISAS56R1D2: 10  1.05  6.0  6.0  0.0  0.0  0.0  13.2  0.0  582.9
ISAS56R1D3: 10  32.34  4.0  0.0  0.0  2.0  2.0  6560.0  4078800.0  320.0
ISAS56R1D4: 10  0.02  2.0  2.0  0.0  0.0  0.0  2.0  0.0  260.0
ISAS56R1D5: 10  0.07  2.0  2.0  0.0  0.0  0.0  2.2  0.0  244.0

```

Figure 6-1 sa_io_mon output

Example 6-18 is an excerpt of Figure 6-1 showing the columns of interest. The ISAS56R1D3 node appears to be working the hardest with respect to I/O. In the Block Queue column, there are 21 processes showing blocked waiting on I/O. The wio (Waiting on I/O) CPU column is at 54.28%, and there are four disk devices running at a high percentage of utilization (two between 60-90% and two near saturation between 90-100%). All these figures are noticeably higher on node ISAS56R1D3 compared with the rest of the nodes on the system. Hence we have to drill down at the node level on ISAS56R1D3 to figure out why this situation is occurring.

Example 6-18 Using sa_io_mon to check I/O consumption

```

ISAS56MGMT: # ./sa_io_mon.pl
sa_io_mon
-----CPU-----
Block  Tot  Tot  Avg/dev  #Active  -- Nbr  IO Device Usage
Queue  wio  #devices %util  devices  -- Nbr  devices in %util range --
0-30%  30-60%  60-90%  90-100%
-----
System Avg: 4.2  11.02  13  6.73  3.6  2.8  0.0  0.4  0.4
-----
ISAS56R1D1: 0.0  0.15  25  0.15  4.0  4.0  0.0  0.0  0.0
ISAS56R1D2: 0.0  0.65  10  1.05  6.0  6.0  0.0  0.0  0.0
ISAS56R1D3: 21.0  54.28  10  32.34  4.0  0.0  0.0  2.0  2.0
ISAS56R1D4: 0.0  0.00  10  0.02  2.0  2.0  0.0  0.0  0.0
ISAS56R1D5: 0.0  0.02  10  0.07  2.0  2.0  0.0  0.0  0.0

```

Checking I/O consumption at node level

On the node level, check the I/O consumption both process and device:

- ▶ Process or thread view:

Check which processes and threads are consuming the most I/O, and which devices they are accessing:

- On Linux, use `dmesg -c`
- On AIX, use `ps`

- ▶ File or device view:

Check which devices are experiencing the heaviest I/O load with `iostat 1 5`. After identifying which device is getting the heaviest I/O hits by specific or all processes, try to get more specific information as to which logical volume and which file system is involved. We provide scripts, `disk2fs.ksh` and `fs2disk.ksh` to aid in this device-to-file system mapping. See Appendix A, “Smart Analytics global performance monitoring scripts” on page 281 for the source code.

Example 6-19 shows the result of using `iostat -k -x 5` on node ISAS56D1R3 to identify the disk devices that are showing the high I/O activity. The drives with the most I/O activity are `sdc`, `sde`, `dm-0`, and `dm-2` at 100% I/O utilization. That is actually four devices saturated at 100%, not two as shown in Example 6-18.

Example 6-19 Identify the disk devices with high I/O activity

```
ISAS56R1D3: # iostat -k -x 5
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           26.74    0.00    2.50   57.71    0.00   13.05
```

Device:	rrqm/s	wrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	0.00	0.60	0.00	0.60	0.00	5.60	18.67	0.00	0.00	0.00	0.00
sdb	32.00	0.60	297.40	0.40	132972.80	4.80	893.07	2.24	7.53	2.21	65.92
sdc	61.80	1.00	3704.40	0.40	263929.60	6.40	142.48	11.23	3.03	0.27	100.00
sdd	32.20	0.60	297.00	0.40	133577.60	4.80	898.33	2.32	7.78	2.21	65.60
sde	76.40	1.60	837.60	0.40	313238.40	8.80	747.61	5.42	6.47	1.19	100.00
dm-0	0.00	0.00	914.00	2.00	313238.40	8.00	683.94	5.80	6.34	1.09	100.00
dm-1	0.00	0.00	330.40	1.00	134089.60	4.00	809.26	2.47	7.43	1.98	65.60
dm-2	0.00	0.00	3766.60	1.40	263993.60	5.60	140.13	11.56	3.07	0.27	100.00
dm-3	0.00	0.00	329.40	1.00	132972.80	4.00	804.94	2.37	7.19	2.00	65.92
dm-4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-6	0.00	0.00	0.00	1.20	0.00	4.80	8.00	0.00	0.00	0.00	0.00
dm-7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sdf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Using the lookup script `disk2fs.ksh`, Example 6-20 shows which file systems map to a specific disk device. We see that many of the devices are actually the synonyms for the same disk storage. For example, `sdc` and `dm-2` are really the same device (both mapped to file system `/db2fs/bcullinux/NODE0006`), and so are `sde` and `dm-0` (both mapped to file system `/db2fs/bcullinux/NODE0008`). The `sa_io_mon.pl` has filtered out the redundant statistics to avoid double counting.

Example 6-20 disk2fs.ksh shows which file system maps to a given disk

```
ISAS56R1D3: I/O device sdb --> filesystem mountdir: /db2fs/bcullinux/NODE0005 (LV:
/dev/vgdb2NODE0005/lvdb2NODE0005)
ISAS56R1D3:/home/pthoreso # ./disk2fs.ksh dm-3
ISAS56R1D3: I/O device dm-3 --> filesystem mountdir: /db2fs/bcullinux/NODE0005 (LV:
/dev/vgdb2NODE0005/lvdb2NODE0005)

ISAS56R1D3: # ./disk2fs.ksh sdc
ISAS56R1D3: I/O device sdc --> filesystem mountdir: /db2fs/bcullinux/NODE0006 (LV:
/dev/vgdb2NODE0006/lvdb2NODE0006)
ISAS56R1D3:# ./disk2fs.ksh dm-2
ISAS56R1D3: I/O device dm-2 --> filesystem mountdir: /db2fs/bcullinux/NODE0006 (LV:
/dev/vgdb2NODE0006/lvdb2NODE0006)

ISAS56R1D3:/home/pthoreso # ./disk2fs.ksh sdd
ISAS56R1D3: I/O device sdd --> filesystem mountdir: /db2fs/bcullinux/NODE0007 (LV:
/dev/vgdb2NODE0007/lvdb2NODE0007)
ISAS56R1D3:/home/pthoreso # ./disk2fs.ksh dm-1
ISAS56R1D3: I/O device dm-1 --> filesystem mountdir: /db2fs/bcullinux/NODE0007 (LV:
/dev/vgdb2NODE0007/lvdb2NODE0007)

ISAS56R1D3:/home/pthoreso # ./disk2fs.ksh sde
ISAS56R1D3: I/O device sde --> filesystem mountdir: /db2fs/bcullinux/NODE0008 (LV:
/dev/vgdb2NODE0008/lvdb2NODE0008)
```

```
ISAS56R1D3:/home/pthoreso # ./disk2fs.ksh dm-0
ISAS56R1D3: I/O device dm-0 --> filesystem mountdir: /db2fs/bculinux/NODE0008 (LV:
/dev/vgdb2NODE0008/1vdb2NODE0008)
```

Now we know which of the disks are running the “hottest” at 100% utilization and what file system they correspond to: `sdcdm-2 = /db2fs/bculinux/NODE0006` (for logical database partition 6) and `sdedm-0=/db2fs/bculinux/NODE0008` (logical database partition 8).

The other busy devices are `sdbdm-3=/db2fs/bculinux/NODE0005` file system for logical database partition 5, and `sdddm-1=/db2fs/bculinux/NODE0007` file system for logical database partition 7, both at 65%.

Because the I/O is only in the 0-30% range for the other physical nodes, this shows that the I/O activity is not evenly spread: database partitions 6 and 8 at 100% utilization, logical database partitions 5 and 7 at 65%, and all the other nodes at less than 30%. This situation might indicate a case of database data skew across the logical database partitions with logical database partitions 6 and 8 having the most data for a given table, logical database partitions 5 and 7 the next most, and much less in all the other database partitions. This information can be confirmed at the DB2 level.

Checking I/O consumption at the hardware layer

Hardware problems also can cause performance issues. In this section we show how to map a disk device, LUN, and controller.

To check which corresponding hardware LUNs are involved, use the following command to see the disk device to LUN mapping:

- ▶ Linux: `/opt/mpp/lsvdev`
- ▶ AIX: `mpio_get_config -Av`

Example 6-21 shows the LUN to disk device mapping of our “hot I/O” node ISAS56R1D3. LUN1 for storage array Storage03 maps to disk device `sdcd`, which corresponds to file system `/db2fs/bculinux/NODE0006` for logical database partition 6, and LUN1 for array Storage04 maps to disk device `sded` corresponding to file system `/db2fs/bculinux/NODE0008`.

Example 6-21 lsvdev shows the hardware LUN to disk device mapping (Linux only)

```
ISAS56R1D3:/home/pthoreso # /opt/mpp/lsvdev
  Array Name      Lun    sd device
-----
Storage03        0     -> /dev/sdb
Storage03       1     -> /dev/sdc
Storage04        0     -> /dev/sdd
Storage04       1     -> /dev/sde
```

To find the mapping between the hardware LUN array and the storage controller and path, use the `mppUtil -S` command.

Example 6-22 shows the controller to LUN array mapping for the array storage Storage03 and Storage04 identified in Example 6-21.

Example 6-22 mppUtil maps the controller to LUN array mappings

```
ISAS56R1D3: # mppUtil -S
H9C0T2      Active           Active           Storage03
            H5C0T2L000 Up       H6C0T2L000 Up
            H7C0T2L000 Up       H8C0T2L000 Up
            H5C0T2L001 Up       H6C0T2L001 Up
            H7C0T2L001 Up       H8C0T2L001 Up
H9C0T3      Active           Active           Storage04
            H5C0T3L000 Up       H6C0T3L000 Up
            H7C0T3L000 Up       H8C0T3L000 Up
            H5C0T3L001 Up       H6C0T3L001 Up
            H7C0T3L001 Up       H8C0T3L001 Up
```

Example 6-23 shows how to interpret the `mppUtil -s` command output. The output shows the following conditions:

- ▶ The path for the two controllers (A & B controllers)
- ▶ Which host, channel, and target make up the path to the hardware LUN arrays
- ▶ Which LUN arrays are up or down

Example 6-23 mppUtil man page

```
ISAS56R1D3:/home/pthoreso # man mppUtil
. . .
Decoding the output of mppUtil -S
Virtual ( Host, Channel, Target )      Controller State      Array Name
  | | |                                | | |                  | | |
  V V V                                V V V                  V V V

H6C0T1      Offline                    Active                 ausct1r_34
            H2C0T4L000 Up
            H2C0T4L001 Up
            H2C0T4L003 Up
            H2C0T4L004 Up
            H2C0T4L004 Up

H6C0T0      Active                      Active                 MPP_Yuma1
            H2C0T2L000 Up
            H2C0T3L000 Up
            H2C0T2L001 Up
            H2C0T3L001 Up
            H2C0T2L088 Up
            H2C0T3L088 Up

            ^ ^ ^                      ^ ^ ^
            | | |                      | | |
            I-T-Nexus                  CTLR A Lun(s)
            Path State                  Path State

            ^ ^ ^                      ^ ^ ^
            | | |                      | | |
            CTLR B Lun(s)
            Path State
```

To drill down further for the LUN, controller and pathway information, use the `mppUtil -a <array name>` command.

Example 6-24 shows the details of array Storage03. Note that there are multiple redundant pathways to the LUN array: through either of the two controllers (A and B) and two paths for each controller. This output shows that LUN1 corresponding to disk sdc has a LUN identifier (WWN) 600a0b80006771ae0000082e4c3f8580. The controllers A and B are with the following pathways:

- ▶ Controller A: Path#1: hostId: 5, channelId: 0, targetId: 2, and Path#2: hostId: 7, channelId: 0, targetId: 2.
- ▶ Controller B: Path#1: hostId: 6, channelId: 0, targetId: 2, and Path#2: hostId: 8, channelId: 0, targetId: 2.

Example 6-24 Show LUN, controller and pathway information

```
ISAS56R1D3:/home/pthoreso # mppUtil -a Storage03 | more
Hostname = ISAS56R1D3
Domainname = N/A
Time = GMT 10/11/2010 10:12:46

MPP Information:
-----
      ModuleName: Storage03                      SingleController: N
VirtualTargetID: 0x002                          ScanTriggered: N
      ObjectCount: 0x000                          AVTEnabled: N
      WWN: 600a0b80006773cb000000004bf40c6c      RestoreCfg: N
      ModuleHandle: none                          Page2CSubPage: Y
FirmwareVersion: 7.35.41.xx
      ScanTaskState: 0x00000000
      LBPolicy: LeastQueueDepth

Controller 'A' Status:
-----
ControllerHandle: none                          ControllerPresent: Y
      UTMlunExists: Y (031)                       Failed: N
      NumberOfPaths: 2                            FailoverInProg: N
                                              ServiceMode: N

      Path #1
      -----
      DirectoryVertex: present                    Present: Y
      PathState: OPTIMAL
      PathId: 77050002 (hostId: 5, channelId: 0, targetId: 2)

      Path #2
      -----
      DirectoryVertex: present                    Present: Y
      PathState: OPTIMAL
      PathId: 77070002 (hostId: 7, channelId: 0, targetId: 2)

Controller 'B' Status:
-----
ControllerHandle: none                          ControllerPresent: Y
      UTMlunExists: Y (031)                       Failed: N
      NumberOfPaths: 2                            FailoverInProg: N
                                              ServiceMode: N

      Path #1
```

```

-----
DirectoryVertex: present                               Present: Y
PathState: OPTIMAL
PathId: 77060002 (hostId: 6, channelId: 0, targetId: 2)

Path #2
-----
DirectoryVertex: present                               Present: Y
PathState: OPTIMAL
PathId: 77080002 (hostId: 8, channelId: 0, targetId: 2)

Lun Information
-----
. . .
Lun #1 - WWN: 600a0b80006771ae0000082e4c3f8580
-----
LunObject: present                                   CurrentOwningPath: B
RemoveEligible: N                                   BootOwningPath: B
NotConfigured: N                                   PreferredPath: B
DevState: OPTIMAL                                   ReportedPresent: Y
                                                    ReportedMissing: N
                                                    NeedsReservationCheck: N
                                                    TASBitSet: Y
                                                    NotReady: N
                                                    Busy: N
                                                    Quiescent: N

Controller 'A' Path
-----
NumLunObjects: 2                                     RoundRobinIndex: 0
Path #1: LunPathDevice: present
          DevState: OPTIMAL
          RemoveState: 0x0 StartState: 0x1 PowerState: 0x0
Path #2: LunPathDevice: present
          DevState: OPTIMAL
          RemoveState: 0x0 StartState: 0x1 PowerState: 0x0

Controller 'B' Path
-----
NumLunObjects: 2                                     RoundRobinIndex: 0
Path #1: LunPathDevice: present
          DevState: OPTIMAL
          RemoveState: 0x0 StartState: 0x1 PowerState: 0x0
Path #2: LunPathDevice: present
          DevState: OPTIMAL
          RemoveState: 0x0 StartState: 0x1 PowerState: 0x0

. . .
ISAS56R1D3:#

```

6.2.3 Memory usage

Memory over-allocation that results in paging is a commonly seen performance degradation indicator. In this section, we discuss how to check for nodes consuming the most memory resources for the IBM Smart Analytics System.

Start your monitoring from the global system level. We use the custom script `sa_paging_mon.pl` to check the entire system. Figure 6-2 shows an output of custom script `sa_paging_mon.pl` with page swapping, real memory usage, and swap space usage information from an IBM Smart Analytics System 5600.

sa_paging_mon	Run Queue	Block Queue	----- CPU -----				-- Page Swapping --		----- Real Memory -----			----- Swap Space -----		
			usr	sys	idle	wio	in	out	Total	Used	Free	Total	Used	Free
System Avg:	0.0	1.6	0.2	0.8	89.8	9.2	0	0	65981668	23395962	42585705	33559744	29	33559714
ISAS56R1D1:	0.0	0.0	0.0	0.0	100.0	0.0	0	0	65981668	17371268	48610400	33559744	0	33559744
ISAS56R1D2:	0.0	4.0	0.0	2.0	75.0	23.0	0	0	65981668	45034408	20947260	33559744	0	33559744
ISAS56R1D3:	0.0	4.0	1.0	2.0	74.0	23.0	0	0	65981668	45132744	20848924	33559744	0	33559744
ISAS56R1D4:	0.0	0.0	0.0	0.0	100.0	0.0	0	0	65981668	4479436	61502232	33559744	0	33559744
ISAS56R1D5:	0.0	0.0	0.0	0.0	100.0	0.0	0	0	65981668	4961956	61019712	33559744	148	33559596
ISAS56MGMT: #														

Figure 6-2 Paging monitoring

For readability, we split the display in half as shown in Example 6-25. There is no swapping currently occurring on this system. However, if there was, we can notice it in the Page Swapping columns for pages swapped in and pages swapped out. Simply monitor for any excessive activity in this area, the free memory decreasing excessively, and the swap space used actually increasing from the usual *zero* normally seen on an IBM Smart Analytics System.

Example 6-25 Formatted sa_paging_mon.pl output

sa_paging_mon	Run Queue	Block Queue	----- CPU -----				-- Page Swapping --	
			usr	sys	idle	wio	in	out
System Avg:	0.0	1.6	0.2	0.8	89.8	9.2	0	0
ISAS56R1D1:	0.0	0.0	0.0	0.0	100.0	0.0	0	0
ISAS56R1D2:	0.0	4.0	0.0	2.0	75.0	23.0	0	0
ISAS56R1D3:	0.0	4.0	1.0	2.0	74.0	23.0	0	0
ISAS56R1D4:	0.0	0.0	0.0	0.0	100.0	0.0	0	0
ISAS56R1D5:	0.0	0.0	0.0	0.0	100.0	0.0	0	0

----- Real Memory -----			----- Swap Space -----		
Total	Used	Free	Total	Used	Free
65981668	23395962	42585705	33559744	29	33559714
65981668	17371268	48610400	33559744	0	33559744
65981668	45034408	20947260	33559744	0	33559744
65981668	45132744	20848924	33559744	0	33559744
65981668	4479436	61502232	33559744	0	33559744
65981668	4961956	61019712	33559744	148	33559596

If you see any abnormal paging activity on a particular node, check the process on the node that consumes the most real memory (RSS) using the **ps aux** command. Following are commands to show the top 10 real memory consuming processes:

- ▶ Linux: **ps aux | head -1; ps aux | sort -nr -k 6 | head**
- ▶ AIX: **ps aux | head -1; ps aux | sort -nr -k 5 | head**

Example 6-26 shows an output of the **ps aux** command on our IBM Smart Analytics System 5600 V1. Here the main DB2 engine processes **db2sysc** (one per logical database partition) are the processes using the most real memory on the system (a little over 3.5 GB of memory for each of the four processes). All other processes are consuming at least an order of magnitude less than them.

If paging was occurring, we want to look at these four DB2 processes to see why they are using so much memory and if something can be done to use less memory until the paging stops.

Example 6-26 Using ps aux to determine top 10 real memory consuming processes

```
ISAS56R1D3: # ps aux | head -1; ps aux | sort -nr -k 6 | head
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
bculinx  30214  640  5.8 13931356 3845884 ?    S1   06:10  14:05 db2sysc 8
bculinx  30181  510  5.7 13930272 3795332 ?    S1   06:10  11:19 db2sysc 6
bculinx  30168  4.1  5.6 13658836 3744564 ?    S1   06:10   0:05 db2sysc 5
bculinx  30204  4.3  5.5 13651612 3685304 ?    S1   06:10   0:05 db2sysc 7
root      8524   1.9  0.7  503004 466216 ?    Ss1  Aug17 1553:40
/opt/ibm/director/agent/_jvm/jre/bin/java -Xmx384m -Xminf0.01 -Xmaxf0.4
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000
-Xbootclasspath/a:/opt/ibm/director/agent/runtime/core/rcp/eclipse/plugins/com.ibm.rcp.base_6.1.2.200
801281200/rcpbootcp.jar:/opt/ibm/director/agent/lib/icl.jar:/opt/ibm/director/agent/lib/jaas2zos.jar:
/opt/ibm/director/agent/lib/jaasmodule.jar:/opt/ibm/director/agent/lib/lwinative.jar:/opt/ibm/directo
r/agent/lib/lwirolemap.jar:/opt/ibm/director/agent/lib/passutils.jar:../runtime/agent/lib/cas-boot
cp.jar -Xverify:none -cp
eclipse/launch.jar:eclipse/startup.jar:/opt/ibm/director/agent/runtime/core/rcp/eclipse/plugins/com.i
bm.rcp.base_6.1.2.200801281200/launcher.jar com.ibm.lwi.LaunchLWI
root      8076   0.0  0.1 100156 92688 ?    Ss   Sep29   0:00 bash -c echo `hostname`:`vmstat 1
2>&1; echo D3 CCC `
root      30166  0.0  0.0 9186600 34080 ?    S1   06:10   0:00 db2wdog 5
root      30189  0.0  0.0 9186600 34064 ?    S1   06:10   0:00 db2wdog 7
root      30212  0.0  0.0 9186596 34060 ?    S1   06:10   0:00 db2wdog 8
root      30178  0.0  0.0 9186600 34060 ?    S1   06:10   0:00 db2wdog 6
ISAS56R1D3: #
```

Drilling down to the thread ID level is not relevant for this type of resource because all children threads show the same memory usage as their parent PID process.

6.2.4 Network

To check status, the number of packets RX/TX, and the number of bytes RX/TX for networks, use **netstat** and **ifconfig**.

Example 6-27 shows an output of the **ifconfig** command.

Example 6-27 ifconfig

```
$ ifconfig bond0 | egrep 'RX|TX';sleep 2;ifconfig bond0 | egrep 'RX|TX'
RX packets:712683689 errors:0 dropped:0 overruns:0 frame:0
TX packets:972731266 errors:0 dropped:0 overruns:0 carrier:0
RX bytes:584318299554 (557249.3 Mb) TX bytes:946042644765 (902216.5 Mb)
RX packets:712683755 errors:0 dropped:0 overruns:0 frame:0
TX packets:972731336 errors:0 dropped:0 overruns:0 carrier:0
RX bytes:584318308198 (557249.3 Mb) TX bytes:946042691841 (902216.6 Mb)
```

You can use the **netstat** command to check the network configuration and activity. Example 6-28 shows an output of **netstat -i**.

Example 6-28 netstat -i

```
$ netstat -i
Name Mtu Network Address ZoneID Ipkts Ierrs Opkts Oerrs Coll
en0 1500 link#2 0.21.5e.79.5d.60 - 15728584 0 12039746 0 0
en0 1500 10.199.67 ISAS56R1ADMmgt - 15728584 0 12039746 0 0
en11 9000 link#3 0.21.5e.89.23.7f - 2509667120 0 2500771760 3 0
en11 9000 10.199.66 ISAS56R1ADMcorp - 2509667120 0 2500771760 3 0
en12 9000 link#4 0.21.5e.89.23.7e - 277706206 0 153448153 3 0
en12 9000 10.199.64 ISAS56R1ADMapp - 277706206 0 153448153 3 0
en12 9000 10.199.64 VISAS56R1ADMapp - 277706206 0 153448153 3 0
lo0 16896 link#1 - 451974151 0 448999647 0 0
lo0 16896 127 loopback - 451974151 0 448999647 0 0
lo0 16896 ::1 0 451974151 0 448999647 0 0
```

The **netstat -s** command shows the statistics for each protocol. Example 6-29 shows an excerpt the **netstat -s** command output.

Example 6-29 netstat -s

```
$netstat -s
icmp:
  3154479 calls to icmp_error
  0 errors not generated because old message was icmp
  Output histogram:
    echo reply: 3042
    destination unreachable: 3154479
    echo: 67
    information request reply: 1
  0 messages with bad code fields
  0 messages < minimum length
  0 bad checksums
  0 messages with bad length
  Input histogram:
    echo reply: 76
    destination unreachable: 3153787
    echo: 3039
    address mask request: 153
  3039 message responses generated
igmp:
  0 messages received
  0 messages received with too few bytes
  0 messages received with bad checksum
  0 membership queries received
  0 membership queries received with invalid field(s)
  0 membership reports received
  0 membership reports received with invalid field(s)
  0 membership reports received for groups to which we belong
  10 membership reports sent
tcp:
  3089466626 packets sent
    488366659 data packets (3898579545 bytes)
    217 data packets (2566776 bytes) retransmitted
    1082199945 ack-only packets (8357336 delayed)
    2 URG only packets
...

```

6.3 DB2 Performance troubleshooting

In the previous section, we reviewed operating system commands and methods to check for CPU, memory, I/O, and network bottlenecks on the system. In this section, we discuss how to check the usage of these resources from a DB2 perspective.

We discuss the most common scenarios that consist of identifying the entire workload of the SQL statement or utility consuming resources such as CPU, memory, I/O, and network. For the memory resources, we examine how to review the overall DB2 memory usage using DB2 commands.

In this section, we use the **db2top** utility in the performance problem determination examples. However, there are other options available using either native DB2 snapshots, the **db2pd** utility, or DB2 9.7 new relational monitoring functions.

For high CPU usage situations, one good approach is to isolate the thread consuming the most CPU at the operating system level as discussed in the previous section. In this section, we discuss how to further identify detail activity of the DB2 thread.

The **db2pd -edus** command is useful for this purpose. Example 6-30 shows an example of **db2pd -edus** output from the first data node for logical database partition one.

Example 6-30 db2pd -edus output

```
db2pd -edus -dbp 1
Database Partition 1 -- Active -- Up 3 days 15:44:58 -- Date 10/04/2010 14:30:37

List of all EDUs for database partition 1

db2sysc PID: 623
db2wdog PID: 618
db2acd  PID: 721
```

EDU ID	TID	Kernel TID	EDU Name	USR (s)	SYS(s)
1318	47785151818048	29586	db2agnta (BCUKIT) 1	82.300000	9.090000
1315	47790205954368	29579	db2agntp (BCUKIT) 1	0.190000	0.080000
1314	47790176594240	29578	db2agntp (BCUKIT) 1	44.920000	6.040000
1308	47790201760064	28469	db2agntdp (BCUKIT) 1	0.000000	0.000000
1245	47785143429440	27960	db2agnta (BCUKIT) 1	833.000000	48.130000
1244	47785164400960	27959	db2agntp (BCUKIT) 1	57.820000	4.200000
1243	47785130846528	27958	db2agntp (BCUKIT) 1	586.960000	33.080000
1239	47790424058176	27953	db2agntp (BCUKIT) 1	6.570000	0.440000
1230	47785218926912	27944	db2agntp (BCUKIT) 1	208.780000	33.750000
1228	47785185372480	27943	db2agntp (BCUKIT) 1	267.160000	36.630000
1220	47785160206656	27935	db2agntp (BCUKIT) 1	11.920000	0.740000
1214	47790264674624	27928	db2agnta (BCUKIT) 1	0.500000	0.210000
1213	47790382115136	27927	db2agntp (BCUKIT) 1	47.070000	2.940000
1203	47785319590208	27918	db2agntp (BCUKIT) 1	2.980000	1.930000
1202	47790298229056	27917	db2agnta (BCUKIT) 1	148.150000	13.540000

1201	47785206344000	27913	db2agnta (BCUKIT)	1	123.120000	17.920000
1198	47790352755008	27912	db2agntp (BCUKIT)	1	6.000000	0.250000
1197	47790411475264	27910	db2agnta (BCUKIT)	1	46.260000	4.460000
1194	47790319200576	27903	db2agntp (BCUKIT)	1	173.240000	7.410000
1182	47790222731584	27896	db2agntp (BCUKIT)	1	167.260000	43.840000
1181	47790235314496	27895	db2agntp (BCUKIT)	1	301.890000	28.780000
1174	47790415669568	27889	db2agntp (BCUKIT)	1	19.950000	2.380000
1159	47785193761088	27874	db2agntp (BCUKIT)	1	62.360000	3.890000
1158	47790260480320	27873	db2agnta (BCUKIT)	1	78.100000	13.630000
1157	47790285646144	27872	db2agntp (BCUKIT)	1	347.670000	44.170000
1156	47790315006272	27870	db2agntp (BCUKIT)	1	18.590000	1.150000
1148	47785235704128	27863	db2agntp (BCUKIT)	1	100.900000	8.310000
1147	47785214732608	27862	db2agent (idle)	1	96.790000	5.150000
1124	47785277647168	26568	db2pfchr (BCUKIT)	1	9.930000	13.190000
1123	47785265064256	26567	db2pfchr (BCUKIT)	1	10.020000	13.210000

The header includes key information such as the time because the database partition has been activated and the **db2sysc** PID associated to the logical database partition.

The **db2pd -edus** command provides the following information:

- ▶ **EDU ID:** DB2 engine dispatchable unit identification (EDU ID) identifies the thread from the DB2 perspective, and is useful to match with DB2 outputs such as LIST APPLICATIONS, db2diag.log messages, and monitoring outputs, as well as running certain DB2 troubleshooting commands.
- ▶ **Kernel TID:** You can match the kernel TID obtained in the operating system level output with this command output to identify a particular DB2 EDU.
- ▶ **EDU name:** Identifies the thread name. This is useful to understand what the thread is used for. The DB2 process model document details the particular threads names and their function in the DB2 9.7 Information Center at the following link:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0008930.html>

- ▶ **USR(s) and SYS(s):** User and system CPU usage elapsed time in seconds. This information can be very useful. Note that the highest elapsed time might not necessarily be associated with the thread consuming CPU at the current time. For example, a DB2 agent used by an application might have completed an expensive SQL increasing its elapsed user CPU usage time, but is not consuming the highest CPU at this time. Other threads that might show a high elapsed CPU usage might be threads spawned at the instance or database activation. For example, **db2fcmr** and **db2fcms** threads might report a high CPU elapsed time, if the instance has been up for a long time.

In order to check for the current CPU consumption, the **db2pd -edus** with the suboption **interval** can be used. This suboption adds two additional columns showing the user and system CPU usage in the interval specified, and orders the output by decreasing CPU usage, according to the excerpt shown in Example 6-31.

Example 6-31 db2pd -edus -interval output

```
# db2pd -edus interval=5 -dbp 1

Database Partition 1 -- Active -- Up 0 days 00:14:28 -- Date 01/06/2011 13:48:58

List of all EDUs for database partition 1

db2sysc PID: 757798
db2wdog PID: 774332
db2acd PID: 737434

EDU ID      TID      Kernel TID    EDU Name                USR (s)    SYS (s)    ...
-----
3343      3343    1384669      db2loggr (BCUDB3) 1    0.818148   0.409035   ...
258       258     729173      db2sysc 1              1.715241   0.311393   ...
2829      2829    643151      db2stmm (BCUDB3) 1    0.067231   0.039958   ...
6427      6427    2089213     db2fw4 (BCUDB3) 1    0.026378   0.025864   ...
5399      5399    2072821     db2fw0 (BCUDB3) 1    0.028868   0.025466   ...
...

...  USR DELTA    SYS DELTA
...  -----
...  0.010282     0.005630
...  0.006918     0.002972
...  0.002075     0.000707
...  0.000228     0.000300
...  0.000229     0.000284
...
```

db2pd -edus interval=5 (for the past 5 seconds for example) can be used when beginning a high CPU usage investigation to check for the top CPU consuming DB2 threads. In certain cases, the CPU consumption will stand out.

Example 6-32 shows a real example of a DB2 page cleaner EDU ID 4040 that has a much higher CPU usage than the rest of the processes on an IBM Smart Analytics System 7600. Further verification at the operating system level confirms that this process is using 100% of the CPU. In this example, the high user CPU elapsed time is caused by the DB2 page cleaner looping.

Example 6-32 db2pd -edus excerpt

EDU ID	TID	Kernel TID	EDU Name	USR (s)	SYS (s)
8967	8967	1426153	db2agntdp (BCUKIT) 4	0.000228	0.000036
8710	8710	569875	db2agent (idle) 4	0.008675	0.010229
7941	7941	3064417	db2agntp (BCUKIT) 4	0.078646	0.038762
8686	8686	1528513	db2agntdp (BCUKIT) 4	0.006494	0.003923
8429	8429	754311	db2agent (idle) 4	0.029122	0.013138
4297	4297	1983293	db2pfchr (BCUKIT) 4	40.026308	22.125700
4040	4040	1766045	db2pclnr (BCUKIT) 4	11530.323230	43.076620
3783	3783	787381	db2dlock (BCUKIT) 4	0.054281	0.010329
3526	3526	1180441	db2lfr (BCUKIT) 4	0.000074	0.000009
3269	3269	1414115	db2loggw (BCUKIT) 4	1.896206	3.202154
3012	3012	1618723	db2loggr (BCUKIT) 4	1.908058	2.395301
2571	2571	508915	db2resync 4	0.000584	0.000768
2314	2314	1073789	db2ipccm 4	0.008313	0.005932
2057	2057	2024273	db2licc 4	0.000106	0.000396
1800	1800	1065723	db2pdbc 4	0.041439	0.019909
1543	1543	2753455	db2extev 4	0.004683	0.014039
1286	1286	1159695	db2fcmr 4	0.022657	0.013068
1029	1029	836607	db2extev 4	0.000105	0.000015
772	772	1143447	db2fcms 4	0.037230	0.026084
515	515	774701	db2thcln 4	0.005262	0.001580
22	2	25017	db2alarm 4	0.054986	0.054972
258	258	733847	db2sysc 4	0.728096	0.821109

6.3.1 CPU consumption

In the following sections we discuss CPU consumption by applications, utilities, and other activities.

Applications consuming CPU

In this section, as an example, we examine a way of identifying the application using the highest amount of CPU, and then use the **db2top** utility to obtain further details about the application using it. Other methods of narrowing down a top CPU consuming thread to an application, and the SQL being executed, are mentioned at the end of this section.

We assume a situation where the IBM Smart Analytics System shows a higher than usual CPU usage.

In 6.2.1, “CPU, run queue, and load average monitoring” on page 137 we have reviewed how to identify the process and threads having the highest CPU usage. Using the process, we first collect a **ps** output that shows the thread with the highest CPU usage on the first data node. Because all **db2sysc** processes appear to consume equally high CPU, we can just pick one **db2sysc** process for further review. In this example, we get the thread level **ps** output for **db2sysc** corresponding to database partition one, with PID 623.

Alternatively, you can also use **db2pd -edus interval=5** to identify the top CPU consuming thread.

The **ps** command output in Example 6-33 shows that thread ID 6644 has the highest CPU usage (based on the fifth column).

Example 6-33 Thread with highest CPU

```
ps -Lm -Fp 623 | sort -rn +4
bcu linux 623 618 - 68 115 3553910 6239920 - Sep30 ? 14:11:15 db2sysc 1
bcu linux - - 6644 10 - - - 12 12:58 - 00:38:18 -
bcu linux - - 6778 8 - - - 4 12:58 - 00:30:49 -
bcu linux - - 6701 7 - - - 9 12:58 - 00:26:49 -
bcu linux - - 6793 6 - - - 7 12:58 - 00:24:14 -
bcu linux - - 6617 6 - - - 7 12:58 - 00:24:38 -
bcu linux - - 6711 5 - - - 7 12:58 - 00:20:18 -
bcu linux - - 6677 5 - - - 7 12:58 - 00:19:55 -
bcu linux - - 6648 5 - - - 10 12:58 - 00:20:29 -
bcu linux - - 11346 5 - - - 8 14:30 - 00:16:05 -
bcu linux - - 11345 5 - - - 14 14:30 - 00:17:05 -
bcu linux - - 6815 4 - - - 7 12:58 - 00:17:20 -
bcu linux - - 6807 4 - - - 7 12:58 - 00:18:46 -
bcu linux - - 6739 4 - - - 2 12:58 - 00:15:37 -
[...]
```

db2pd -edus -dbp 1 is used on data node 1 to get the details about all the threads running on database partition 1. Example 6-34 shows an excerpt of **db2pd -edus -dbp 1**.

Example 6-34 db2pd -edus -dbp 1 output from the first data node

```
# db2pd -edus -dbp 1

Database Partition 1 -- Active -- Up 0 days 20:51:15 -- Date 10/01/2010 19:36:54

List of all EDUs for database partition 1

db2sysc PID: 623
db2wdog PID: 618
db2acd  PID: 721

EDU ID      TID          Kernel TID   EDU Name                                     USR (s)  SYS (s)
-----
908      47790222731584 11458        db2agntdp (BCUKIT ) 1                    0.160000    0.020000
907      47790226925888 11457        db2agntdp (BCUKIT ) 1                   235.070000   30.130000
903      47790231120192 11361        db2agntdp (BCUKIT ) 1                    47.450000    6.330000
899      47790235314496 11348        db2agntdp (BCUKIT ) 1                     0.010000    0.010000
898      47790239508800 11347        db2agntp (BCUKIT ) 1                   320.480000   17.560000
897      47790243703104 11346        db2agntdp (BCUKIT ) 1                    844.460000  120.990000
896      47790247897408 11345        db2agntdp (BCUKIT ) 1                    955.250000   70.430000
894      47790252091712 11265        db2agntdp (BCUKIT ) 1                     75.910000    9.880000
893      47790256286016 11263        db2agntdp (BCUKIT ) 1                     0.490000     0.170000
892      47790260480320 11262        db2agntdp (BCUKIT ) 1                     45.380000    4.990000
885      47790264674624 7124         db2agntdp (BCUKIT ) 1                    626.010000   77.660000
884      47790268868928 7123         db2agntdp (BCUKIT ) 1                     90.400000   11.510000
883      47790273063232 7122         db2agntdp (BCUKIT ) 1                    243.340000   14.670000
882      47790277257536 7121         db2agntdp (BCUKIT ) 1                    140.020000   17.340000
879      47790281451840 6825         db2agnta (BCUKIT ) 1                    465.080000   71.830000
869      47790285646144 6816         db2agntdp (BCUKIT ) 1                     31.360000    2.190000
868      47790289840448 6815         db2agntdp (BCUKIT ) 1                    959.880000   81.030000
867      47790294034752 6814         db2agntdp (BCUKIT ) 1                    550.330000   67.770000
861      47790298229056 6807         db2agntdp (BCUKIT ) 1                   1015.500000  110.870000
850      47790302423360 6796         db2agnta (BCUKIT ) 1                     848.240000   50.070000
[...]
```

The kernel TID returned in the `ps` command is in the third column in `db2pd -edus` output. We can use the `grep` command to filter out the output, as shown in Example 6-35. In this example, we are looking for TID 6644.

Example 6-35 Filtering the output

```
db2pd -edus | grep 6644
698      47785227315520 6644        db2agntp (BCUKIT ) 1                    2160.030000  160.520000
```

In this example, it turns out that the highest CPU consumer is a DB2 subagent with EDU ID 698. If a db2 agent or subagent consumes a high amount of user CPU, the CPU consumption is generally related to an expensive query.

The next step consists in narrowing down the SQL statement executed by that particular agent, for further investigation. We use `db2top` to check the application consuming the most CPU, and match it back to the thread seen at the DB2 level.

Figure 6-3 shows the welcome screen for **db2top** run from the administration node using the following command:

```
db2top -d bcukit
```

In this case, **bcukit** represents the database name.

```
[~] 14:09:13, refresh=2secs (0.007) Linux, part=[9/9], BCULINUX:BCUKIT
[d=Y, a=N, e=N, p=ALL] [qp=off]

#####          #####          #####          #####          #####          #####          For help type h or ...
# # # # # # # # # # # # # # # # # # # # # # # db2top -h: usage
# # # # # # # # # # # # # # # # # # # # # # #
# # #####          #####          # # # #####          Status: Active
# # # # # # # # # # # # # # # # # # # # # # # Uptime: 01h:11m
# # # # # # # # # # # # # # # # # # # # # # # Last backup
#####          #####          # # #####          # # # # # # # # # # # # # # # # # # # # # # # None

DB2 Interactive Snapshot Monitor V2.0
Use these keys to navigate:
d - Database          l - Sessions          a - Agent
t - Tablespaces      b - Bufferpools      T - Tables
D - Dynamic SQL      U - Locks           m - Memory
s - Statements       p - Partitions       u - Utilities
A - HADR             F - Federation      B - Bottlenecks
J - Skew monitor     q - Quit

Licensed Materials - Property of IBM
Copyright IBM Corp. 2005, 2006 All Rights Reserved. db2top 2.0
```

Figure 6-3 db2top welcome screen

Press **1** to go to the Sessions screen, which lists all the applications. Press **z** to sort the columns in ascending order. You are looking for the application consuming the most CPU, so sort per column Cpu% total, which is column one (column ordering starts at zero). Enter **1** when prompted for the “Column number for descending sort.”

For a screen that shows all columns ordered by percent total CPU consumption, see Figure 6-4. In our example, the application handle 456 consumes the most CPU.

```

[ \ ] 9:44:07,refresh=2secs(0.043) Sessions Linux,part=[9/9] BCUJINX:BCUKI
[d=v,a=N,e=N,p=ALL]
-----
Local sessions...: 15 x ActSessions x-----x Local in exec...: 14
Remote sessions...: 151 xActSessions x-----x Remote in exec...: 147
FCM buffers sent.: 927103747 xSys/User Cpu% xu x Active utilities...: 8
FCM buffers rec.: 695448756 xrt+w/sec% x-----x FCM buffers LWM...: 74,799
Piped sorts req.: 3,753 Agentstolen...: 2,650 Agent overflows...: 0 Piped sorts acc.: 3,753
Max Agents...: 0 Max Sort Heap...: 1.0G Sort heap alloc...: 2.5G Connections HWM...: 47
Sheaphres...: 20.5G Static SQL stmts.: 6 Post hash joins...: 0 Private memory...: 375.3M
Dynamic SQL stmts.: 1,004,062 Bufferpool reads.: 127,054 Bufferpool writes.: 1 Direct reads...: 0 Direct writes...: 0
-----
Application Handle(Stat) Cpu% Total IO% Total Mem% Total Application Status Application Name Delta RowsRead/s Delta RowsWritten/s Delta IOReads/s Delta IOWrites/s
-----
456(*) 45.99% 28.62% 0.13% UOW Executing db2hpp 8,939,701 0 36,364 0
414(*) 22.29% 9.38% 32.91% UOW Executing db2batch 1,330,901 4,014 49,444 0
418(*) 7.23% 9.38% 22.18% UOW Executing db2batch 2,095,129 271 11,916 0
400(*) 6.04% 7.09% 10.24% UOW Executing db2batch 1,841,415 106 9,008 0
410(*) 3.97% 4.40% 6.42% UOW Executing db2batch 873,760 109 5,586 1
417(*) 3.14% 3.55% 10.15% UOW Executing db2batch 723,275 78 4,506 0
401(*) 3.10% 3.49% 3.72% UOW Executing db2batch 694,860 169 4,436 0
413(*) 2.89% 3.74% 4.73% UOW Executing db2batch 701,656 31 4,757 0
406(*) 2.60% 0.21% 1.84% UOW Executing db2batch 93,664 438 262 0
1(*) N/A N/A N/A Setup/load/build[*] : 99% Load 200,000,007 0 88 0
407(*) 0.97% 0.07% 1.85% UOW Executing db2batch 45,991 0 64 0
405(*) 0.76% 0.05% 1.80% UOW Executing db2batch 25,943 0 89 0
402(*) 0.52% 0.07% 0.24% UOW Executing db2batch 16,851 173 107 0
419(*) 0.46% 0.08% 0.33% UOW Executing db2batch 63,928 6 427 0
411(*) 0.05% 0.34% 0.35% UOW Executing db2batch 569 6 0 0
378(*) 0.00% 0.00% 1.12% UOW Executing db2lused 0 0 0 0
379(*) 0.00% 0.00% 0.01% UOW Executing db2Fw0 0 0 0 0
380(*) 0.00% 0.00% 0.01% UOW Executing db2Fw1 0 0 0 0
381(*) 0.00% 0.00% 0.01% UOW Executing db2Fw2 0 0 0 0
382(*) 0.00% 0.00% 0.01% UOW Executing db2Fw3 0 0 0 0
383(*) 0.00% 0.00% 0.01% UOW Executing db2Fw4 0 0 0 0
384(*) 0.00% 0.00% 0.01% UOW Executing db2Fw5 0 0 0 0
385(*) 0.00% 0.00% 0.01% UOW Executing db2Fw6 0 0 0 0
386(*) 0.00% 0.00% 0.01% UOW Executing db2Fw7 0 0 0 0
387(*) 0.00% 0.00% 0.01% UOW Executing db2Fw8 0 0 0 0
388(*) 0.00% 0.00% 0.01% UOW Executing db2Fw9 0 0 0 0
389(*) 0.00% 0.00% 0.01% UOW Executing db2Fw10 0 0 0 0
390(*) 0.00% 0.00% 0.01% UOW Executing db2Fw11 0 0 0 0
391(*) 0.00% 0.00% 0.01% UOW Executing db2Fw12 0 0 0 0
392(*) 0.00% 0.00% 0.01% UOW Executing db2Fw13 0 0 0 0
393(*) 0.00% 0.00% 0.01% UOW Executing db2Fw14 0 0 0 0
-----
Quit: q, Help: h 161 active session(s) out of 166 db2top 2.0
  
```

Figure 6-4 db2top Sessions screen

In order to see details of the application execution, press **a** (for agent ID). Enter **456** when prompted with “Please enter agent id:.” You get further details about the SQL statement being executed by the agent.

Figure 6-5 shows further details on the actual application running the SQL statement, including further information about the application (Application name, client PID, DB2 user), and various statistics such as cpu elapsed time, sorts.

```
(*)18:14:46,refresh=4!secs(0.010) Sessions Linux,part=[9/9],BCULINX:BCUKIN
[d=Y,a=N,e=N,p=ALL] [qp=off]

*N0.bcullinux.101001192140, UOW Executing

  ConnTime...   14:21:40.557  UOW Start..   14:22:27.813  Appl name..       db2bp  DB2 user..       BCULINX
  OS user...    bcullinux   Agent id...    456   Coord DBP..      0     Coord id..      309
  Client pid..  6273      Hash joins..   0     Hash loops..     0     HJoin ovf..     0
  SQL Stmt...   4         Sorts.....   20    Sort time..     11850.805  Sorts ovf..     0
  Rows Read..  16,638,210,967  Rows Sel...   9     Read/Sel...    1,848,690,107  Rows Wrtn..    0
  Rows Ins...   0         Rows Upd...   0     Rows Del...     0     Locks held:     215
  Trans.....   0         Open Curs...  1     Rem Cursor..    0     Memory.....    1.8M
  Dyn. SQL...   16       Static SQL... 0     Cpu Time...     11665.5802  AvgCpuStmt:    729.987

-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  Start.....   17:50:30.273  Stop.....    18:08:36.458  Cpu Time...     3247.92473  Elapse.....    1086.18394
  FetchCount.. 0         Cost Est...   17,793,351   Card Est...     54   AgentTop...    1
  SortTime...  2957489   SortOvf...   0         Sorts.....   4     Degree.....    1
  Agents....   9         l_reads....  22,046,042   p_reads...     5,202,819  DataReads...  22,046,042
  IndexReads: 0         TempReads.. 0         HitRatio...   76.40%  MaxDbpCpu..   3248.00021[0]
  IntrRowsDel: 0         IntrRowsUpd: 0         IntrRowsIns: 0

Sub      Cpu  Cpu  Row      Rows      Rows      TqRows      TqRows      Tq      Exec # of      SubSection Waiting
Sec      (Sys+Usr)  Skew  Skew     Read     Written   Read     Written   Spills  Memory  Time  DBP  Ag.  Status TQueue(s)
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  0      0.128  0%  0%      0         0         0         0         0         192.0K  1456  1  1  R1      q1
  1    3248.086  31%  21%   4718749603      0         0         16         0         2.5M   1456  8  4  C4  E4

select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price, sum(l_extendedprice * (1 - l_discount)) as sum_d
isc_price, sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge, avg(l_quantity) as avg_qty, avg(l_extendedprice) as avg_price, avg(l_d
iscount) as avg_disc, count_big(*) as count_order from tpod.lineitem where l_shipdate <= date ('1998-12-01') - 90 day group by l_returnflag, l_linestatu
s order by l_returnflag, l_linestatus

Quit: q, Help: h          Tot cpu 3248.00021, associated memory 1.6M (total 1.8M), enter to refresh          db2top 2.0
```

Figure 6-5 Application details window

At this screen, we can get further details about the various agents associated to this application by pressing **d**.

Figure 6-6 shows all the agents associated to the application handle 456 on all the nodes. You can see that the agent TID 698 is associated to the application on database partition 1. So, we have matched the thread with the highest CPU usage with an application, and the SQL statement being executed by the application.

Sessions				
Assoc	Agent Tid	Node Number	Memory Size	Pool Number
Yes	309	0	192.0K	1
Yes	698	1	320.0K	1
Yes	773	6	384.0K	1
Yes	747	8	384.0K	1
Total			1.2M	
No	694	2	192.0K	1
No	857	3	192.0K	1
No	881	4	192.0K	1
No	694	5	192.0K	1
No	754	7	256.0K	1
Total			1.0M	

Figure 6-6 db2top associated agents screen

You can generate detailed query optimizer information, including an access plan, using the **db2exfmt** explain tool if you press **x**. Further information about the **db2exfmt** tool can be found at the following link:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0002353.html>

Figure 6-7 shows the **db2exfmt** output, which is edited using the **vi** editor. At the bottom of the file, you can see that the file is saved under **/tmp/explain.<nnnn>**. Press **:wq** to exit the **vi** editor and save the file.

Press **r** to return to the previous Sessions screen.

```
DB2 Universal Database Version 9.7, 5622-044 (c) Copyright IBM Corp. 1991, 2008
Licensed Material - Program Property of IBM
IBM DATABASE 2 Explain Table Format Tool

***** EXPLAIN INSTANCE *****

DB2_VERSION:      09.07.2
SOURCE_NAME:      SQLC2H21
SOURCE_SCHEMA:    NULLID
SOURCE_VERSION:
EXPLAIN_TIME:    2010-10-01-16.57.29.706785
EXPLAIN_REQUESTER: BCU LINUX

Database Context:
-----
Parallelism:      Inter-Partition Parallelism
CPU Speed:        2.360000e-07
Comm Speed:       100
Buffer Pool size: 180200
Sort Heap size:   12000
Database Heap size: 1200
Lock List size:   16384
Maximum Lock List: 10
Average Applications: 1
Locks Available:  52428

Package Context:
-----
SQL Type:         Dynamic
Optimization Level: 7
Blocking:         Block All Cursors
Isolation Level:  Cursor Stability

----- STATEMENT 1 SECTION 201 -----
QUERYNO:          1
QUERYTAG:         CLP
Statement Type:   Select
Updatable:        No
Deletable:        No
Query Degree:     1

Original Statement:
"/tmp/explain.5755" 2181L, 54798C                                     1,1      Top
```

Figure 6-7 *db2exfmt* output for longest running SQL

If you want to further identify the application running the culprit SQL statement, you can press **S** and capture a global application snapshot.

Figure 6-8 shows the global application snapshot containing all the information necessary to isolate the application submitting this SQL, which includes the application name, the inbound IP address, and the connect Authorization ID.

```

set snapshot for application agentid 456 global

Application Snapshot
Application handle           = 456
Application status          = UOW Executing
Status change time         = 10/01/2010 17:55:55.180635
Application code page      = 1208
Application country/region code = 1
DUOW correlation token     = *N0.bculinux.101001192140
Application name           = db2bp
Application ID              = *N0.bculinux.101001192140
Sequence number            = 00001
FP Monitor client user ID  =
FP Monitor client workstation name =
FP Monitor client application name = CLP qtext1.sql
FP Monitor client accounting string =

Connection request start timestamp = 10/01/2010 14:21:40.557395
Connect request completion timestamp = 10/01/2010 14:21:40.557824
Application idle time       =
CONNECT Authorization ID   = BCULINUX
Client login ID            = bculinux
Configuration NNAME of client = ISAS56R1D1
Client database manager product ID = SQL09072
Process ID of client application = 6273
Platform of client application = LINUXAMD64
Communication protocol of client = Local Client
Inbound communication address = *N0.bculinux

Database name              = BCURIT
Database path              = /db2fs/bculinux/NODE0000/SQL00001/
Client database alias      = BCURIT
Input database alias       =
Last reset timestamp      =
Snapshot timestamp        = 10/01/2010 17:55:59.305014
Authorization level granted =
User authority:
  DBADM authority
  SECADM authority
  DATAACCESS authority
  ACCESSCTRL authority
Group authority:
  SYSADM authority
  CREATETAB authority
"/tmp/snapshot.7508" 544L, 24930C
1,1 Top

```

Figure 6-8 Global application snapshot

An alternate way is to look directly for top consuming SQL. From the welcome screen, you can press **D** for Dynamic SQL. You will see all the statements executed on the system. Press **z** to sort per descending column order, followed by the column number five for CPU time.

Figure 6-9 shows the SQL screen. After sorting, we recognize the top consuming SQL, in terms of CPU time in the top row.

```

[//]15:00:04,refresh=2secs(0.008)                               SQL                               Linux,part=[9/9],BCULINUX:BCUKIT
[d=N,a=N,e=N,p=ALL]                                             [qp=off]
Enter SQL hash string: 00000011967827008915350482
SQL Statement          Sql          Num          Exec          Avg          Cpu          Avg          Rows
HashValue             Statement (30 first char.) Execution      Time      ExecTime      Time      CpuTime      read
-----
00000011967827008915350482 select l_returnflag, l_linesta  9 5885.02191 653.891323 2072.73764 230.304182 2979845723
00000014625833282084814102 select o_orderpriority, coun  9 12162.3758 1351.37508 1615.32700 179.480778 3785800744
00000016826973971099182528 select l_orderkey, sum(l_ext  9 6371.40978 707.934420 1590.91742 176.768602 2096327612
00000012478331566011111355 select c_custkey, c_name, s  9 20741.0745 2304.56383 1497.20398 166.355998 3161680489
00000016576471327228927760 select 100.00 * sum(case wh  9 737.319359 81.924373 149.856837 16.650759 138,829,141
00000002049266262197181246 select 100.00 * sum(case wh  9 723.014529 80.334947 146.794044 16.310449 138,825,035
00000016559989360269107959 select c_custkey, c_name, s  9 17879.7595 1986.63994 1370.77556 152.308395 3158858279
00000014934050767620923078 select sum(l_extendedprice *  9 366.804031 40.756003 120.715854 13.412872 456,307,872
00000001857365641925831948 select n_name, sum(l_extende  9 0.399899 0.044433 0.399775 0.044419 0
00000010113130495587294550 select o_year, sum(case wh  9 0.290126 0.032236 0.290038 0.032226 0
00000001388160218809444373 select supp_nation, cust_nat  9 0.251603 0.027955 0.251526 0.027947 0
00000017943188655970957270 select supp_nation, cust_nat  9 1.879993 0.208888 0.227180 0.025242 0
00000007714669665519719484 select nation, o_year, sum(  9 0.217027 0.024114 0.216970 0.024107 0
00000011058504738675622298 select nation, o_year, sum(  9 2.145332 0.238370 0.140141 0.015571 3
00000000864765591625414478 select s_acctbal, s_name, n_na  10 1021.33413 102.133413 0.103814 0.010381 10
0000009668008295969272269 select s_name, s_address fro  9 0.040719 0.004524 0.040706 0.004522 0
0000009452366330990901415 select c_name, c_custkey, o  9 1.542327 0.171369 0.014197 0.001577 1
00000010567183381451522844 with revenue (supplier_no, tot  9 0.007424 0.000824 0.007419 0.000824 0
0000007975626770754306051 select sum(l_extendedprice* (  9 0.007112 0.000790 0.007107 0.000789 0
00000011537732487362555236 select p_brand, p_type, p_s  9 0.006123 0.000680 0.006118 0.000679 0
00000008381702381420925853 select p_brand, p_type, p_s  9 0.006107 0.000678 0.006102 0.000678 0
0000004265983134564085415 select c_count, count(*) as  9 0.005405 0.000600 0.005379 0.000597 0
0000004354003115962724009 SELECT CURRENT QUERY OPTIMIZAT  20 0.004412 0.000220 0.003615 0.000180 2
00000002519323609090988725 SET CURRENT OPTIMIZATION PROFI  1 0.002207 0.002207 0.001870 0.001870 1
Quit: q, Help: h                               Dynamic SQL 241 (Cached=241), L: Query Text                               db2top 2.0

```

Figure 6-9 db2stop SQL screen shot

You can then press *L* to obtain further details about the SQL statement. As shown in Figure 6-9, *db2top* prompts for the SQL hash string shown, which is located on the first column.

You get the actual statement as shown in Figure 6-10 after you enter the SQL hash string. Press *x* to get its access plan for further review.

If the application running the SQL statement is still running, you can use the Sessions screen to identify the top CPU consuming application, and get further details about the application.

- ▶ DB2 9.7 relational monitoring interface:
 - You also can match the agent EDU ID from the **db2pd -edus** output to the AGENT_TID column in the WLM_GET_SERVICE_CLASS_AGENTS_V97 table function output to determine details about the application, including the application handle, and, if relevant, what request and statement the thread is working on. You can use the EXECUTABLE_ID returned by WLM_GET_SERVICE_CLASS_AGENTS_V97 to generate the actual access plan of the statement being executed by the agent, as described in the DB2 9.7 Information Center:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.sql.rtn.doc/doc/r0056251.html>

Utilities consuming CPU

The same method applies to identify the threads consuming CPU at the beginning of the investigation. Example 6-36 shows the **ps** command output and the **db2pd** command to check for the threads consuming the most CPU in this case.

Example 6-36 Identify the thread consuming the most CPU

```
# ps -Lm -Fp 623 | sort -rn +4 |more
bcu linux 623 618 - 75 126 3533942 6161004 - Sep30 ? 16:15:17 db2sysc 1
bcu linux - - 22326 20 - - - 10 20:09 - 00:01:02 -
bcu linux - - 22324 15 - - - 13 20:09 - 00:00:47 -
bcu linux - - 22323 15 - - - 13 20:09 - 00:00:47 -
bcu linux - - 22322 15 - - - 13 20:09 - 00:00:47 -
bcu linux - - 22321 15 - - - 13 20:09 - 00:00:47 -
bcu linux - - 6644 11 - - - 8 12:58 - 00:51:56 -
bcu linux - - 6778 7 - - - 0 12:58 - 00:34:40 -
bcu linux - - 6701 6 - - - 9 12:58 - 00:26:49 -
bcu linux - - 6615 6 - - - 9 12:58 - 00:27:06 -
bcu linux - - 6793 5 - - - 7 12:58 - 00:24:14 -
bcu linux - - 6771 5 - - - 5 12:58 - 00:23:47 -
bcu linux - - 6631 5 - - - 14 12:58 - 00:22:26 -
bcu linux - - 6617 5 - - - 7 12:58 - 00:24:38 -
bcu linux - - 6807 4 - - - 15 12:58 - 00:18:46 -
[...]
```

```
# db2pd -edus | grep 2232
973 47790205954368 22329 db21bm2 1 0.040000 0.080000
972 47790214342976 22328 db21bm1 1 0.050000 0.110000
971 47790218537280 22327 db21bm0 1 0.030000 0.100000
970 47790176594240 22326 db21rid 1 78.300000 2.260000
969 47790189177152 22325 db21mr 1 5.700000 9.030000
968 47790193371456 22324 db21frm3 1 58.430000 2.860000
967 47790197565760 22323 db21frm2 1 58.330000 3.090000
966 47790180788544 22322 db21frm1 1 58.590000 3.030000
965 47790184982848 22321 db21frm0 1 58.130000 3.000000
```

In this example, we can see that the top consuming thread is the **db21rid** process with EDU thread ID 970, as well as certain **db21frmX** threads. These threads are related to the LOAD utility. The **db21frmX** threads format the records from the flat file into an internal record format.

Because the high CPU usage is related to a utility, you can press **u** to get details about utilities running on the system.

Figure 6-11 shows that there is LOAD utility running on the system.

```

[~] 3:46:29, refresh=2secs(0.003) Utilities Linux, part=[9/9], BCUKIT:BCUKIT
[d=Y,a=N,e=N,p=ALL] [qp=off]

Hash      # of  Utility      Utility      Uti Utility  Invoker  Completed  Work      Phase      Progress
Value     entries Start Time   Type         Pri State  Type      Work      Unit      Prog% Start Time  Description
-----
5720862   8+ 18:27:48.033098 Load         0 Execute  User      7 Indexes  87% 18:34:34.651875 BUILD

Quit: q, Help: h                                     ISAS56R1D1                                     db2top 2.0

```

Figure 6-11 db2top Utilities screen

You can get further details about the utility by running the command **LIST UTILITIES SHOW DETAIL**. Example 6-37 shows an excerpt of the command output, ran from the catalog partition, which lists the status of the LOAD command on all the partitions.

Example 6-37 DB2 list utilities show detail output

```

# db2 list utilities show detail
list utilities show detail

ID = 1
Type = LOAD
Database Name = BCUKIT
Partition Number = 1
Description = OFFLINE LOAD CURSOR AUTOMATIC INDEXING
              REPLACE NON-RECOVERABLE TPCD .PARTSKW
Start Time = 10/01/2010 18:27:48.033098
State = Executing
Invocation Type = User
Progress Monitoring:
  Phase Number = 1
  Description = SETUP
  Total Work = 0 bytes
  Completed Work = 0 bytes
  Start Time = 10/01/2010 18:27:48.033108

  Phase Number = 2
  Description = LOAD
  Total Work = 40006935 rows
  Completed Work = 40006935 rows
  Start Time = 10/01/2010 18:27:52.134872

  Phase Number [Current] = 3
  Description = BUILD
  Total Work = 1 indexes
  Completed Work = 1 indexes
  Start Time = 10/01/2010 18:34:35.651875

ID = 1
Type = LOAD
Database Name = BCUKIT
Partition Number = 2
Description = OFFLINE LOAD CURSOR AUTOMATIC INDEXING

```

```

                                REPLACE NON-RECOVERABLE TPCD .PARTSKW
Start Time                       = 10/01/2010 18:27:48.334842
State                             = Executing
Invocation Type                   = User
Progress Monitoring:
  Phase Number                     = 1
    Description                     = SETUP
    Total Work                      = 0 bytes
    Completed Work                  = 0 bytes
    Start Time                      = 10/01/2010 18:27:48.334851

  Phase Number                     = 2
    Description                     = LOAD
    Total Work                      = 39997915 rows
    Completed Work                  = 39997915 rows
    Start Time                      = 10/01/2010 18:27:51.835023

  Phase Number [Current]          = 3
    Description                     = BUILD
    Total Work                      = 1 indexes
    Completed Work                  = 1 indexes
    Start Time                      = 10/01/2010 18:34:34.547275
[...]
```

After you have identified the LOAD job, you have various options to minimize its impact on the system including:

- ▶ Adjust the number of **db21frmX** processes on each database partition by setting CPU_PARALLELISM option. In this particular example, the 5600 V1 has 16 logical CPUs per data node. Each data node has four logical database partitions. So, DB2 spawns a total of 16 **db21frmX**, with 4 **db21frm** per logical partition. You can reduce this number to get a smaller number of threads per partition.
- ▶ Reduce the priority of LOAD using DB2 workload manager if implemented.

Other DB2 utilities such as BACKUP or RUNSTATS consuming an excessive amount of CPU, can be throttled dynamically using the command SET UTIL_IMPACT_PRIORITY. This command is documented in the DB2 9.7 Information Center:

<http://publib.boulder.ibm.com/infocenter/db21uw/v9r7/topic/com.ibm.db2.1uw.admin.cmd.doc/doc/r0011773.html>

Note that the same command as shown previously can be used to limit the impact of ASYNCHRONOUS INDEX CLEANUP following an ALTER TABLE... DETACH PARTITION statement, for range partitioned tables containing global indexes. By default, this utility has an impact limited to 50, but can be reduced in case it is still disruptive to the production system. Consult the following link for further information about ASYNCHRONOUS INDEX CLEANUP:

<http://publib.boulder.ibm.com/infocenter/db21uw/v9r7/topic/com.ibm.db2.1uw.admin.perf.doc/doc/c0021597.html>

Instead of setting the priority at each individual utility level, all utilities running within the instance can be throttled using the database manager configuration parameter `UTIL_IMPACT_LIM`:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.config.doc/doc/r0010968.html>

Other activities consuming CPU

Most high CPU situations begin with identifying the top three or five threads consuming CPU on the system. There are situations where the thread consuming CPU is not directly associated to an application or a utility running on a system. This is the case for threads belonging to the DB2 engine, which performs tasks for the entire workload running on the system.

In this case, you need to review thoroughly the entire workload running on the system to see if anything unusual is running on the system. As shown in Example 6-32, you can check with the `db2pd -edus -a11dbp` command to see if there is anything unusual that stands out on the system, and verify at the OS level that the thread is still consuming a high amount of CPU. You can then use tools such as `db2top` or the `LIST UTILITY SHOW DETAIL` statement to determine if the thread consuming CPU can be associated to a particular activity.

For example, if the `db2loggw` thread (which writes transaction log records) is associated to an unusually high CPU consumption, you can review all transaction intensive applications running on your system such as a large `INSERT` or `DELETE` SQL statement.

6.3.2 I/O usage

In the previous section, we reviewed how to match a physical device with a DB2 file system. Recent IBM Smart Analytics System offerings using automatic storage have one file system per device dedicated to one logical database partition. If the high I/O activity is isolated to a single device, the mapping discussed in 6.2.2, “Disk I/O and block queue” on page 142 can help you identify the database partition associated with the high I/O activity.

It is important to isolate the scope of the high I/O activity. For the IBM Smart Analytics System, during a high I/O activity, in most cases, all the file systems associated with the automatic storage path will show that all the file systems have an equally high I/O activity. If the high I/O is isolated to a particular file system and to a particular group of file systems, it might be caused as follows:

- ▶ **Data skew:** Certain database partitions might always lag behind during query execution, or show a higher than average CPU and I/O usage than the rest of the database partitions. This specific scenario is discussed in 6.4, “Common scenario: Data skew” on page 195.

- ▶ Database partition group: The high I/O usage on certain file systems can correspond to a specific database partition group, on which you have a specific workload or utility running.
- ▶ Hardware issues: If all the file systems reside on the same external storage, the storage can be running with performance degraded, as a result of maintenance being run, or other hardware issues. You can check the IBM Storage Manager Console to see if there are any error messages related to that specific storage.

In this section, we discuss how to narrow down a high I/O activity seen on the operating system level down to the database object, and the application consuming I/O.

Application using high I/O

In the test case, the I/O usage reported appears higher than usual. The goal is to identify if there is anything unusual running on this system. A single SQL statement cannot be singled out, as being the culprit for an I/O saturation. Instead, it is generally a combination of concurrent workload causing the issue.

In order to better understand the workload, we can look at the number of applications connected and the queries being executed to see if anything looks unusual. To better understand where the most I/O is being done and the associated database objects (tables), we can identify the top three SQL statements, or database objects which are the most frequently accessed.

In Example 6-38, `vmstat` reports I/O wait time up to 76%, with a high number of threads in the block queue (b column). The system is I/O bound.

Example 6-38 vmstat output on I/O bound system

```
# vmstat 2
procs -----memory----- --swap--  -----io----- -system--  -----cpu-----
r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
1  63  0  17360300  1788424  42525572  0  0  1135  241  0  0  2  0  95  2  0
3  63  0  17360564  1788428  42525568  0  0  840760  35994  27857  129297  21  9  2  69  0
4  57  0  17361344  1788432  42525564  0  0  680720  20620  26648  125769  17  8  1  74  0
6  64  0  17360656  1788444  42525552  0  0  693600  22538  27904  129783  16  8  2  74  0
0  69  0  17362144  1788444  42525552  0  0  698272  22284  28463  132483  16  9  2  73  0
2  61  0  17362516  1788444  42525552  0  0  741072  19504  27668  128692  16  8  3  73  0
6  47  0  17363320  1788448  42525548  0  0  683792  2572  27666  127690  18  8  2  72  0
3  51  0  17363700  1788448  42525548  0  0  852912  16406  31498  140055  21  9  3  67  0
3  56  0  17363416  1788456  42525540  0  0  610752  26616  23410  116020  15  7  2  76  0
5  53  0  17364028  1788456  42525540  0  0  805120  13354  25128  116890  20  8  3  69  0
4  57  0  17364160  1788456  42525540  0  0  666080  26168  24629  119777  16  8  6  70  0
1  62  0  17364040  1788468  42525528  0  0  665816  24212  26898  127206  16  8  2  74  0
4  60  0  17364672  1788468  42525528  0  0  610168  27854  24761  120212  15  7  2  76  0
[...]
```

Example 6-39 shows an **iostat** command excerpt with mostly read activity with block reads in the range of 457K to 928K versus 11K to 22K for blocks written in a 2-second interval.

Example 6-39 iostat command output sample

```
# iostat 2
Linux 2.6.16.60-0.21-smp (ISAS56R1D2) 10/04/2010

[...]
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           39.46    0.00  10.17  47.04    0.00    3.34

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                  3.98          0.00          123.38         0          248
sdb                 4877.61       227836.82       5492.54       457952       11040
sdc                 5712.94       461882.59       8007.96       928384       16096
sdd                 4054.23       372155.22      10969.15       748032       22048
sde                 1712.94       303653.73       5504.48       610344       11064
dm-0                 1819.40       303601.99       5500.50       610240       11056
dm-1                 4155.72       372187.06      10969.15       748096       22048
dm-2                 5831.34       461611.94       8007.96       927840       16096
dm-3                 4949.75       227231.84       5078.61       456736       10208
dm-4                  4.48          0.00          35.82         0           72
dm-5                  0.00          0.00          0.00         0           0
dm-6                  2.49          0.00          19.90         0           40
dm-7                  0.00          0.00          0.00         0           0
sdf                  0.00          0.00          0.00         0           0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           41.47    0.00  10.31  44.47    0.00    3.75

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                  0.00          0.00          0.00         0           0
sdb                 5333.50       268688.00      13716.00      537376       27432
sdc                 5381.50       474768.00        656.00      949536       1312
sdd                 3464.00       477328.00         56.00      954656       112
sde                 1910.50       342320.00       6624.00      684640       13248
dm-0                 1981.50       342224.00       6624.00      684448       13248
dm-1                 3523.50       477264.00         52.00      954528       104
dm-2                 5454.00       474800.00       1008.00      949600       2016
dm-3                 5535.50       269776.00      14200.00     539552       28400
dm-4                  0.00          0.00          0.00         0           0
dm-5                  0.00          0.00          0.00         0           0
dm-6                  0.00          0.00          0.00         0           0
dm-7                  0.00          0.00          0.00         0           0
sdf                  0.00          0.00          0.00         0           0
```

Based this output, we are looking for applications driving a high read activity on the system. **db2top** can be used for that purpose. The method is fairly identical to the one used to identify the application consuming the highest CPU. **db2top** is started from the administration node using the following command:

```
db2top -d bcukit
```

On the welcome screen, press 1 to get to the Sessions screen. Press z to order the columns per the third column, which is I/O% total, and enter 2 when prompted for the column number for descending sort. Figure 6-12 shows the **db2top** screen. We can see that the application doing the most I/O is application handle 1497.

```

[//116:36:26,refresh=2secs(0.052)                               Sessions                               Linux,part=[9/9],BCU:LINUX:BCU:K1:
[d=Y,a=N,e=N,p=ALL]                                             [qp=off]

lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
Local sessions...:      21          x          25%x          50%x          75%x          100%x      Local in exec....:      21
Remote sessions...:   151          x          x          x          x          x          Remote in exec....:   147
FCM buffers sent...: 1545864601      xSys/User Cpu%  xu          x          x          x          Active utilities...:    0
FCM buffers rec...:  1159603190      xrt+w/sec%      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxw      x          FCM buffers LWM...:   74,486
Piped sorts req...:    5,901      mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj]      Agent overflows...:    0          Piped sorts acc...:   5,901
Max Agents...:         0          Agents stolen...:      4,696          Sort heap alloc...:   3.2G      Connections HRM...:   49
Sheapthres...:        20.5G      Max Sort Heap....:   1.0G          Post hash joins...:   0          Private memory...:   1.0G
Dynamic SQL stmts...: 3          Static SQL stmts...: 0          Direct reads.....:    0          Post sorts.....:     0
Bufferpool reads...: 147,387      Bufferpool writes:    3          Direct writes.....:   0          Direct writes.....:   0

Application  Cpu%  I/O%  Mem%  Application  Application  Delta  Delta  Delta  Delta
Handle (Stat) Total  Total  Total  Status      Name          RowsRead/s  RowsWritten/s  IOReads/s  IOWrites/s
-----
1497(*) 3.22% 27.13% 15.72% UOW Executing db2batch 35,281 0 39,985 1
1610(*) 5.53% 10.61% 0.08% UOW Executing db2bp 1,134,575 0 15,642 0
1605(*) 4.63% 10.53% 0.08% UOW Executing db2bp 1,132,182 0 15,526 0
1501(*) 14.53% 9.83% 25.56% UOW Executing db2batch 2,056,301 588 14,490 0
1514(*) 16.10% 9.33% 21.29% UOW Executing db2batch 2,086,952 423 13,752 1
1606(*) 4.51% 8.84% 0.08% UOW Executing db2bp 976,501 0 13,032 1
1601(*) 4.13% 8.46% 0.08% UOW Executing db2bp 919,673 0 12,476 0
1498(*) 10.31% 5.35% 3.86% UOW Executing db2batch 1,413,680 0 7,884 0
1513(*) 6.64% 4.43% 6.00% UOW Executing db2batch 959,959 0 6,528 0
1495(*) 22.61% 4.12% 20.82% UOW Executing db2batch 3,517 2,409 6,079 0
1502(*) 0.27% 0.83% 0.33% UOW Executing db2batch 6,192 4 1,223 0
1496(*) 3.03% 0.15% 1.76% UOW Executing db2batch 54,713 219 216 0
1504(*) 0.87% 0.08% 0.40% UOW Executing db2batch 5,854 72 114 0
1508(*) 0.76% 0.06% 0.29% UOW Executing db2batch 27,690 0 86 0
1503(*) 0.47% 0.05% 0.38% UOW Executing db2batch 2,656 35 67 0
1510(*) 0.66% 0.05% 0.23% UOW Executing db2batch 23,644 0 79 0
1511(*) 0.56% 0.04% 0.27% UOW Executing db2batch 39,943 0 66 0
1512(*) 0.62% 0.04% 0.24% UOW Executing db2batch 21,204 0 65 0
1509(*) 0.49% 0.03% 0.21% UOW Executing db2batch 14,699 0 51 0
1499(*) 0.02% 0.01% 0.04% UOW Executing db2batch 960 0 13 0
1507(*) 0.03% 0.01% 0.07% UOW Executing db2batch 1,101 0 13 0
1473(*) 0.00% 0.00% 0.63% UOW Executing db2lused 0 0 0 0
1474(*) 0.00% 0.00% 0.01% UOW Executing db2fw0 0 0 0 0
1475(*) 0.00% 0.00% 0.01% UOW Executing db2fw1 0 0 0 0
1476(*) 0.00% 0.00% 0.01% UOW Executing db2fw2 0 0 0 0
1477(*) 0.00% 0.00% 0.01% UOW Executing db2fw3 0 0 0 0
1478(*) 0.00% 0.00% 0.01% UOW Executing db2fw4 0 0 0 0
1479(*) 0.00% 0.00% 0.01% UOW Executing db2fw5 0 0 0 0
1480(*) 0.00% 0.00% 0.01% UOW Executing db2fw6 0 0 0 0
1481(*) 0.00% 0.00% 0.01% UOW Executing db2fw7 0 0 0 0
1482(*) 0.00% 0.00% 0.01% UOW Executing db2fw8 0 0 0 0

Quit: q, Help: h 168 active session(s) out of 172 db2top 2.0

```

Figure 6-12 db2top sessions screen display

We press **a** to get further details about the SQL statement ran by this application, and enter 1497 when prompted for the agent ID.

Figure 6-13 shows the details of the application. We notice the very large number of rows read by this application. It has read more than 25 billion rows. We can generate a `db2exfmt` output to see if the plan of the query has changed.

We can also verify if the table where most of the I/O occurs is used in the current SQL. In order to do this, the actual SQL statement gives all the tables in the FROM clause of the query.

```
[*]16:38:13,refresh=2secs(0.005) Sessions Linux,part=[9/9],BCULINUX:BCURIT
[d=Y,a=N,e=N,p=ALL] [qp=off]
*N0.bculinux.101004180207, UOW Executing

ConnTime.: 13:02:05.877 UOW Start.: 15:50:54.927 Appl name.: db2batch DB2 user.: BCULINUX
OS user.: bculinux Agent id.: 1497 Coord DBP.: 0 Coord id.: 324
Client pid: 737 Hash joins: 40 Hash loops: 0 HJoin ovf.: 24
SQL Stmts.: 5 Sorts.....: 16 Sort time.: 4241.644 Sorts ovf.: 0
Rows Read.: 25,042,630,743 Rows Sel.: 7 Read/Sel.: 3,577,518,677 Rows Wrtn.: 12,755,006
Rows Ins.: 0 Rows Upd.: 0 Rows Del.: 0 Locks held: 914
Trans.....: 27 Open Curs.: 1 Rem Cursor: 0 Memory.....: 470.9M
Dyn. SQL.: 12 Static SQL: 3 Cpu Time.: 12163.2263 AvgCpuStm: 810.881

-----+ Dynamic statement [Fetch] -----
Start.....: 15:50:55.133 Stop.....: 15:51:44.254 Cpu Time.: 2943.49624 Elapse....: 49.116358
FetchCount: 0 Cost Est.: 18,623,970 Card Est.: 81 AgentTop...: 1
SortTime...: 0 SortOvf...: 0 Sorts.....: 0 Degree.....: 1
Agents....: 9 l_reads...: 41,224,222 p_reads...: 5,381,691 DataReads.: 40,064,227
IndexReads: 2,490 TempReads.: 1,157,505 HitRatio...: 86.95% MaxDbpCpu.: 2945.00060[0]
IntRowsDel: 0 IntRowsUpd: 0 IntRowsIns: 0

-----+-----
Sub Cpu Cpu Row Rows Rows TqRows TqRows Tq Memory Exec # of SubSection Waiting
Sec (Sys+User) Skew Skew Read Written Read Written Spills Memory Time DBP Ag. Status TQueue(s)
-----+-----
0 0.219 0% 0% 0 0 0 0 0 192.0K 2838 1 1 R1 q1
1 4.519 17% 1% 10,000,000 6,825 0 0 0 720.0M 2838 8 8 R8 q2
2 47.547 15% 10% 9,100,831 0 24 0 0 159.0M 2838 8 8 E8
3 2844.685 16% 1% 6463813759 12,742,259 10,654,784 7,595,411 97,057 2.8M 2838 8 8 E8 q3
4 48.271 17% 1% 200,000,000 0 0 1,331,848 0 0 49 8 0 C8
5 0.154 0% 0% 10 0 0 5 0 0 0 1 0 C1
6 0.209 0% 0% 25 0 0 25 0 0 0 1 0 C1

select o_year, sum(case when nation = 'SAUDI ARABIA' then volume else 0 end) / sum(volume) as mkt_share from ( select year(o.orderdate) as
o_year, l_extendedprice * (1 - l_discount) as volume, n2.n_name as nation from tpcd.part, tpcd.supplier, tpcd.lineitem, tpcd.order
s, tpcd.customer, tpcd.nation n1, tpcd.nation n2, tpcd.region where p_partkey = l_partkey and s_suppkey = l_suppkey and l_orderk
ey = o_orderkey and o_custkey = c_custkey and c_nationkey = n1.n_nationkey and n1.n_regionkey = r_regionkey and r_name = 'MIDDLE EAST' a
nd s_nationkey = n2.n_nationkey and o_orderdate between date('1995-01-01') and date ('1996-12-31') and p_type = 'MEDIUM POLISHED STEEL' ) as all
_nations group by o_year order by o_year

Quit: q, Help: h Tot cpu 2945.00060, associated memory 469.4M (total 470.9M), enter to refresh db2top 2.0
```

Figure 6-13 db2top session details screen shot

We can press `r` to get back to the Sessions screen, and press `T` to identify the tables statistics, where most of the I/O is being done. Because the `iostat` showed that the nature of the I/O workload was mostly read (according to Example 6-39 on page 171), we can sort the columns by the number of rows read per second, which is the second column, by pressing `z` followed by `1`.

Figure 6-14 shows that the table on which the most I/O is being done is TPCD.LINEITEM, which is the largest table used in the previous query.

Note that depending on the case, we might have started the investigation by looking at the most frequently accessed tables.

DB2 buffer pool hit ratio

By default, the IBM Smart Analytics System uses a single unified buffer pool. Details of the IBM Smart Analytics System buffer pool design are discussed in Chapter 7, “Advanced configuration and tuning” on page 203. All relevant DB2 buffer pool metrics are related to the single BP16K buffer pool.

A good metric for buffer pool monitoring is the overall buffer pool hit ratio. As shown in Figure 6-15, **db2top** can be used to show a high level buffer pool ratio; you can get to the Bufferpool metrics by pressing **b**.

```

[ / ] 18:38:01, refresh=2secs(0.005)                               Bufferpools                               Linux, part=[9/9], SCODINUX=BCUR1
[d=Y, a=N, e=N, p=ALL]                                           [qp=off]

lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x IoType x x x x x x x x x x x x x x x x x x x x x x x x x x
x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x
x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x      x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq]
                                xData Hit% x-----
                                xIdx Hit% x-----
                                xTemp Hit% x-----
                                xrw/sec  x-----
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq]

Bufferpool      Delta      Delta      Hit      Async      Delta      Delta      Async      Delta      Async      Async      # of      BP      File
Name            l_reads/s  p_reads/s  Ratio%  Reads%  Writes/s  a_reads/s  ReadMs  a_writes/s  WrtMs  wrt%  DBP      Pages  Closed/s
-----
BP16K           166,117    68,696    58.65%  48.59%    5,820     33,381    0.00     5,769    0.00  99.12%  9      1612800  0
IBMDEFAULTBP   0           0         0.00%  0.00%     0         0         0.00     0         0.00  0.00%  9      9000     0
IBMSYSTEMBP16K 0           0         0.00%  0.00%     0         0         0.00     0         0.00  0.00%  9      144     0
IBMSYSTEMBP32K 0           0         0.00%  0.00%     0         0         0.00     0         0.00  0.00%  9      144     0
IBMSYSTEMBP4K  0           0         0.00%  0.00%     0         0         0.00     0         0.00  0.00%  9      144     0
IBMSYSTEMBP8K  0           0         0.00%  0.00%     0         0         0.00     0         0.00  0.00%  9      144     0

Quit: q, Help: h                                           ISAS56R1D1                               db2top 2.0
  
```

Figure 6-15 db2top Bufferpools screen shot

The DB2 9.7 relational monitoring function `MON_GET_BUFFERPOOL` provides detailed information about the buffer pool activity, including the buffer pool hit ratio. Example 6-40 shows an example of an SQL query returning the overall buffer pool hit ratio.

Example 6-40 Overall buffer pool hit ratio

```

WITH BPMETRICS AS
( SELECT bp_name, pool_data_l_reads + pool_temp_data_l_reads
+ pool_index_l_reads + pool_temp_index_l_reads + pool_xda_l_reads
+ pool_temp_xda_l_reads as logical_reads, pool_data_p_reads
+ pool_temp_data_p_reads + pool_index_p_reads + pool_temp_index_p_reads
+ pool_xda_p_reads + pool_temp_xda_p_reads as physical_reads,
pool_read_time, member
FROM TABLE(MON_GET_BUFFERPOOL('',-2)) AS METRICS)
SELECT MEMBER, VARCHAR(bp_name,20) AS bp_name,
logical_reads, physical_reads, pool_read_time,
CASE
WHEN logical_reads > 0
THEN DEC((1 - (FLOAT(physical_reads) / FLOAT(logical_reads))) * 100,5,2)
ELSE NULL END
AS HIT_RATIO
FROM BPMETRICS
WHERE BP_NAME not like 'IBM%'
ORDER BY MEMBER, BP_NAME
  
```

MEMBER	BP_NAME	LOGICAL_READS	PHYSICAL_READS	POOL_READ_TIME	HIT_RATIO
0	BP16K	2684	89	31	96.68
1	BP16K	219495951	58457595	270435893	73.36

2	BP16K	223792271	63126325	224470159	71.79
3	BP16K	223496129	65649024	184997732	70.62
4	BP16K	221308949	63640936	169667848	71.24
5	BP16K	221315888	63604146	195739067	71.26
6	BP16K	220618242	58597348	263506121	73.43
7	BP16K	220317104	61109821	243800130	72.26
8	BP16K	216830746	58440331	266063848	73.04

In a data warehousing environment, the buffer pool ratio can be very low due to the relatively large size of the table scans compared to the buffer pool size.

However, the previous result gives a good baseline on the amount of physical versus logical reads taking place on the database, for tracking purposes, and establishing a baseline. This can be useful to identify if an I/O bound system is the result of an increased number of physical reads, for example. If not, and the system is experiencing a high I/O wait, there might be other issues within the I/O subsystem where the I/O runs in degraded mode (due to hardware issues, for example).

Rows read per row returned

A metric that can help measure the efficiency of the I/O made by the application is the ratio of the rows read per rows returned. A high ratio might be an indication of poor access plan choices. Regular collection of this data can also help in establishing a baseline for your application I/O consumption pattern. A sudden degradation can be attributed to poor access plans.

The DB2 administrative view `MON_CONNECTION_SUMMARY` offers this metric, with the column `ROWS_READ_PER_ROWS_RETURNED`. Note that other relevant data related to the I/O returned by this administrative view include `IO_WAIT_TIME_PERCENT` and `TOTAL_BP_HIT_RATIO_PERCENT`.

“Hot” tables

In order to get a quick idea of the regular tables where most of the I/O is being done within your database, we can run an SQL query based on DB2 9.7 relational monitoring function `MON_GET_TABLE`, as shown in Example 6-41. We can see that the table `TPCD.LINEITEM` is the far most accessed table within this database. From a database administration perspective, we must ensure that the table is well maintained by running the `REORGCHK_TB_STATS` procedure, for example, to make sure that the table does not need a reorganization.

In certain production databases, many unused tables can accumulate. This can impact the performance of utilities such as `BACKUP`, and affect data placement within the table space. Note that these statistics apply since the last time the database was activated. So, it might be normal to see certain tables not having any usage yet. If the database has been active for a long time where all the workload on your system has gone through an entire cycle, and there are still a few unused tables, check with your application team if they can be dropped.

Generally, the preferred method is to rename the tables first for an entire workload cycle after consulting with the application team. This action helps ensure that no applications receive errors because they try to access the tables. After it has been verified, the tables can be dropped. See Example 6-41.

Example 6-41 “Hot” tables

```
SELECT SUBSTR(TABSHEMA,1,10) as TABSHEMA,
SUBSTR(TABNAME,1,15) as TABNAME, SUM(TABLE_SCANS) AS SUM_TSCANS,
SUM(ROWS_READ) AS SUM_RW_RD, SUM(ROWS_INSERTED) AS SUM_RW_INS,
SUM(ROWS_UPDATED) AS SUM_RW_UPD, SUM(ROWS_DELETED) AS SUM_RW_DEL
FROM TABLE(MON_GET_TABLE('', '-2'))
WHERE TAB_TYPE='USER_TABLE'
GROUP BY TABSHEMA,TABNAME ORDER BY SUM_TSCANS DESC
```

TABSHEMA	TABNAME	SUM_TSCANS	SUM_RW_RD	SUM_RW_INS	SUM_RW_UPD	SUM_RW_DEL
TPCD	LINEITEM	15963	144456035062	0	0	0
TPCD	ORDERS	6034	13649670205	0	0	0
TPCD	PART	328	8115490496	0	0	0
TPCD	PARTSUPP	176	17103799091	0	0	0
TPCD	SUPPLIER	88	125316398	0	0	0
TPCD	CUSTOMER	64	1319998703	0	0	0
BCULINUX	WLM_EVENT	0	0	215	0	0
BCULINUX	WLM_EVENT_CONTR	0	0	18	0	0
BCULINUX	WLM_EVENT_STMT	0	0	215	0	0
BCULINUX	WLM_EVENT_VALS	0	0	0	0	0
BCULINUX	WLM_STATS_CONTR	0	0	18	0	0
BCULINUX	WLM_STATS_HIST	0	0	0	0	0
BCULINUX	WLM_STATS_Q	0	0	0	0	0
BCULINUX	WLM_STATS_SC	0	0	0	0	0
BCULINUX	WLM_STATS_WC	0	0	0	0	0
BCULINUX	WLM_STATS_WL	0	0	0	0	0
BCULINUX	WLM_THRESH_CONT	0	0	18	0	0
BCULINUX	WLM_THRESH_VIOL	0	0	0	0	0
SYSTOOLS	OPT_PROFILE	0	1	0	0	0
TPCD	NATION	0	925	0	0	0
TPCD	REGION	0	25	0	0	0

21 record(s) selected.

Index usage

A query similar to those shown previously can be run to identify most used indexes, as well as unused ones, as shown in Example 6-42.

We can see that the most frequently accessed index is on TPCD.SUPPLIER. Also, you can see that there are indexes not being used at all. In this case, you can further check if the tables have appropriate indexes. In Example 6-41, we saw that TPCD.LINEITEM table had the most table scans, but not a single index access. In this case, we can run the DB2 Design advisor `db2adv is` utility to check if there are suggestions for a better index. There might be none. `db2adv is` is documented at the following link:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.cmd.doc/doc/r0002452.html>

Example 6-42 Index usage

```
SELECT SUBSTR(S.INDSCHEMA,1,10) AS INDSCHEMA,  
SUBSTR(S.INDNAME,1,15) AS INDNAME,  
SUBSTR(S.TABNAME,1,15) AS TABNAME,  
SUM(T.INDEX_SCANS) AS SUM_IX_SCANS,  
SUM(T.INDEX_ONLY_SCANS) AS SUM_IX_ONLY_SCANS  
FROM TABLE(MON_GET_INDEX('', '', -2)) as T,  
SYSCAT.INDEXES AS S WHERE T.TABSCHEMA = S.TABSCHEMA  
AND T.TABNAME = S.TABNAME AND T.IID = S.IID  
AND T.TABSCHEMA not like 'SYS%'  
GROUP BY S.INDSCHEMA,S.INDNAME,S.TABNAME  
ORDER BY SUM_IX_SCANS DESC
```

INDSCHEMA	INDNAME	TABNAME	SUM_IX_SCANS	SUM_IX_ONLY_SCANS
TPCD	S_NK	SUPPLIER	184	0
TPCD	N_NK	NATION	33	0
TPCD	C_NK	CUSTOMER	32	0
TPCD	PS_PKSK	PARTSUPP	16	16
TPCD	C_CK	CUSTOMER	8	8
TPCD	O_OK	ORDERS	8	0
TPCD	PS_PK	PARTSUPP	8	0
TPCD	S_SK	SUPPLIER	8	0
TPCD	N_RK	NATION	5	0
TPCD	R_RK	REGION	5	0
TPCD	L_OKLN	LINEITEM	0	0
TPCD	O_CK	ORDERS	0	0
TPCD	PS_SK	PARTSUPP	0	0
TPCD	PS_SKPK	PARTSUPP	0	0
TPCD	P_PK	PART	0	0

15 record(s) selected.

Prefetching and page cleaning

IBM Smart Analytics System prefetch related settings are discussed in 7.1.2, “DB2 configuration” on page 217. The SYSIBMADM.MON_BP_UTILIZATION administrative view provides relevant metrics, including these:

- ▶ The prefetch ratio represents the ratio of physical reads that were done using asynchronous I/O (prefetching).
- ▶ The unread prefetch metric represents the ratio of pages that were retrieved through prefetching, but were not used by the buffer pool.
- ▶ Percentage of synchronous writes represents the ratio of synchronous writes needed to be performed by agents to get more space to accommodate their own pages into the buffer pool. This number is generally very low when page cleaning is efficient. See further information about the related parameter CHNGPGS_THRESH set lower on the IBM Smart Analytics System 7700 environment in “Database configuration settings” on page 228.

Example 6-43 shows a usage example of the administrative view, SYSIBMADM.MON_BP_UTILIZATION.

Example 6-43 Index hit ration and prefetch ration from

```
SELECT SUBSTR(BP_NAME,1,10) AS BP_NAME,  
MEMBER, DATA_HIT_RATIO_PERCENT AS DATA_HIT_RATIO,  
INDEX_HIT_RATIO_PERCENT AS INDEX_HIT_RATIO,  
PREFETCH_RATIO_PERCENT AS PREFETCH_RATIO,  
ASYNC_NOT_READ_PERCENT AS UNRD_PREFETCH_RATIO,  
SYNC_WRITES_PERCENT AS SYNC_WRITES_RATIO  
FROM SYSIBMADM.MON_BP_UTILIZATION  
WHERE BP_NAME not like 'IBM%'  
ORDER BY BP_NAME, MEMBER
```

BP_NAME	MEMBER	DATA_HIT_RATIO	INDEX_HIT_RATIO	PREFETCH_RATIO	UNRD_PREFETCH_RATIO	SYNC_WRITES_RATIO
BP16K	0	99.29	98.86	0.00	-	-
BP16K	1	60.94	99.41	40.50	4.05	0.47
BP16K	2	53.36	99.37	68.60	1.60	0.41
BP16K	3	51.88	99.37	78.84	0.94	0.41
BP16K	4	52.49	99.36	71.64	1.34	0.44
BP16K	5	52.68	99.36	70.34	1.20	0.43
BP16K	6	59.24	99.38	46.99	3.28	0.46
BP16K	7	57.66	99.37	52.37	2.41	0.47
BP16K	8	58.47	99.37	48.64	2.92	0.46

9 record(s) selected.

Temporary table space I/O

The I/O usage of the temporary table space is a key metric to monitor and keep track of the temporary spill usage within your database. For an IBM Smart Analytics System that has SSD devices such as the IBM Smart Analytics System 5600 with the SSD option and the IBM Smart Analytics System 7700, it is essential to understand the system temporary table space usage of your database.

Consumers of the temporary table space include these possibilities:

- ▶ Sort spills: Sorts can be spilled to temporary table space during query processing. Sorts can also spill during index creation. These sorts spills can be monitored through the sort overflow metrics.
- ▶ Hash join spills: These spills occur when hash join data exceeds sortheap. These spills can be monitored through the hash join overflow metric.
- ▶ Optimizer TEMP operations: The optimizer Low level plan operator during query processing. The query access plan shows the TEMP operator, along with an estimated size of the spill. Cardinality under estimations on the access plan can cause a high temporary table space usage.
- ▶ Table queue spills: When the receiver end cannot receive table queue buffers fast enough, the table queue buffers are spilled to the temporary table space. The table queue spills can be monitored through the application snapshot metric.
- ▶ Utility usage: Utilities such as LOAD, REORG, or REDISTRIBUTE can use the temporary table space.

To find out more details about the application consuming temporary table space, we press any key to get back to the Tables screen, followed by a. When prompted for the agent ID, we enter 132. We can get further details about the number of sorts and hash join overflows. See Figure 6-18. For this particular query, it appears that it is not the problem. We do see that there are table queue spills, which are likely the source of the temporary table space usage. We can also get details about the SQL statement being executed. So, we can get an explanation in the output.

```
(*)15:19:17,refresh=3!secs(0.007) Tables Linux,part=[9/9] BCULINUX:BCUKI [qp=off]
[d=Y,a=N,e=N,p=ALL]
*NO.bcunix.101005190443, UOW Executing

ConnTime..: 14:04:33.364 UOW Start.: 14:04:33.365 Appl name.: db2batch DB2 user.: BCULINUX
OS user...: bcunix Agent id...: 125 Coord DBF.: 0 Coord id...: 138
Client pid: 24312 Hash joins: 0 Hash loops: 0 HJoin ovf.: 0
SQL Stmt...: 2 Sorts.....: 0 Sort time.: 0.000 Sorts ovf.: 0
Rows Read.: 1,698,396,585 Rows Sel...: 1 Read/Sel...: 1,698,396,585 Rows Wrtn.: 1,480,296,549
Rows Ins...: 0 Rows Upd...: 0 Rows Del...: 0 Locks held: 707
Trans.....: 0 Open Curs.: 1 Rem Cursor: 0 Memory...: 48.1M
Dyn. SQL...: 4 Static SQL: 0 Cpu Time...: 1899.42967 AvgCpuStmt: 474.857

-----+-----+-----+ Dynamic statement [Fetch] -----+-----+-----+
Start.....: 14:04:33.368 Stop.....: N/A Cpu Time...: 1899.42127 Elapse....: 0.000030
FetchCount: 0 Cost Est...: 17,536,707 Card Est...: 455,895 AgentTop...: 1
SortTime...: 0 SortOvf...: 0 Sorts.....: 0 Degree....: 1
Agents.....: 9 l_reads...: 10,128,055 p_reads...: 2,673,941 DataReads.: 7,377,456
IndexReads: 37,466 TempReads.: 2,713,133 HitRatio...: 73.60% MaxDbpCpu.: 1901.00062[0]
IntRowsDel: 0 IntRowsUpd: 0 IntRowsIns: 0

Sub Cpu Cpu Row Rows Rows TgRows TgRows Tq Exec # of SubSection Waiting
Sec (Sys+Usr) Skew Skew Read Written Read Written Spills Memory Time DBP Ag. Status TQueue(s)
-----
0 0.315 0% 0% 0 0 0 0 0 0 192.0K 4484 1 1 R1 q1
1 1.339 35% 0% 0 0 0 0 0 0 2.8M 4484 8 8 R8 q2
2 111.679 5% 0% 0 0 0 168,559,730 0 0 79.6M 4484 8 8 R8 q3
3 1788.294 33% 5% 1698396585 1480296549 0 198,157,949 401,679 7.5M 4484 8 8 E6 s2 q3

select c_count, count(*) as custdist from ( select c_custkey, count(o_orderkey) from tpod.customer left outer join tpod.orders on c
_custkey = o_custkey and o_comment not like '%express%accounts%' group by c_custkey ) as c_orders (c_custkey, c_count) group by c_count orde
r by custdist desc, c_count desc

Quit: q, Help: h Tot cpu 1901.00062, associated memory 48.1M (total 48.1M), enter to refresh db2top 2.0
```

Figure 6-18 Checking sort and hash join overflows

Here are relevant metrics to get an idea of the temporary table space usage:

- ▶ Sort, hash join, and table queue overflows made by the various applications:

We have shown how to monitor these metrics at the application level based on the application temporary table space usage. Further examples of tracking these overflows are discussed in 7.1.2, “DB2 configuration” on page 217.

- ▶ Maximum high water mark usage:

The maximum high water mark table space usage represents the highest water mark in table space usage reached since the database was activated. This metric allows you to check if your temporary table space is of sufficient size. Specifically, on the IBM Smart Analytics System 5600 and 7700,

this metric allows you to verify if the spills are contained within the SSD container. 7.1.2, “DB2 configuration” on page 217 has details on how to verify the temporary table space usage.

► Buffer pool temporary hit ratio:

The Table screen in **db2top** (Figure 6-16 on page 180) shows detailed information about the temporary table accesses in terms of rows read and written; the rows can be sorted by Rows read or Rows written per second. You can also use relational monitoring function `MON_GET_BUFFERPOOL` to get details on the buffer pool hit ratio for temporary table space, as shown in Example 6-44. This can give you an idea of the amount of physical reads versus logical reads for temporary data.

Example 6-44 Temporary buffer pool hit ratio

```
WITH BPMETRICS AS ( SELECT bp_name, pool_data_l_reads
+ pool_temp_data_l_reads + pool_index_l_reads
+ pool_temp_index_l_reads + pool_xda_l_reads
+ pool_temp_xda_l_reads as logical_reads,
pool_data_p_reads + pool_temp_data_p_reads
+ pool_index_p_reads + pool_temp_index_p_reads
+ pool_xda_p_reads + pool_temp_xda_p_reads as physical_reads,
pool_read_time, member
FROM TABLE(MON_GET_BUFFERPOOL('-',-2)) AS METRICS)
SELECT MEMBER, VARCHAR(bp_name,20) AS bp_name,
logical_reads, physical_reads, pool_read_time,
CASE
WHEN logical_reads > 0
THEN DEC((1 - (FLOAT(physical_reads) / FLOAT(logical_reads))) * 100,5,2)
ELSE NULL
END AS HIT_RATIO
FROM BPMETRICS
WHERE BP_NAME not like 'IBM%' ORDER BY MEMBER, BP_NAME
```

MEMBER	BP_NAME	LOGICAL_READS	PHYSICAL_READS	POOL_READ_TIME	HIT_RATIO
0	BP16K	18572	152	91	99.18
1	BP16K	337839353	84574232	370561738	74.96
2	BP16K	32576653	95117898	271783119	71.39
3	BP16K	323382702	94027348	202064324	70.92
4	BP16K	330858029	95877816	247567098	71.02
5	BP16K	333520520	96618758	257182659	71.03
6	BP16K	336701269	88682531	347805047	73.66
7	BP16K	338874468	91523392	324759633	72.99
8	BP16K	338250903	89991268	335470505	73.39

9 record(s) selected.

Relational monitoring function `MON_GET_TABLESPACE` can be used to track the amount of physical writes made to the temporary table space as well, according to the query shown in Example 6-45.

Example 6-45 Tracking physical writes on the temporary table space

```
SELECT SUBSTR(TBSP_NAME,1,10) AS TBSP_NAME, TBSP_ID,
SUM(POOL_TEMP_DATA_L_READS + POOL_TEMP_XDA_L_READS
+ POOL_TEMP_INDEX_L_READS) AS SUM_TEMP_LOG_RD,
SUM(pool_temp_data_p_reads + pool_temp_index_p_reads
+ pool_temp_xda_p_reads) AS SUM_TEMP_PHYS_RD,
SUM(POOL_DATA_WRITES + POOL_XDA_WRITES
```

```

+ POOL_INDEX_WRITES) AS SUM_TEMP_POOL_WRITES,
MAX(TBSP_MAX_PAGE_TOP) AS MAX_TBSP_MAX_PAGE_TOP
FROM TABLE(MON_GET_TABLESPACE(' ', -2))
WHERE TBSP_CONTENT_TYPE like '%TEMP'
GROUP BY TBSP_NAME, TBSP_ID
ORDER BY TBSP_NAME, TBSP_ID

```

```

TBSP_NAME  TBSP_ID                SUM_TEMP_LOG_RD      SUM_TEMP_PHYS_RD      ...
-----
TEMP16K                260                  201654246             48394124 ...

```

1 record(s) selected.

```

...SUM_TEMP_POOL_WRITES MAX_TBSP_MAX_PAGE_TOP
-----
...                61706796             2689088

```

► Temporary table compression:

DB2 9.7 can potentially use temporary tables compression for sort spills, optimizer TEMP operations, and table queue spills. You can measure how effective the compression is by running a **db2pd** with the **-temptable** flag.

In Example 6-46, we can see the following information for the first logical database partition on the first data node:

- There were 106 system temporary tables in total since the database was activated.
- Four out of 106 were eligible for compression (flagged by DB2 optimizer as being a candidate for compression), and one was actually compressed (triggered during runtime, based on criteria such as the size of the spill).
- A total of 853 MB were spilled. The compression ratio is around 10%. The compression ratio is defined as follows:

$$\text{Total Sys Temp Bytes Saved} / (\text{Total Sys Temp Bytes Saved} + \text{Total Sys Temp Bytes Stored})$$

Example 6-46 db2pd -db bcukit -temptable output

```

db2pd -db bcukit -temptable -alldbp
System Temp Table Stats:
  Number of System Temp Tables : 106
  Comp Eligible Sys Temps      : 4
  Compressed Sys Temps         : 1
  Total Sys Temp Bytes Stored   : 895165003
  Total Sys Temp Bytes Saved    : 101051400
  Total Sys Temp Compressed Rows : 61200
  Total Sys Temp Table Rows:    : 7811414

```

```

User Temp Table Stats:
  Number of User Temp Tables   : 0
  Comp Eligible User Temps     : 0
  Compressed User Temps        : 0
  Total User Temp Bytes Stored  : 0
  Total User Temp Bytes Saved   : 0
  Total User Temp Compressed Rows : 0

```

```

Total User Temp Table Rows:      : 0
System Temp Table Stats:
Number of System Temp Tables    : 113
Comp Eligible Sys Temps        : 10
Compressed Sys Temps           : 3
Total Sys Temp Bytes Stored     : 1001945997
Total Sys Temp Bytes Saved     : 1591836900
Total Sys Temp Compressed Rows : 702200
Total Sys Temp Table Rows:     : 7823364

```

```

User Temp Table Stats:
Number of User Temp Tables      : 0
Comp Eligible User Temps       : 0
Compressed User Temps          : 0
Total User Temp Bytes Stored    : 0
Total User Temp Bytes Saved    : 0
Total User Temp Compressed Rows : 0
Total User Temp Table Rows:    : 0
System Temp Table Stats:
Number of System Temp Tables    : 114
Comp Eligible Sys Temps        : 11
Compressed Sys Temps           : 7
Total Sys Temp Bytes Stored     : 1020964698
Total Sys Temp Bytes Saved     : 449107550
Total Sys Temp Compressed Rows : 1870400
Total Sys Temp Table Rows:     : 7841952

```

```

User Temp Table Stats:
Number of User Temp Tables      : 0
Comp Eligible User Temps       : 0
Compressed User Temps          : 0
Total User Temp Bytes Stored    : 0
Total User Temp Bytes Saved    : 0
Total User Temp Compressed Rows : 0
Total User Temp Table Rows:    : 0
System Temp Table Stats:
Number of System Temp Tables    : 109
Comp Eligible Sys Temps        : 6
Compressed Sys Temps           : 1
Total Sys Temp Bytes Stored     : 990281374
Total Sys Temp Bytes Saved     : 10051760
Total Sys Temp Compressed Rows : 61200
Total Sys Temp Table Rows:     : 7813753

```

```

User Temp Table Stats:
Number of User Temp Tables      : 0
Comp Eligible User Temps       : 0
Compressed User Temps          : 0
Total User Temp Bytes Stored    : 0
Total User Temp Bytes Saved    : 0
Total User Temp Compressed Rows : 0
Total User Temp Table Rows:    : 0

```

Utilities using high I/O

If you do not see any applications doing an excessive amount of I/O, check for utilities doing a high amount of I/O, such as LOAD or BACKUP. These utilities do not perform I/O using the DB2 buffer pool. Their I/O is tracked as direct reads and direct writes. These metrics are returned by MON_GET_BUFFERPOOL relational monitoring function as well as buffer pool snapshots.

As shown in Example 6-47, you can track the number of direct reads and writes, as well as the time it takes to perform those I/O operations. You can compute the number of direct read and write operations performed per request. As we can see in the example, in this case, there appears to be I/O done against specific database partitions 1, 2, 6, and 8 only.

Example 6-47 Direct reads and writes output

```

SELECT SUBSTR(BP_NAME,1,10) AS BP_NAME, MEMBER, DIRECT_READS,
CASE WHEN DIRECT_READ_REQS > 0
THEN INT(BIGINT(DIRECT_READS) / BIGINT (DIRECT_READ_REQS))
ELSE NULL END AS DIRECT_READ_PER_REQ,
DIRECT_READ_TIME, DIRECT_WRITES,
CASE WHEN DIRECT_WRITE_REQS > 0 THEN INT(BIGINT(DIRECT_WRITES) / BIGINT
(DIRECT_WRITE_REQS))
ELSE NULL END AS DIRECT_WRITE_PER_REQ,
DIRECT_WRITE_TIME FROM TABLE(MON_GET_BUFFERPOOL('-',-2))
WHERE BP_NAME not like 'IBM%'
ORDER BY BP_NAME, MEMBER

```

BP_NAME	MEMBER	DIRECT_READS	DIRECT_READ_PER_REQ	DIRECT_READ_TIME	...
BP16K	0	24	2	1	...
BP16K	1	480	32	126	...
BP16K	2	480	32	245	...
BP16K	3	256	32	76	...
BP16K	4	256	32	14	...
BP16K	5	256	32	7	...
BP16K	6	96	32	18	...
BP16K	7	256	32	5	...
BP16K	8	480	32	135	...

...	DIRECT_WRITES	DIRECT_WRITE_PER_REQ	DIRECT_WRITE_TIME
...	1080	77	254
...	7623552	955	396753
...	7617184	955	283409
...	1120	280	12
...	1120	280	9
...	1120	280	10
...	8779296	914	389882
...	1120	280	13
...	7596800	955	243858
...			

9 record(s) selected.

If you see a large number of direct reads and direct writes potentially affecting the I/O on your system, you can narrow down any utilities running on the system using the LIST UTILITIES SHOW DETAIL command, and throttle the utility.

6.3.3 DB2 memory usage

In order to investigate any issues related to memory on an IBM Smart Analytics System, it is essential to understand how DB2 is using memory on an IBM Smart Analytics System. In this section we discuss the various memory allocations done within DB2, in order to account for all the memory usage.

Global memory management parameters

Two global memory parameters are used to cap the memory usage on each server. These parameters are left set to AUTOMATIC by default:

► **INSTANCE_MEMORY:**

This database manager configuration parameter controls the maximum amount of memory that can be used by each DB2 logical database partition, including all shared memory and private memory usage for the agents associated to that particular logical database partition.

INSTANCE_MEMORY can be set either to AUTOMATIC or a specific value:

- If set to AUTOMATIC and SELF_TUNING_MEM (STMM) is ON, this parameter enables automatic tuning of instance memory according to the memory available at the operating system level. In this configuration, DB2 will consume between 75% and 95% of the RAM for all the database partitions within a server. However, because STMM is disabled by default for the 5600, 7600, and 7700 offerings, this parameter is ignored for IBM Smart Analytics System 5600, 7600, and 7700.
- If set to a specific value, this value will cap the amount of memory used by the logical database partition. This setting is not desirable for the IBM Smart Analytics System.

► **DATABASE_MEMORY:**

This database configuration parameter represents the maximum amount of memory usable for the database shared memory allocations. On DB2 9.7, this value is equal to individual memory pool allocations within the database shared memory set such as buffer pool, utility heap size with additional memory to accommodate for dynamic memory growth. This parameter is left to its default value of AUTOMATIC for the IBM Smart Analytics System. The following settings are allowed:

- **AUTOMATIC:** Default value. If STMM is enabled, DATABASE_MEMORY will be tuned automatically.
- **COMPUTED:** Values are calculated based on the sum of the database shared memory set and other heap settings during database startup. There is provisioning done for dynamic growth in the calculations. If STMM is disabled as in the IBM Smart Analytics System environments, AUTOMATIC and COMPUTED hold the same meaning.

- Specific value: The value will act as a cap for all database shared memory requirement. If this value cannot be allocated initially or is higher than INSTANCE_MEMORY database activation will fail.
- ▶ SELF_TUNING_MEM:

This database configuration parameter allows you to turn the DB2 Self-Tuning Memory Manager ON. This parameter is turned OFF by default with the IBM Smart Analytics System.

DB2 memory allocations

There are mainly two types of memory allocation within DB2:

- ▶ Shared memory allocations: There are various types of shared memory sets allocated within DB2 at the instance, database, and application level.
- ▶ Private memory allocations: Private memory allocated at each individual DB2 EDU level.

A summary of the DB2 memory usage statistics is provided by the command **db2pd -dbptnmem**, as shown in Example 6-48. It shows the Memory allocation limit for the database partition corresponding to INSTANCE_MEMORY (Memory Limit), and the current total memory consumption for the logical database partition (Current usage), and separates that into individual consumers, including the total application memory, instance shared memory set, private memory, FCM shared memory set, database shared memory, and FMP shared memory set, giving the current usage, the high watermark usage, and the cached memory usage for each set. In the following section, these memory sets are discussed in more detail.

Example 6-48 db2pd -dbptnmem

```
bculinux@ISAS56R1D2:~> db2pd -dbptnmem -dbp 1

Database Partition 1 -- Active -- Up 0 days 03:05:59 -- Date 10/06/2010
00:45:24

Database Partition Memory Controller Statistics

Controller Automatic: Y
Memory Limit:          14838876 KB
Current usage:         5474048 KB
HWM usage:             5954368 KB
Cached memory:        1062848 KB

Individual Memory Consumers:

Name                   Mem Used (KB) HWM Used (KB) Cached (KB)
=====
```

APPL-BCUKIT	160000	160000	150784
DBMS-bcu linux	34048	34048	1344
FMP_RESOURCES	22528	22528	0
PRIVATE	914880	1234368	361728
FCM_RESOURCES	355136	562560	0
DB-BCUKIT	3987200	3987200	548992
LCL-p9043	128	128	0
LCL-p9043	128	128	0

Note that the **db2pd -dbptnmem** command output includes additional virtual memory allocation to accommodate any potential growth, not just the actual system memory usage.

The table function `ADMIN_GET_DBP_MEM_USAGE` can also be used to show the instance memory usage as shown in Example 6-49.

Example 6-49 ADMIN_GET_DBP_MEM_USAGE

```
$ db2 "select * from table (sysproc.admin_get_dbp_mem_usage()) as t where
DBPARTITIONNUM=1"
```

DBPARTITIONNUM	MAX_PARTITION_MEM	CURRENT_PARTITION_MEM	PEAK_PARTITION_MEM
1	15195009024	5605425152	6097272832

1 record(s) selected.

Shared memory allocations

The following shared memory sets are allocated by DB2:

- ▶ Instance level shared memory sets: Allocated when the instance is started. When performing memory usage calculations, these segments are allocated once per server. For example, these allocations must not be counted multiple times for each logical database partition on the same server. This includes the following shared memory segments:
 - Database manager shared memory set: Includes memory allocations for heaps such as monitor heaps, and other internal heaps.
 - FCM shared memory set: Used for FCM resources memory allocation, such as FCM buffers, and channels. Generally the main memory consumer at the instance level for IBM Smart Analytics System.
 - FMP shared memory set: Shared memory segment allocated for communication with db2fmp threads.
 - Trace shared memory segment: Allocated for the trace segment for DB2 trace.

Instance-level shared memory sets allocations can be tracked using the `db2pd -inst -memsets` command, as shown in Figure 6-19.

```

bcu@linux@ISA856R1D2:~> db2pd -inst -memsets
Database Partition 1 -- Active -- Up 0 days 02:07:00 -- Date 10/05/2010 23:46:25

Memory Sets:
Name      Address      Id      Size (Kb)  Key      DBP      Type  Unrsv (Kb)  Used (Kb)  HWM (Kb)  Cmt (Kb)  Uncmt (Kb)
DBMS     0x0000000200000000  1505886209  34048    0xEFC9E461  1      0      1344      11392      11392      11392      22656
FMP      0x0000000210000000  1506017285  22592    0x0         0      2      0          448        448        22592      0
Trace    0x0000000000000000  1505853440  39240    0xEFC9E474  1      -1     0          39240     0          39240      0
FCM      0x0000000220000000  1506050054  4329088  0xEFC9E462  0      11     2901376   1427712   2250240   1427712   2901376
  
```

Figure 6-19 `db2pd -inst -memsets`

The Size (Kb) column represents the shared memory segment size. The Committed memory Cmt (Kb) column represents the amount of memory committed at the operating system level, and the HWM (Kb) is the highest memory usage in the pool since the instance was started. For an actual memory usage calculation, you can rely on the Cmt (Kb) column for the actual allocation. This shared memory is allocated at the server level.

You can further drill down the various memory pool allocations within each of these memory sets by using the `db2pd -inst -memsets -mempools` command, as shown in Figure 6-20. This output contains each individual memory pool allocation within each memory set. You can, for example, see the drill down for all memory allocations within the FCM shared memory set. The physical HWM sum for all memory pools within the set has to match approximately the HWM (Kb) column for the memory set in the output in Figure 6-19.

```

bcu@linux@ISA856R1D2:~> db2pd -inst -mempools
Database Partition 1 -- Active -- Up 0 days 02:20:44 -- Date 10/06/2010 00:00:09

Memory Pools:
Address      MemSet  PoolName  Id  Overhead  LogSz  LogUpBnd  LogHWM  PhySz  PhyUpBnd  PhyHWM  Bnd  BlkCnt  CfgParm
0x00000002000000C50  DBMS    fcm       74  0          659128  652502    659128  720896  655360  720896  Ovf  1003  n/a
0x00000002000000B08  DBMS    monh     11  122560    494354  368640    495994  655360  393216  655360  Ovf  286  MGN_HEAP_02
0x00000002000000C0C  DBMS    reasynch  62  0          19320    2752512   19320    65536  2752512  65536  Ovf  2    n/a
0x00000002000000878  DBMS    apmh     70  4512      2585268  7798784   2585652  3276800  7798784  3276800  Ovf  143  n/a
0x00000002000000730  DBMS    kexh     52  0          2124416  4128768   2124416  2424832  4128768  2424832  Ovf  223  n/a
0x000000020000005E8  DBMS    baah     71  0          1523595  14548992  1597326  1638400  14548992  1638400  Ovf  142  n/a
0x000000020000004A0  DBMS    sqlich   50  0          2685747  2686976   2685747  2686976  2686976  2686976  Ovf  203  n/a
0x00000002000000358  DBMS    krcbh    69  0          77944    131072    78120    131072  131072  131072  Ovf  15  n/a
0x00000002200000878  FCM     fcmseas  77  65376     32605280  14999552  32605280  32899072  15007744  32899072  Ovf  6  n/a
0x00000002200000730  FCM     fcmchan  19  65376     42490048  36651008  44265920  42860544  36700160  44695552  Ovf  12  n/a
0x000000022000005E8  FCM     fcmcbp  13  65376     1346010048  599711744  2189171648  1347551232  599719936  2191654912  Ovf  360  n/a
0x000000022000004A0  FCM     fcmctl  73  331584    6675452  20166280  6675452  7274496  20185088  7274496  Ovf  4053  n/a
0x00000002200000358  FCM     eduah    72  8000      27648024  27648064  27648024  27656192  27656192  27656192  Ovf  1  n/a
0x00000002100000358  FMP     undefh   59  24000     368700    22971520  368700    393216  23003136  393216  Phy  3  n/a
  
```

Figure 6-20 `db2pd -inst -memsets -mempools` output

- ▶ Database and application level shared memory sets: Database level shared memory sets are allocated when the database is activated. These shared memory segments are allocated for each database partition:
 - Database shared memory set: Generally the main memory consumer for IBM Smart Analytics System. Includes allocations for memory heaps such as buffer pools, package cache, locklist, utility heap, and database heap.

- Application shared memory set: Application group shared memory segment controlled by APPLICATION_MEMORY.
- Application control shared heap segments: Application level shared memory set. Not allocated when the database is activated, but allocated as needed, depending on the number of applications connected to the database partition.

Database level shared memory allocation set can be tracked for each logical database partition using the `db2pd -db <db-name> -memsets` command, as shown in Figure 6-21.

For example, for database partition 1, the output shows that the total of the memory committed is 3226112 KB, which is approximately 3 GB. The default memory database allocation is identical for the four logical database partitions. Therefore, we are using approximately 12 GB for database shared memory per server. This can be verified using the command `db2pd -db bcukit -memsets -alldbp | grep BCUKIT`. The application memory allocation amounts for approximately 36 MB for the entire database partition when the output is collected, which is negligible in this case.

```
bcuclinux@ISAS56R1D2:~> db2pd -db bcukit -memsets
```

Database Partition 1 -- Database BCUKIT -- Active -- Up 0 days 02:33:42 -- Date 10/06/2010 00:16:39

Memory Sets:

Name	Address	Id	Size(Kb)	Key	DBP	Type	Unrsv(Kb)	Used(Kb)	HWM(Kb)	Cmt(Kb)	Uncmt(Kb)
BCUKIT	0x00002AC1D5795000	1507000334	3987264	0x0	0	1	540736	3182912	3226112	3226112	761152
AppCt1	0x00002AC1CBB45000	1506869258	160064	0x0	0	12	0	8896	9280	9408	150656
App65618	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65611	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65624	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65617	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65610	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65623	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65616	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65609	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65622	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65668	0x000000005A298013	1512669203	128	0x0	0	4	0	128	0	128	0
App65615	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65608	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65621	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65614	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65620	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65613	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65619	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0
App65612	0x0000000000000000	0	0	0x0	0	0	0	0	0	0	0

Figure 6-21 Database shared memory set

Private memory allocations

The private memory allocations are made in a single large memory area per database partition, allocated from the **db2sysc** process private memory. At the OS level, this area of memory is shared by all the threads within the **db2sysc** process. At the DB2 level, DB2 manages this memory in thread specific memory pools. Private memory allocations includes all the private memory allocations made by the various DB2 EDUs within a **db2sysc** process. For the IBM Smart Analytics System, the main consumer is SORTHEAP. The default configuration uses private sorts, with a large SHEAPTHRES. See 7.1.2, “DB2 configuration” on page 217 for further details about sort configurations with the IBM Smart Analytics System.

To estimate the total amount of private memory allocations for all the DB2 agents for each database partition, we can use the data returned from **db2pd -dbptnmem** output, as shown previously in Example 6-48 on page 187.

The output shows that the total number of private memory allocations used is 914880 KB, which is approximately 893 MB. The memory used corresponds to the actual memory allocated. In order to account for all private memory allocations within this server, we need to add this value for the four logical partitions within the server, which can be obtained through the command, **db2pd -dbptnmem -alldbp**.

Memory usage calculation example

In this example, we perform an estimate of the memory used by the first data node, based on the outputs of commands reviewed previously.

- ▶ Instance level shared memory:

Figure 6-22 shows a sample output of **db2pd -inst -memsets**.

Based on this output, the total committed memory for instance level shared memory set is as follows:

Instance level shared memory Cmt(Kb) = 12480 + 22592 + 39240 + 1325568 = 1399880 Kb

```
Database Partition 1 -- Active -- Up 0 days 21:57:30 -- Date 10/06/2010 19:36:55
```

Memory Sets:											
Name	Address	Id	Size(Kb)	Key	DBP	Type	Unrsrv(Kb)	Used(Kb)	HWM(Kb)	Cmt(Kb)	Uncmt(Kb)
DBMS	0x0000000200000000	1505886209	34048	0xEFC9E461	1	0	1344	12480	12480	12480	21568
FMP	0x0000000210000000	1506017285	22592	0x0	0	0	2	0	448	22592	0
Trace	0x0000000000000000	1505853440	39240	0xEFC9E474	1	-1	0	39240	0	39240	0
FCM	0x0000000220000000	1506050054	4329088	0xEFC9E462	0	11	3003520	1325568	2250240	1325568	3003520

Figure 6-22 db2pd -inst -memsets output

- Database and application level shared memory:

Figure 6-23 shows the `db2pd -db bcukit -memsets` output for logical database partition 1.

```
bculinux@ISAS56R1D2:~> db2pd -db bcukit -memsets
Database Partition 1 -- Database BCUKIT -- Active -- Up 0 days 03:13:14 -- Date 10/06/2010 19:52:24

Memory Sets:
Name      Address      Id      Size(Kb)  Key      DBP      Type      Unrsv(Kb)  Used(Kb)  HWM(Kb)  Cmt(Kb)  Uncmt(Kb)
BCUKIT    0x00002AC1D5795000 1558577166 3987264  0x0      0        1        589056     3134656   3160832  3161984  825280
AppCtl1   0x00002AC1CBB45000 1558511628 160064   0x0      0        12       0          9216     9792     9984     150080
App66046  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66052  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66045  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66058  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66051  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66044  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66057  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66050  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66043  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66056  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66049  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66042  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66055  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66048  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66054  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66047  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
App66053  0x0000000000000000 0          0         0x0      0        0        0          0        0        0        0
```

Figure 6-23 Database level shared memory allocations

After issuing the command with the `-alldbp` flag, we have verified that there are no application control shared memory allocations “AppXXXXX” on any database partition, so we can filter it out with the following `db2pd` command:

```
db2pd -db bcukit -memsets -alldb | egrep "BCUKIT |AppCtl|Cmt" | grep -v Database
```

Figure 6-24 shows the output. The total database level shared memory is:

Database level shared memory Cmt (Kb) = 3161984 + 9984 + 3177536 + 9856 + 3151680 + 9600 + 3144128 + 9792 = 12674560 KB

```
bculinux@ISAS56R1D2:~> db2pd -db bcukit -memsets -alldb | egrep "BCUKIT |AppCtl|Cmt" | grep -v Database
Name      Address      Id      Size(Kb)  Key      DBP      Type      Unrsv(Kb)  Used(Kb)  HWM(Kb)  Cmt(Kb)  Uncmt(Kb)
BCUKIT    0x00002AC1D5795000 1558577166 3987264  0x0      0        1        602496     3121216   3160832  3161984  825280
AppCtl1   0x00002AC1CBB45000 1558511628 160064   0x0      0        12       0          9344     9792     9984     150080
Name      Address      Id      Size(Kb)  Key      DBP      Type      Unrsv(Kb)  Used(Kb)  HWM(Kb)  Cmt(Kb)  Uncmt(Kb)
BCUKIT    0x00002B6A0DA8D000 1558642704 3987264  0x0      0        1        661632     3062080   3174976  3177536  809728
AppCtl1   0x00002B6A03D3D000 1558446090 160064   0x0      0        12       0          9472     9728     9856     150208
Name      Address      Id      Size(Kb)  Key      DBP      Type      Unrsv(Kb)  Used(Kb)  HWM(Kb)  Cmt(Kb)  Uncmt(Kb)
BCUKIT    0x00002AE0F8ED2000 1558609935 3987264  0x0      0        1        662592     3056192   3151680  3151680  835584
AppCtl1   0x00002AE0EF182000 1558544397 160064   0x0      0        12       0          9280     9472     9600     150464
Name      Address      Id      Size(Kb)  Key      DBP      Type      Unrsv(Kb)  Used(Kb)  HWM(Kb)  Cmt(Kb)  Uncmt(Kb)
BCUKIT    0x00002B81E3ABD000 1558675473 3987264  0x0      0        1        662592     3059776   3143680  3144128  843136
AppCtl1   0x00002B81D9E6D000 1558478859 160064   0x0      0        12       0          9472     9792     9792     150272
```

Figure 6-24 Database level shared memory sets

► Private memory:

To get an estimate of the total private memory used, we can get an output of **db2pd -dbptnmem -alldbp**, as shown in Example 6-50.

Private Mem Used (Kb) = 770240 + 683584 + 600000 + 769792 = 2823616 KB

Example 6-50 Private memory

```
db2pd -dbptnmem -alldbp | egrep "PRIVATE|Used"
Name           Mem Used (KB) HWM Used (KB) Cached (KB)
PRIVATE        770240      1234368      436352
Name           Mem Used (KB) HWM Used (KB) Cached (KB)
PRIVATE        683584      1209024      357824
Name           Mem Used (KB) HWM Used (KB) Cached (KB)
PRIVATE        600000      1167616      262912
Name           Mem Used (KB) HWM Used (KB) Cached (KB)
PRIVATE        769792      1182208      440576
```

► Total Memory used:

From a DB2 perspective, the total amount of memory currently used on the system can be roughly accounted for as follows:

```
Total amount of memory used
= Instance level shared memory + Database level shared memory + Private memory used
= 1399880 KB + 12674560 KB + 2823616 KB
= 16898056 KB
= 16.1 GB
```

So, DB2 is approximately using 16 GB of memory out of 64 GB available on this system.

6.3.4 DB2 network usage

The main network usage with an IBM Smart Analytics System is generally with the DB2 FCM internal communications between database partitions. In order to understand the network usage for internal communications between database partitions and establish a baseline, you have to monitor the FCM activity with the **db2top** utility, which provides live information about the traffic in terms of buffers received and sent per second, on the Partitions Screen. You can get to the Partition screen by pressing **p**, as shown in Figure 6-25.

```

[~] 21:33:16,refresh=2secs(0.260) Partitions Linux,part=[9/9] BCU LINX:BCUR14
[d=Y,a=N,e=N,p=ALL] [qp=off]

```

Partition Number	Partition Status	Buffer LMM	Delta BufSent/s	Delta BufRcvd/s	Pool CurrSize	Pool HWM	Channels Free	Space Used	Total Space	Log Current	Log First
0*	Active	74525	149	416	3.2G	3.4G	2468	255.8M	116.3G	0	0
1	Active	294179	328	166	4.2G	5.1G	4979	188.5G	343.3G	0	0
2	Active	294179	251	174	4.2G	5.1G	4979	170.6G	343.2G	0	0
3	Active	294179	266	167	4.2G	5.1G	4979	166.7G	343.3G	0	0
4	Active	294179	258	171	4.2G	5.1G	4979	167.0G	343.2G	0	0
5	Active	294024	220	173	4.2G	5.1G	4979	166.3G	343.1G	0	0
6	Active	294024	358	178	4.2G	5.1G	4979	188.2G	343.2G	0	0
7	Active	294024	260	179	4.2G	5.2G	4979	181.0G	343.2G	0	0
8	Active	294024	257	173	4.2G	5.1G	4979	168.0G	343.1G	0	0

```

Quit: q, Help: h
DB Size 1.3T/2.7T
db2top 2.0

```

Figure 6-25 db2top Partitions Screen

Outside the scope of a network bottleneck, an uneven usage of FCM resources (except from the administration node which communicates with the rest of the nodes) can be an indication of data skew. This situation occurs unless you have defined custom database partition groups where the workload can be isolated to a few database partitions belonging to the same database partition group.

The relational monitoring function **MON_GET_CONNECTION** gives you the details of the FCM volume per application in order to narrow down the FCM resource usage per application, and detect the applications consuming the most FCM buffers, as shown in Example 6-51.

Example 6-51 FCM buffer usage

```

SELECT APPLICATION_HANDLE AS AGENT_ID,
MEMBER, FCM_RECV_VOLUME AS FCM_RCV_BYTES,
FCM_RECVS_TOTAL AS FCM_RCV_BUFFS,
FCM_SEND_VOLUME AS FCM_SND_BYTES,
FCM_SENDS_TOTAL AS FCM_SND_BUFFS
FROM TABLE(MON_GET_CONNECTION(cast(NULL as bigint), -2))
ORDER BY APPLICATION_HANDLE, MEMBER

```

AGENT_ID	MEMBER	FCM_RCV_BYTES	FCM_RCV_BUFFS	FCM_SND_BYTES	FCM_SND_BUFFS
51	0	8616	31	0	0
51	1	0	0	0	0
51	2	0	0	0	0
51	3	0	0	0	0
51	4	0	0	0	0

51	5	0	0	0	0
51	6	471905	160	19160	20
51	7	0	0	0	0
51	8	471905	160	19160	20
78	0	4503616	1200	3464	8
78	1	0	0	0	0
78	2	0	0	0	0
78	3	0	0	0	0
78	4	0	0	0	0
78	5	0	0	0	0
78	6	0	0	0	0
78	7	0	0	0	0
78	8	0	0	0	0

18 record(s) selected.

In this example, we notice that application handle 51 is driving an uneven FCM resource usage on database partitions 0, 6 and 8.

Alternatively, the **db2pd -fcm** output allows you to further narrow down the FCM traffic per database partition to a given application handle (agent ID). You can then use **db2top** Sessions screen to narrow down to the particular SQL executed by the application handle which might be driving a high FCM consumption. In addition, DB2 9.7 Fix Pack 2 has the **MON_GET_FCM** and **MON_GET_FCM_CONNECTION_LIST** table functions that give FCM monitor metrics.

6.4 Common scenario: Data skew

In this section, we show an example of an uneven resource consumption on a IBM Smart Analytics System caused by data skew.

6.4.1 Operating system monitoring

Through ongoing monitoring, the system administrator notices an unusual pattern of the second data node being used much more than the first data node.

6.4.2 DB2 monitoring

In order to narrow down what is causing an uneven resource consumption, we can use **db2top** to check the resource usage pattern from a DB2 perspective. We start **db2top** as follows:

```
db2top -d bcukit
```

We press **J** to get to the skew detection screen, as shown by Figure 6-26, and we notice that there is indeed a skew in the number of rows read, rows written on database partition 6 and database partition 8. There is also a significant difference in the number of FCM buffer usage.

```
[~] 7:40:41, refresh=2secs(0.144) Skew monitor Linux, part=[9/9], BCULIND
[d=Y, a=N, e=N, p=ALL]
```

Node Nbr	Exec Time	Cpu Time	Rows Second	Rows Read	Rows Written	TqRows Read	TqRows Written	Tq Spills	FCM Sent	FCM Recvd	Subs Exec	Subs Comp	Subs Send
0	83,216	0.308100	0.00	0	0	0	0	0	7,746	7,014	0	0	0
1	85,245	1707.82929	40908.16	3487216107	0	0	0	0	5,733	12,526	10	250	0
2	85,720	1734.99640	41466.52	3554509683	0	0	0	0	5,609	12,306	6	250	0
3	82,220	1634.20817	44315.75	3643641065	0	0	0	0	5,740	12,243	9	250	0
4	82,250	1627.58428	44143.80	3630827144	0	0	0	0	5,656	12,291	11	250	0
5	82,710	1262.56221	35242.69	2914922855	0	0	0	0	5,832	12,681	7	250	0
6	162365	10769.6344	144650.66	23476167667	10,035,345	0	0	0	53,111	14,788	426	0	6
7	82,550	1256.32046	35360.25	2918988972	0	0	0	0	5,709	12,323	6	250	0
8	148080	8845.05162	134523.22	19900126168	20,071,420	0	0	0	45,155	13,907	59	213	1

```
Quit: q, Help: h ISAS56R1D dB
```

Figure 6-26 db2top Skew Monitor screen shot

We can, for example, rely on the FCM traffic usage and track the applications sending most of the traffic on database partitions 6 and 8. Here we use new monitoring function **MON_GET_CONNECTION** which provides the FCM usage per application, as shown in Example 6-52.

Example 6-52 FCM usage

```
SELECT APPLICATION_HANDLE AS AGENT_ID,
MEMBER, ROWS_READ, ROWS_MODIFIED,
FCM_RECV_VOLUME AS FCM_RCV_BYTES,
FCM_SEND_VOLUME AS FCM_SND_BYTES
FROM TABLE(MON_GET_CONNECTION(cast(NULL as bigint), -2))
WHERE MEMBER=6 OR MEMBER=8 ORDER BY FCM_SND_BYTES DESC
```

AGENT_ID	MEMBER	ROWS_READ	ROWS_MODIFIED	FCM_RCV_BYTES	FCM_SND_BYTES
107	8	71924735	92	680626	3034036
84	8	71513627	91	618100	3025924
89	8	71513627	91	617804	3025924
93	8	71565879	90	632696	3025776
127	8	71565067	91	621712	3025480
108	8	71488495	91	617212	3021424
94	8	71488112	90	605933	3021424
96	8	71463759	91	629380	3017664
[...]					
88	8	64594919	78	646046	2738844
105	8	59424897	64	560726	2514733
107	6	55810061	0	692496	2372923
95	8	55703475	64	536982	2359128
126	6	54965316	1	640065	2339884
98	8	55115256	59	477178	2338108
84	6	51653227	6	590509	2189227
108	6	50723946	1	592873	2148668
88	6	50377728	0	593909	2140557
87	6	50420967	0	798972	2139816
118	8	50514698	35	506459	2139811
81	6	50667868	3	575465	2132296
125	6	50318861	6	588225	2120277
127	6	49795137	5	576945	2112018
112	6	49684008	0	649952	2109146
104	6	49417721	4	576797	2103906
119	6	49706626	9	564333	2099702
95	6	49244202	0	550181	2083923
105	6	48634131	0	561017	2071755
122	6	48746276	0	468174	2070423
78	6	48634633	2	601990	2067403
113	6	48601675	12	568537	2067403
116	6	48253805	0	495677	2050587
92	6	48427260	0	468322	2050143
121	6	47559034	1	631969	2022937
[...]					

102 record(s) selected.

We notice that there are a few applications which are the top FCM senders, and receivers, and many have them show around the same FCM application usage. We pick the top two agent IDs and find out what query they are executing using **db2top**.

We press **1** to get to the sessions screen on **db2top**, then press **a**, and enter 107 when prompted for the agent ID, as shown in Figure 6-27.

```
[*]18:42:19, refresh=2secs (0.156) Sessions Linux,part=[9/9], BCULINUX:BCUKIT
[d=Y,a=N,e=N,p=ALL] [qp=off]
Please enter agent id: 107

Local sessions...: 51          Local in exec....: 50
Remote sessions...: 148       Remote in exec....: 144
FCM buffers sent..: 150,002   Active utilities..: 0
FCM buffers rec...: 118,328   FCM buffers LWM..: 131,100
Piped sorts req..: 108       Fiped sorts acc..: 108
Max Agents.....: 0           Agents stolen....: 73     Agent overflows..: 0
Sheaphrns.....: 20.5G      Max Sort Heap....: 608.0K  Sort heap alloc..: 512.0K  Connections HRM..: 72
Dynamic SQL stmts.: 225     Static SQL stmts.: 0       Post hash joins..: 0       Private memory...: 462.8M
Bufferpool reads.: 124,669   Bufferpool writes: 0       Direct reads.....: 0       Post sorts.....: 0
                                                         Direct writes....: 0

Application  Cpu% IO% Mem% Application      Application      Delta      Delta      Delta      Delta
Handle (Stat) Total Total Status      Name            RowsRead/s      RowsWritten/s  IOWrites/s  IOWrites/s
-----
115(*) 34.98% 18.45% 1.49% UOW Executing    db2batch        6,331,290       0             22,997       0
101(*) 4.24% 4.92% 1.60% UOW Executing    db2batch        927,586         0             6,130        0
102(*) 4.16% 4.60% 1.60% UOW Executing    db2batch        846,960         0             5,741        0
79(*) 3.46% 3.85% 1.60% UOW Executing    db2batch        662,022         0             4,802        0
110(*) 3.34% 3.42% 1.69% UOW Executing    db2batch        686,361         0             4,268        0
103(*) 3.13% 4.12% 1.69% UOW Executing    db2batch        711,558         0             5,142        0
95(*) 2.87% 3.05% 1.60% UOW Executing    db2batch        605,243         0             3,802        0
113(*) 2.55% 2.72% 1.69% UOW Executing    db2batch        521,007         0             3,389        0
106(*) 2.53% 2.85% 1.69% UOW Executing    db2batch        492,097         0             3,550        0
81(*) 2.44% 2.58% 1.69% UOW Executing    db2batch        478,913         0             3,222        0
87(*) 2.38% 2.71% 1.77% UOW Executing    db2batch        517,582         0             3,373        0
92(*) 2.21% 2.68% 1.69% UOW Executing    db2batch        463,014         0             3,346        0
96(*) 2.17% 2.28% 1.69% UOW Executing    db2batch        432,662         0             2,843        0
111(*) 2.11% 2.19% 1.60% UOW Executing    db2batch        402,361         0             2,735        0
100(*) 2.03% 2.86% 1.69% UOW Executing    db2batch        465,738         0             3,562        0
93(*) 2.01% 1.99% 1.52% UOW Executing    db2batch        378,070         0             2,484        0
78(*) 2.00% 2.32% 1.77% UOW Executing    db2batch        397,653         0             2,888        0
82(*) 1.92% 2.17% 1.69% UOW Executing    db2batch        392,021         0             2,708        0
120(*) 1.88% 1.96% 1.69% UOW Executing    db2batch        357,833         0             2,444        0
85(*) 1.75% 1.75% 1.69% UOW Executing    db2batch        330,353         0             2,187        0
119(*) 1.75% 1.81% 1.69% UOW Executing    db2batch        345,509         0             2,260        0
123(*) 1.72% 1.77% 1.69% UOW Executing    db2batch        333,120         0             2,209        0
86(*) 1.67% 2.16% 1.69% UOW Executing    db2batch        340,807         0             2,698        0
126(*) 1.65% 1.71% 1.60% UOW Executing    db2batch        313,747         0             2,138        0
118(*) 1.34% 1.73% 1.69% UOW Executing    db2batch        301,867         0             2,153        0
98(*) 1.12% 1.79% 1.69% UOW Executing    db2batch        277,933         0             2,236        0
107(*) 0.88% 1.35% 1.60% UOW Executing    db2batch        157,570         0             1,680        0
114(*) 0.88% 1.40% 1.60% UOW Executing    db2batch        230,642         0             1,742        0
91(*) 0.83% 5.90% 1.32% UOW Executing    db2batch        893,311         0             7,354        0
122(*) 0.82% 1.07% 1.60% UOW Executing    db2batch        204,365         0             1,334        0
116(*) 0.76% 1.23% 1.43% UOW Executing    db2batch        198,776         0             1,536        0

Quit: q, Help: h                               194 active session(s) out of 199                               db2top 2.0
```

Figure 6-27 db2stop Sessions screen

Figure 6-28 shows the Session details screen. We review the SQL being executed by the application. It is a simple SQL statement selecting data only from one table TPCD.PARTSKW.

```
[*]17:18:17, refresh=2secs (0.005) Sessions Linux,part=[9/9], BCULINUX:BCUKIT
[d=Y,a=N,e=N,p=ALL] [qp=off]

*NO.bculinix.101001192140, UOW Executing
lqQuery textxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x
Sub x
Sec (Sx) x SubSection
--- ---- x
0 x
1 3x select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty x1 R1
x, sum(l_ext x3 C5 E3
x, sum(l_ext x
) as sum_base_price, sum(l_ext x
t) as sum_disc endedprice * (1 - l_discount)) as sum_disc_price, sum(l_ext xprice * (1 - l_discount
) as avg_price endedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge, avg(l_quantity) xy, avg(l_extendedprice
y group by l_x as avg_qty, avg(l_ext x('1998-12-01') - 90 da
x endedprice) as avg_price, avg(l_discount) as avg_disc, count_big(*) x
x as count_order x
x from tpcd.linitem x
x where l_shipdate <= date ('1998-12-01') - 90 day x
x group by l_returnflag, l_linestatus x
x order by l_returnflag, l_linestatus x
x
x
x
mqs=db2explng x=db2exfmt qqw=writeqqE=editxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx]

Quit: q, Help: h                               Tot cpu 3606.00046, associated memory 1.2M (total 2.3M), enter to refresh                               db2top 2.0
```

Figure 6-28 Session detail screen

We do the same investigation for the second application with agent ID 84, and it turns out that the application is executing the same query.

At this point, we can further check the table and get a **db2look** output to verify its distribution key. Then we can run a query, as shown in Example 6-53, to verify the distribution of the table.

Example 6-53 Using db2look to obtain DDL

```
# db2look -e -d bcukit -z TPCD -t PARTSKW
-- No userid was specified, db2look tries to use Environment variable USER
-- USER is: BCULINUX
-- Specified SCHEMA is: TPCD
-- The db2look utility will consider only the specified tables
-- Creating DDL for table(s)

-- Schema name is ignored for the Federated Section
-- This CLP file was created using DB2LOOK Version "9.7"
-- Timestamp: Thu 07 Oct 2010 05:30:45 PM EST
-- Database Name: BCUKIT
-- Database Manager Version: DB2/LINUX8664 Version 9.7.2
-- Database Codepage: 1208
-- Database Collating Sequence is: IDENTITY

CONNECT TO BCUKIT;

-----
-- DDL Statements for table "TPCD  "."PARTSKW"
-----

CREATE TABLE "TPCD  "."PARTSKW" (
    "P_PARTKEY" INTEGER NOT NULL ,
    "P_NAME" VARCHAR(55) NOT NULL ,
    "P_MFGR" CHAR(25) NOT NULL ,
    "P_BRAND" CHAR(10) NOT NULL ,
    "P_TYPE" VARCHAR(25) NOT NULL ,
    "P_SIZE" INTEGER NOT NULL ,
    "P_CONTAINER" CHAR(10) NOT NULL ,
    "P_RETAILPRICE" DOUBLE NOT NULL ,
    "P_COMMENT" VARCHAR(23) NOT NULL )
    COMPRESS YES
    DISTRIBUTE BY HASH("P_MFGR")
    IN "OTHERS" ;

-- DDL Statements for indexes on Table "TPCD  "."PARTSKW"

CREATE UNIQUE INDEX "TPCD  "."P_PSKW" ON "TPCD  "."PARTSKW"
    ("P_PARTKEY" ASC,
    "P_MFGR" ASC)
    PCTFREE 0
    COMPRESS NO ALLOW REVERSE SCANS;

COMMIT WORK;

CONNECT RESET;

TERMINATE;
```

We can check the data distribution using an SQL query, as shown in Example 6-54. The query shows that the entire table appears to have 200 million rows (by doing a sum of each node cardinality). We can also see that the table has roughly 60% of rows located on database partition 6 and 40% located on database partition 8. No rows are located on any other database partition. These partitions show the highest FCM buffer usage on Example 6-52 on page 197. Based on this output, we have a very significant data skew.

Example 6-54 Data distribution for TPCD.PARTSKW

```
# db2 "select dbpartitionnum(p_mfgr) as NODE_NUMBER, count(*) AS NODE_CARD from
tpcd.partskw group by dbpartitionnum(p_mfgr) order by dbpartitionnum(p_mfgr)"
```

```
NODE_NUMBER NODE_CARD
-----
          6  120000973
          8   79999027
```

2 record(s) selected.

We can further verify the domain for the column P_MFGR as shown in Example 6-55. We notice that only three unique values are currently being used in this column, which does not make this particular column an ideal distribution key.

Example 6-55 Domain for column P_MFGR

```
# db2 "select distinct p_mfgr, count(*) from tpcd.partskw group by p_mfgr"
```

```
P_MFGR                2
-----
Manufacturer#1         79998128
Manufacturer#3         40002845
Manufacturer#4         79999027
```

3 record(s) selected.

In this particular scenario, we showed that the uneven resource usage is caused by a data skew. The data skew results from a poor distribution key choice.

For further guidelines on distribution keys, see the following documentation:

- ▶ DB2 information Center:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.partition.doc/doc/c0004906.html>

- ▶ IBM developerWorks article, *Choosing partitioning keys in DB2 Database Partitioning Feature environments*:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-1005partitioningkeys/>



Advanced configuration and tuning

In this chapter we discuss details of the configuration of your IBM Smart Analytics System for your particular environment. We provide a configuration parameter summary for the IBM Smart Analytics System 5600 V1/V2, 7600, and 7700. We discuss parameters that can be adjusted to meet your specific workload needs.

We also describe how to configure DB2 workload manager to minimize the performance degradation seen due to a concurrent or conflicting use of resources.

In certain cases, additional tuning or workload management might not be sufficient or appropriate to resolve the performance degradation experienced and to meet your service level agreement (SLA). The answer in these cases might reside in increasing the resources available on the existing physical partitions, such as additional memory or a Solid State Drive (SSD), to provide better performance, or scaling up your system by adding additional partitions. We discuss these options in the last section.

7.1 Configuration parameters

The IBM Smart Analytics System is designed to provide optimal performance for business intelligence workloads and comes with a prescribed configuration. The configuration is tailored based on the hardware specifications, and the entire architecture of the solution. It integrates the best practices in the field and takes into account the customer's typical business intelligence environment workloads and constraints to provide a good starting point for most customer environments.

In this section, we discuss the various types of parameters, along with guidelines on whether these parameters can be changed or not. We review the parameters for the IBM Smart Analytics System 7600 and 7700, as well as the IBM Smart Analytics System 5600 V1 and V2. Certain 7600 configurations use DB2 9.5. However, this chapter focuses only on configurations that use DB2 9.7.

Configuration parameters are also described for specific environments in the official IBM Smart Analytics System documentation, available for download at the following link:

https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=idwbcu

These parameters are specific to the IBM Smart Analytics System, and might not be appropriate for tuning other environments. If there is any discrepancy on your system to what is described in this section, check the latest IBM Smart Analytics System official documentation. If there are discrepancies between the official documentation and your original system settings, contact your IBM Smart Analytics System Support.

7.1.1 Operating system and kernel parameters

In this section, we discuss the various operating system and kernel parameters for the AIX-based and the Linux-based environments.

AIX: 7600 and 7700 based environments

The AIX kernel parameters to use are almost identical for the IBM Smart Analytics System 7600 and 7700. Differences, if any, are mentioned in this section. Details about the meaning of the parameters, the commands to set them, and how to display these parameters, are documented in the AIX 6.1 Information Center at this address:

<http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp>

All the commands must be submitted as *root* unless specified otherwise.

Important: Do not use any deviation for the AIX kernel parameters described in this section without consulting your IBM Smart Analytics System Support.

AIX Virtual Memory Manager

IBM Smart Analytics System 7600 and 7700 environments use the default AIX V6.1 Virtual Memory Manager (VMM) settings. These parameters are accessible through the AIX **vmo** command. You can use the following **vmo** command to display these parameters:

```
vmo -L
```

Example 7-1 shows an output of **vmo -L** to display VMM parameters on an IBM Smart Analytics System 7700 environment.

Example 7-1 Virtual Memory Manager parameters for the 7700

```
(0) root @ isas77adm: 6.1.0.0: /
# vmo -L
```

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
DEPENDENCIES							
ame_cpus_per_pool	n/a	8	8	1	1K	processors	B
ame_maxfree_mem	n/a	24M	24M	320K	16G	bytes	D
ame_minfree_mem							
ame_min_ucpool_size	n/a	0	0	5	95	% memory	D
ame_minfree_mem	n/a	8M	8M	64K	16383M	bytes	D
ame_maxfree_mem							
ams_loan_policy	n/a	1	1	0	2	numeric	D
enhanced_affinity_affin_time	1	1	1	0	100	numeric	D
enhanced_affinity_vmpool_limit	10	10	10	-1	100	numeric	D
force_realias_lite	0	0	0	0	1	boolean	D
kernel_heap_psize	64K	0	0	0	16M	bytes	B
lgpg_regions	0	0	0	0	8E-1		D
lgpg_size							
lgpg_size	0	0	0	0	16M	bytes	D
lgpg_regions							
low_ps_handling	1	1	1	1	2		D
maxfree	1088	1088	1088	16	25548K	4KB pages	D
minfree							
memory_frames							
maxperm	27897K		27897K				S
maxpin	25731K		25731K				S

maxpin%	80	80	80	1	100	% memory	D
pinnable_frames							
memory_frames							
memory_frames	31936K		31936K			4KB pages	S
memplace_data	0	0	0	0	2		D
memplace_mapped_file	0	0	0	0	2		D
memplace_shm_anonymous	0	0	0	0	2		D
memplace_shm_named	0	0	0	0	2		D
memplace_stack	0	0	0	0	2		D
memplace_text	0	0	0	0	2		D
memplace_unmapped_file	0	0	0	0	2		D
minfree	960	960	960	8	25548K	4KB pages	D
maxfree							
memory_frames							
minperm	952232		952232				S
minperm%	3	3	3	1	100	% memory	D
nokilluid	0	0	0	0	4G-1	uid	D
npskill	96K	96K	96K	1	12M-1	4KB pages	D
npswarn	384K	384K	384K	1	12M-1	4KB pages	D
numpsblks	12M		12M			4KB blocks	S
pinnable_frames	30184K		30184K			4KB pages	S
relalias_percentage	0	0	0	0	32K-1		D
scrub	0	0	0	0	1	boolean	D
v_pinshm	0	0	0	0	1	boolean	D
vmm_default_pspa	0	0	0	-1	100	numeric	D
wlm_memlimit_nonpg	1	1	1	0	1	boolean	D

n/a means parameter not supported by the current platform or kernel

Parameter types:

S = Static: cannot be changed
D = Dynamic: can be freely changed
B = Bosboot: can only be changed using bosboot and reboot
R = Reboot: can only be changed during reboot
C = Connect: changes are only effective for future socket connections
M = Mount: changes are only effective for future mountings
I = Incremental: can only be incremented
d = deprecated: deprecated and cannot be changed

Value conventions:

K = Kilo: 2¹⁰ G = Giga: 2³⁰ P = Peta: 2⁵⁰
M = Mega: 2²⁰ T = Tera: 2⁴⁰ E = Exa: 2⁶⁰

I/O tuning parameters

IBM Smart Analytics System 7600 and 7700 environments use the default AIX V6.1 I/O tuning parameters, except for these:

- ▶ `j2_minPageReadAhead=32`
This parameter represents the minimum number of pages read ahead when VMM first detects a sequential reading pattern.
- ▶ `j2_maxPageReadAhead=512`
This parameter represents the maximum number of pages that VMM can read ahead during a sequential access.

These parameters are beneficial for buffered file system I/O. DB2 does not use file system caching for the bulk of its I/O activity, such as table space access as well as active transaction log files. Therefore, these parameters will not have a significant impact on DB2 performance. However, they might benefit any application or utility performing buffered file system I/O on the system.

You can access these parameters through the AIX `ioo` command. The following command can be used to check these parameters:

```
ioo -L
```

Example 7-2 shows how to set these parameters with `ioo`.

Example 7-2 Setting I/O tuning parameters with ioo on a 7700

```
# ioo -p -o j2_minPageReadAhead=32 -o j2_maxPageReadAhead=512
```

Network parameters

The following kernel network parameters are changed from the default value in order to provide an optimal performance on the network:

- ▶ `sb_max=1310720`
This parameter represents the maximum buffer space that can be used by a socket send or receive buffer. This value caps the maximum size that can be set for `udp_sendspace`, `udp_recvspace`, `tcp_sendspace`, and `tcp_recvspace`.
- ▶ `rfc1323=1`
This parameter enables the TCP scaling option. With this parameter set, the maximum TCP window size can grow up to 4 GB.

- ▶ `ipqmaxlen=250`

This parameter sets the maximum internet protocol (IP) input queue length to 250. This value is increased to limit any input queue overflow. You can monitor overflows using the following command:

netstat -p ip

Example 7-3 shows an example of this command on a 7700 system.

Example 7-3 Monitoring IP input queue

```
# netstat -p ip | grep overflows
0 ipintrq overflows
```

- ▶ `udp_sendspace=65536`

This parameter represents the size of the largest User Datagram Protocol (UDP) that can be sent.

- ▶ `udp_recvspace=655360`

This parameter represents the amount of incoming data that can be queued on each UDP socket. `udp_recvspace` is set to 10x `udp_sendspace` to provide buffering according to best practice.

- ▶ `tcp_sendspace=221184`

This parameter specifies how many bytes of data can be buffered in the kernel (using the mbufs kernel memory buffers) by the TCP sending socket before getting blocked. For IBM Smart Analytics System, the value is equal to `tcp_recvspace` parameter.

- ▶ `tcp_recvspace=221184`

This parameter specifies how many bytes of data can be buffered in the kernel (using the mbufs kernel memory buffers) on the receiving socket queue. This value is significant because it is used by the TCP protocol to determine the TCP window size and limit the number of bytes it sends to a receiver.

The following AIX command can show the current settings in your environment:

```
no -L
```

Example 7-4 shows an example of how to set these parameters with `no`.

Example 7-4 Setting network parameters with no

```
no -p -o sb_max=1310720 -o rfc1323=1 -o udp_sendspace=65536 -o
udp_recvspace=655360 -o tcp_sendspace=221184 -o tcp_recvspace=221184
no -r -o ipqmaxlen=250
```

Jumbo frames

The network interfaces for the internal application network, also known as the DB2 FCM network, have jumbo frames enabled. The following command enables jumbo frames:

```
chdev -l ent2 -a jumbo_frames=yes
```

In this command, **ent2** is the network interface corresponding to the internal application network used for the DB2 FCM internal communications between database partitions.

Example 7-5 shows the command to display the settings of a network interface device on an IBM Smart Analytics System 7700.

Example 7-5 Displaying a network interface device settings

```
# lsattr -El ent2
alt_addr      0x000000000000 Alternate ethernet address      True
busintr       74261          Bus interrupt level              False
busmem        0xff9f0000     Bus memory address              False
chksum_offload yes            Enable hardware transmit and receive checksum True
delay_open    no             Delay open until link state is known True
flow_ctrl     yes           N/A                             True
intr_priority 3             Interrupt priority              False
iomem         0xff000        Bus I/O address                 False
jumbo_frames  yes           Enable jumbo frames            True
large_receive yes           Enable receive TCP segment aggregation True
large_send    yes           Enable hardware transmit TCP segmentation True
rom_mem       0xffb00000     ROM memory address             False
rx_coalesce   16            Receive packet coalesce count   True
rx_intr_delay 100           Receive Interrupt Delay timer    True
rx_intr_limit 1000          Max receive buffers processed per interrupt True
rx_ipkt_idelay 4             Inter-packet Interrupt Delay timer True
rxdesc_queue_size 1024         Receive descriptor queue size    True
tx_queue_size 8192          Software transmit queue size     True
txdesc_queue_size 512         Transmit descriptor queue size    True
use_alt_addr  no            Enable alternate ethernet address True
```

Fibre Channel device settings

The following Fiber Channel parameters are set for all Fibre Channel devices on the system. These parameters help in providing optimal performance for large sequential I/O block size, which DB2 is using during sequential prefetching:

► **lg_term_dma=0x1000000**

This parameter controls the direct memory access (DMA) memory in bytes that the Fibre Channel adapter can use.

► **max_xfer_size=0x100000**

This parameter determines the maximum I/O transfer size in bytes that the adapter can support.

- ▶ num_cmd_elems=1024

This parameter determines the maximum number of commands that can be queued to the adapter.

You can use the following command to display the settings for your Fibre Channel devices:

```
lsattr -E1 fcs0
```

In this command, **fcs0** represents the Fibre Channel device, and **fcs0-fc3** are configured in an IBM Smart Analytics System 7600 or 7700 environment.

Example 7-6 shows an example of **lsattr -E1** output for a Fibre Channel device on a 7700.

Example 7-6 Displaying the Fibre Channel adapter settings

```
# lsattr -E1 fcs0
bus_io_addr 0xff800          Bus I/O address          False
bus_mem_addr 0xff9f8000         Bus memory address       False
init_link   auto            INIT Link flags          True
intr_msi_1  254487           Bus interrupt level      False
intr_priority 3             Interrupt priority       False
lg_term_dma 0x1000000         Long term DMA            True
link_speed  auto            Link Speed Setting       True
max_xfer_size 0x100000    Maximum Transfer Size    True
num_cmd_elems 1024       Maximum number of COMMANDS to queue to the adapter True
pref_alpa    0x1            Preferred AL_PA          True
sw_fc_class  3             FC Class for Fabric      True
```

Example 7-7 shows how to set these parameters using the **chdev -l** command.

Example 7-7 Using chdev -l to set Fibre Channel adapter parameters

```
chdev -l fcs0 -a lg_term_dma=0x1000000 -a max_xfer_size=0x100000 -a
num_cmd_elems=1024
```

Hdisk devices settings

The following hdisk device settings are set for the external storage hdisks on the IBM Smart Analytics System 7600 and 7700, all of which use the AIX PCM MPIO (Multiple Path I/O) driver.

- ▶ max_transfer=0x100000

This parameter determines the maximum transfer size in bytes in a single operation. Larger I/O block requests exceeding this size will be broken into smaller block sizes by the MPIO driver.

- ▶ `queue_depth=128`
This parameter represents the maximum number of requests that can be queued to the device.
- ▶ `reserve_policy=no_reserve`
This parameter represents the reservation policy for the device. `no_reserve` setting does not apply a reservation methodology for the device. The device might be accessed by other initiators, which might be on other host systems.
- ▶ `algorithm=round_robin`
This parameter represents the algorithm used by the MPIO driver to distribute I/O across the multiple paths for the device. The `round_robin` setting distributes the I/O across all paths configured.

You can use the following command to display your `hdisk` settings:

```
lsattr -E1 hdisk10
```

In this command, **hdisk10** represents a LUN on the external storage in this example.

Example 7-8 shows how to set these parameters.

Example 7-8 Setting `hdisk` parameters

```
chdev -l hdisk10 -a max_transfer=0x100000 -a queue_depth=128 -a
reserve_policy=no_reserve -a algorithm=round_robin
```

IOCP enablement

DB2 9.7 supports I/O Completion Port (IOCP) for asynchronous I/O by default.

Example 7-9 shows how to check if IOCP is enabled at the AIX level. The output shows IOCP in “Available” state.

Example 7-9 Checking IOCP enablement

```
# lsdev -Cc iocp
iocp0 Available I/O Completion Ports
```

In order to make sure that IOCP works with DB2, you can monitor the `db2diag.log` DB2 diagnostics log file when the instance is starting up, you will get a warning message if IOCP is not enabled.

Example 7-10 shows an example of a `db2diag.log` message when IOCP is not enabled on the system.

Example 7-10 IOCP message in db2diag.log

```
2010-09-17-09.25.05.888484-300 E3313413A406      LEVEL: Warning
PID      : 54919616          TID   : 258          PROC  : db2sysc 1
INSTANCE: bcuaix           NODE  : 001
EDUID   : 258              EDUNAME: db2sysc 1
FUNCTION: DB2 UDB, oper system services, sqloStartAIOWriterEDUs, probe:30
MESSAGE : ADM0513W db2start succeeded. However, no I/O completion port (IOCP)
          is available.
```

Maximum number of processes per user

The AIX **maxuproc** parameter is set to 4096. The following command, issued as root, sets **maxuproc** to 4096:

```
chdev -l sys0 -a maxuproc=4096
```

Example 7-11 shows how to verify the **maxuproc** setting on a 7700.

Example 7-11 Verifying maxuproc setting

```
# lsattr -El sys0 | grep maxuproc
maxuproc      4096          Maximum number of PROCESSES allowed per user
True
```

User limits

The following user limits are set to unlimited for all the users on the system:

- ▶ Core size (core)
- ▶ Data size (data)
- ▶ File size (fsize)
- ▶ Number of open file descriptors (nofiles)
- ▶ Stack size (stack) - with a hard limit of 4 GB

You can update their values in the `/etc/security/limits` file on each node of the cluster. All these values can be set to -1 for default, and `stack_hard` can be set to 4194304.

Example 7-12 shows how to update the `/etc/security/limits` file.

Example 7-12 Setting user limits using /etc/security/limits file

```
default:
    fsize = -1
    core = -1
    cpu = -1
    data = -1
    rss = -1
```

```

stack = -1
stack_hard = 4194304
nofiles = -1

```

Another method consists of using the **chuser** command as root for all users on the system. Example 7-13 shows an example of **chuser** usage for this purpose.

Example 7-13 Set user limits using chuser

```
chuser core=-1 data=-1 fsize=-1 nofiles=-1 stack=-1 stack_hard=4194304 bcuaix
```

Linux: IBM Smart Analytics System 5600 V1 and 5600 V2 environments

In this section we list all the kernel parameters settings for the IBM Smart Analytics System 5600 V1 and 5600 V2 environments with or without SSD options. The kernel parameters settings apply to all environments, unless specified otherwise (for example, kernel IPC parameters). At the end of the section, there is an example of how to update these parameters.

Kernel parameters

The following Linux kernel parameters are set for the IBM Smart Analytics System 5600 V1 and 5600 V2:

- ▶ Kernel IPC parameters:

Table 7-1 lists the IPC related kernel parameters and their settings on the IBM Smart Analytics System 5600 V1 and 5600 V2.

Table 7-1 IPC related kernel parameters

Kernel parameter	Meaning	5600 V1	5600 V2
kernel.msgmni (MSGMNI)	Maximum number of system wide message queue identifiers.	16384	131072
kernel.msgmax (MSGMAX)	Maximum size of a message that can be sent by a process	Default (65536)	65536
kernel.msgmnb (MSGMNB)	Maximum number of a bytes in a message queue	Default	65536
kernel.sem (SEMMSL)	Maximum number of semaphores per array	250	250
kernel.sem (SEMMNS)	Maximum number of semaphores system wide:	256000	256000

Kernel parameter	Meaning	5600 V1	5600 V2
kernel.sem (SEMOPM)	Maximum number of operations in a single semaphore call	32	32
kernel.sem (SEMMNI)	Maximum number of semaphores array	8192	32768
kernel.shmmni (SHMMNI)	Maximum number of shared memory segments	32768	32768
kernel.shmmax (SHMMAX)	Maximum size in bytes of a shared memory segment	Default	128 000 000 000
kernel.shmall (SHMALL)	Maximum amount of shared memory that can be allocated	Default	256 000 000 000

IPC resources: The Linux kernel parameters related to IPC resources are a good starting point for IBM Smart Analytics System 5600 V1 and 5600 V2. If there is suspicion or evidence of DB2 errors due to a shortage of IPC resources, consult with your IBM Smart Analytics System support.

In order to check the value for a parameter, you can read the corresponding parameter from `/proc/sys/kernel`. Example 7-14 shows how to display these values.

Example 7-14 Displaying the value of IPC related kernel parameters

```
# cat /proc/sys/kernel/msgmni
131072
# cat /proc/sys/kernel/sem
250    256000  32    32768
```

You can use the `ipcs -l` command to display the IPC related kernel parameters in effect for your system. Example 7-15 shows an output from an IBM Smart Analytics System 5600 V1 system.

Example 7-15 Displaying IPC related kernel parameters

```
# ipcs -all

----- Shared Memory Limits -----
max number of segments = 16128
max seg size (kbytes) = 18014398509481983
max total shared memory (kbytes) = 4611686018427386880
min seg size (bytes) = 1

----- Semaphore Limits -----
```

```
max number of arrays = 8192
max semaphores per array = 250
max semaphores system wide = 256000
max ops per semop call = 32
semaphore max value = 32767
```

```
----- Messages: Limits -----
max queues system wide = 16384
max size of message (bytes) = 65536
default max size of queue (bytes) = 65536
```

Important: For all other Linux kernel parameters listed next, do not deviate from the configuration listed without consulting the IBM Smart Analytics System support services.

► `kernel.suid_dumpable=1`

This kernel parameter controls if a core file can be dumped from a **suid** program such as DB2. This setting enables DB2 core dump files for problem description and problem source identification (PD/PSI) for IBM Smart Analytics System Support services. You can verify this parameter setting by running the following command:

```
cat /proc/sys/kernel/suid_dumpable
```

► `kernel.randomize_va_space=0`

This kernel parameter disables the Linux address space randomization. You need to disable this feature because it can cause errors with the DB2 backup utility or log archival process. You can verify this parameter setting by running the following command:

```
cat /proc/sys/kernel/randomize_va_space
```

► `vm.swappiness=0`

This parameter determines the kernel preference for using swap space versus RAM. When set to the minimal value 0, it means that swap space usage is not favored at all by the kernel. With this configuration, in general, the kernel will delay swapping until it becomes necessary. You can verify this parameter setting by running the following command:

```
cat /proc/sys/vm/swappiness
```

- ▶ `vm.dirty_background_ratio=5` and `vm.dirty_ratio=10`

The `vm.dirty_background_ratio` parameter represents the percentage of dirty pages resulting from I/O write operations which triggers a background flush of the pages. This parameter works in conjunction with `vm.dirty_ratio`.

The `vm.dirty_ratio` parameter represents the percentage of dirty pages ratio in memory resulting from I/O write operations before they are being forced to flush on the system, causing I/O writes to be blocked till the flush completes.

These settings help in limiting dirty page caching. On the 5600 environments, DB2 does not use file system caching for table spaces and transaction log files, and is not impacted by this setting.

All the kernel settings listed previously can be updated in the `/etc/sysctl.conf` file on each node on the cluster. Example 7-16 shows an example of `/etc/sysctl.conf` from an IBM Smart Analytics System 5600 V1 environment.

Example 7-16 sysctl.conf

```
# Disable response to broadcasts.
# You don't want yourself becoming a Smurf amplifier.
net.ipv4.icmp_echo_ignore_broadcasts = 1
# enable route verification on all interfaces
net.ipv4.conf.all.rp_filter = 1
# enable ipv6 forwarding
#net.ipv6.conf.all.forwarding = 1
kernel.msgmni=16384
kernel.sem=250 256000 32 32768
vm.swappiness=0
vm.dirty_ratio=10
vm.dirty_background_ratio=5
kernel.suid_dumpable=1
kernel.randomize_va_space=0
```

After the `/etc/sysctl.conf` file has been updated, run the following command to load these kernel parameters and make them effective at next reboot:

```
sysctl -p
```

User limits

The IBM Smart Analytics System 5600 V1 and V2 use the default `ulimit` parameters, with the exception of `nfiles`, which is set explicitly to 65536. Example 7-17 shows the default parameters in a 5600 V1 environment.

Example 7-17 User limits of 5600 V1

```
# ulimit -a
core file size          (blocks, -c) 0
data seg size          (kbytes, -d) unlimited
file size              (blocks, -f) unlimited
pending signals        (-i) 540672
max locked memory      (kbytes, -l) 32
max memory size        (kbytes, -m) unlimited
open files             (-n) 65536
pipe size              (512 bytes, -p) 8
POSIX message queues   (bytes, -q) 819200
stack size             (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes     (-u) 540672
virtual memory         (kbytes, -v) unlimited
file locks             (-x) unlimited
```

7.1.2 DB2 configuration

In this section, we provide a summary of the DB2 instance and database configuration for the IBM Smart Analytics System 5600 V1/V2, 7600, and 7700. For detailed information about any parameter in this section, consult the DB2 9.7 Information Center at the following link:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>

Configuration and design: The DB2 configuration and design choices on the IBM Smart Analytics System result from a thorough performance validation and testing, and provide a strong performance for the hardware specifications of the appliance. Also, this configuration integrates the best practices in the field for business intelligence workloads.

Design choices such as the file system layout and the number of logical database partitions per physical partition, must not be modified, because doing so constitutes a major deviation from the prescribed configuration.

Configuration settings closely tied to the hardware specifications of the appliance (DB2_PARALLEL_IO registry variable, or NUM_IOSERVERS, for example) must not be modified without consulting your IBM Smart Analytics System support. Changes to such configuration settings can result in a performance degradation.

Parameters related to best practices in the field, such as optimizer related registry variable settings, provide you strong performance for the analytical query workload. If there is evidence that these parameters are not beneficial for your specific query workload or environment, you can disable them.

DB2 memory related parameters, such as buffer pool sizing or sort heap listed for IBM Smart Analytics System offerings, constitute a good starting point for most business intelligence workloads. However, these parameters can be subject to further tuning and adjustments, depending on your particular workload.

Overall, parameters in the DB2 configuration are a good starting point for most workloads. However, further adjustments and tuning might be required depending on your specific requirements.

It is critical to understand the impact and thoroughly test the effect of any configuration changes that you plan to make on the environment:

- ▶ Only consider making configuration changes to address performance bottlenecks, and only if there is an expected benefit.
- ▶ When tuning DB2, proceed with one change at a time in order to keep track and understand the impact of each change.

Configuration changes considerably affect the DB2 engine behavior, such as turning INTRA_PARALLEL ON, are not desirable. In case of doubt, consult with IBM Smart Analytics System support.

Registry variables

Table 7-2 shows a summary of all registry variables settings for the IBM Smart Analytics System 5600 V1/V2, 7600, and 7700 environments. Next, we discuss the registry variables affecting your performance directly.

Table 7-2 DB2 registry variables

Registry variable	Linux environment		AIX environment	
	5600 V1	5600 V2	7600	7700
DB2_EXTENDED_OPTIMIZATION	ON	ON	ON	ON
DB2_ANTIJOIN	YES	EXTEND	ON (DB2 9.5) EXTEND (DB2 9.7)	EXTEND
DB2COMM	TCPIP	TCPIP	TCPIP	TCPIP
DB2_PARALLEL_IO	*:5	*:5	*:8	*
DB2RSHCMD	/usr/bin/ssh	/usr/bin/ssh	/usr/bin/ssh	/usr/bin/ssh

DB2_EXTENDED_OPTIMIZATION

With the setting enabled, the optimizer uses optimization extensions. It is best practice to enable this setting for analytical query workloads, and is proven to provide strong performance for most business intelligence workloads.

DB2_ANTIJOIN

- ▶ DB2_ANTIJOIN=YES causes the optimizer to search for opportunities to transform subqueries of a NOT IN clause into an antijoin that can be processed more efficiently.
- ▶ DB2_ANTIJOIN=EXTEND causes the optimizer to equally consider subqueries of a NOT EXISTS clause. This parameter can benefit most queries with a NOT IN clause and a NOT EXISTS clause. You can identify all the queries in your environment using these clauses and validate the benefits of this variable for your specific environment.

DB2_PARALLEL_IO

The DB2_PARALLEL_IO setting determines the prefetching parallelism that corresponds to the number of parallel prefetch requests to satisfy your table space prefetch size.

Recent Linux and AIX environments use automatic storage with a single storage path, so all table spaces have a single container. The single container is located on a redundant array of independent disks (RAID) array with multiple disk spindles. This registry variable needs to be enabled to benefit from the prefetching parallelism and leverage all disk spindles available on each LUN.

The following settings determine how sequential prefetching is performed in your environment:

- ▶ **DB2_PARALLEL_IO** setting: *n:d*
 - *n* is the table space ID where the parallelism is to be applied. All IBM Smart Analytics System settings use the wildcard “*” to specify all table spaces.
 - *d* represents the number of disks for the containers for the specific table space. The best practices is to set this value either to the number of active spindles or total spindles, depending on your RAID level and OS.
- ▶ **Table space EXTENT** size: Each individual prefetch request will be the size of an extent.
- ▶ **Table space PREFETCH** size: The prefetch size represents the number of pages requested at a time for a specific table space. Most IBM Smart Analytics System environments use a prefetch size of AUTOMATIC, except for the IBM Smart Analytics System 7700. When prefetch size is set to AUTOMATIC, for this specific environment with a single container per table space, the prefetch size is determined as follows:

Prefetch size (Pages) = DB2_PARALLEL_IO disk setting x extent size (Pages)

For example, based on Table 7-3, we have for 5600 V1:

- PREFETCHSIZE=AUTOMATIC
- DB2_PARALLEL_IO=*:5
- Table space extent size=32

The prefetch size is computed to 32 pages x 5 = 160 pages

The 7700 environment does not use the AUTOMATIC setting for the PREFETCHSIZE, and uses a fixed prefetch size of 384 instead which provides a good performance in that environment.

- ▶ **Database configuration NUM_IOSERVERS**: Most IBM Smart Analytics System environments use the AUTOMATIC setting. NUM_IOSERVERS in this case is determined by the following formula:
$$\text{NUM_IOSERVERS} = \text{MAX}(\text{number of containers in the same stripe set from all your tablespaces}) \times \text{DB2_PARALLEL_IO disk setting}$$

Because IBM Smart Analytics System uses a single container per table space, NUM_IOSERVERS will be equal to the DB2_PARALLEL_IO disk setting. 7700 uses a fixed NUM_IOSERVERS of 12.
- ▶ **Buffer pool** setting: Linux based environments uses vectored I/O to perform sequential prefetching, as this provides good performance in Linux environments. AIX based environments uses block-based I/O with a block area buffer pool. The block size is equal to the extent size.

Table 7-3 shows a summary of the number of parallel prefetch requests, as well as the total prefetch request size, depending on your environment:

Table 7-3 Summary of the number of parallel prefetch request and the total prefetch request size

Environment	RAID array and segment size	EXTENT size (pages)	PREFETCH SIZE setting (pages)	NUM_IOSERVERS	DB2_PARALLEL_IO setting	Number of parallel prefetch requests	Total prefetch size (KB)
5600 V1 and 5600 V2	RAID-6 (4+P+Q) segment 128K	32	AUTO(160)	AUTO(5)	*:5	5	2560
7600	RAID-5 (7+P) segment 256K	16	AUTO(128)	AUTO(8)	*:8	8	2048
7700	RAID-6 (10+P+Q), segment 256K	16	384	12	*	24	6144

The extent size is a multiple of the RAID segment size for all IBM Smart Analytics System offerings to get I/O alignment, and optimize table space access. In IBM Smart Analytics System 5600 environments, the RAID segment size is set to 128K, with an extent size set to 512K. Each extent is striped onto four disks, which is corresponding to the number of active disks in the array. On the IBM Smart Analytics System 7600 and 7700 environments, each extent is equal to the segment size of 256K. So, each extent is on one disk.

For all environments except the IBM Smart Analytics System 7700, the prefetch size is equal to the EXTENT size times the DB2_PARALLEL_IO disk setting. DB2 will satisfy the prefetching request by running a number of prefetch requests in parallel equal to the DB2_PARALLEL_IO disk setting. Each of these prefetch requests is of one extent, assigned each to a DB2 prefetcher. For example on the IBM Smart Analytics System 7600, DB2 will assign eight prefetch requests, of one extent each, to each prefetcher.

For the IBM Smart Analytics System 7700 environment, we have a fixed number of prefetchers 12 (matching the number of spindles in the RAID array). The prefetch size is also fixed at 384 pages. Because the prefetch size is not AUTOMATIC, DB2 computes the prefetching parallelism degree by dividing the PREFETCH size by the extent size, which is equal to 24. DB2 satisfies the prefetch request by assigning in parallel 24 requests of one extent each to 12 prefetchers. This results in assigning two extents request per prefetcher. This setting helps in achieving a more aggressive prefetching, which leverages the higher I/O bandwidth available with the 7700.

Based on performance testing, these settings provide a good I/O sequential throughput from the storage and are tied to the specific storage configuration and specifications. Do not change this parameter value.

Database manager configuration

All the database manager configuration settings are the default ones, except for the settings listed in Table 7-4.

Table 7-4 Database manager configuration parameters

DBM configuration	5600 V1	5600 V2	7600 value	7700 value
CPUSPEED	2.36E-07	2.36E-07	2.70E-07	2.70E-07
COMM_BANDWIDTH	100	100	100	100
NUMDB	1	1	2	1
DIAGPATH	/db2fs/bcuaix/ db2dump	/db2fs/bcuaix/ db2dump	/db2path/bcuaix /db2dump	/db2fs/bcuaix /db2dump
SHEAPTHRES	600000, 1200000 with SSD option	600000, 1400000 with SSD option	600000	1400000
FCM_NUM_BUFFERS	AUTOMATIC(131072)	AUTOMATIC(131072)	AUTOMATIC (8192)	AUTOMATIC (16384)

CPUSPEED and COMM_BANDWIDTH

The database manager configuration parameters CPUSPEED, and COMM_BANDWIDTH are used by the DB2 Optimizer to compute the most optimal access plan. CPUSPEED helps the optimizer in estimating the CPU cost associated for low level operations taken during the query execution. COMM_BANDWIDTH helps the optimizer in evaluating the cost of performing internal communications between database partitions operations during query processing.

The current settings shown for the IBM Smart Analytics System are a good baseline to reflect the system specifications to the DB2 Optimizer. Any change of these values requires a thorough testing of your entire query workload, as it might impact access plans.

SHEAPTHRES

IBM Smart Analytics System does not use shared sorts. Private sorts provide an optimal performance for sort intensive queries in this specific environment. SHEAPTHRES is set to cap the sum of private sort memory allocated in SORTHEAP concurrently for all agents connected to a logical database partition to perform private sorts. SORTHEAP is used for sorting, as well as hash joins processing.

SHEAPTHRES value represents a cap per database logical partition. It is currently set to around 28% to 33% of the total RAM available per logical partition for the IBM Smart Analytics System 5600, and 7600/7700, which is relatively conservative.

This parameter can be further tuned depending on the nature of your query workload (for example, sorts and hash joins), and its concurrency. This parameter can be tuned in conjunction with SORTHEAP.

You can monitor the occurrences of post-threshold sorts using either the database snapshot monitoring, the **db2top** utility, or the DB2 New monitoring facility. Example 7-18 shows an example of a new monitoring query showing a post-threshold sort where SHEAPTHRES has been exceeded. This query shows occurrences of post threshold sorts for each application aggregated for all the partitions.

Example 7-18 Query shows post threshold sorts

```
SELECT application_handle AS app_handle,
SUM(total_sorts) AS sum_sorts,
SUM(sort_overflows) AS SUM_OVERFLOWS,
SUM(post_threshold_sorts) AS sum_post_tresh_sort FROM
TABLE(mon_get_connection(CAST(NULL AS bigint),-2))
GROUP BY application_handle ORDER BY application_handle
```

APP_HANDLE	SUM_SORTS	SUM_OVERFLOWS	SUM_POST_TRESH_SORT
76	0	0	3
77	0	0	0
78	0	0	2
79	0	0	3
80	0	0	0
81	0	0	0
82	0	0	0
83	0	0	0
84	0	0	8
85	0	0	0
86	0	0	8
87	0	0	0
94	0	0	0

13 record(s) selected.

Example 7-19 shows occurrences of post threshold sorts and hash joins from the database manager global snapshot.

Example 7-19 Using snapshot for the occurrences of post threshold sorts and hash joins

```
# db2 get snapshot for dbm global | egrep -i "hash|sort" | grep -v "Sorting"
Private Sort heap allocated           = 43200
Private Sort heap high water mark    = 901680
Post threshold sorts                  = 27
Piped sorts requested                 = 85
Piped sorts accepted                  = 85
Hash joins after heap threshold exceeded = 48
```

FCM_NUM_BUFFERS

The FCM_NUM_BUFFERS configuration parameter has been configured with an initial value that is a good starting point for most environment. For both Linux-based and AIX-based environments, DB2 preallocates additional shared memory to accommodate higher requirements and increase the resources dynamically during runtime, in a transparent manner for applications. However, if the requirement exceeds the additional memory reserved by DB2, you might still be exposed to running out of fast communications manager (FCM) resources.

From a best practice prospective, you can adjust this value closer to your peak requirement and limit the automatic adjustments which can have a small impact on your overall system performance. You can monitor the FCM resources through the **db2pd** utility or the database manager snapshot and verify if the initial values are increased by DB2, or their current level of utilization.

Example 7-20 shows an example of a **db2pd -fcm** command that was run on all partitions with a summary of the FCM usage per physical host. Note that the FCM resources are shared between logical partitions on a given physical partition. So, the usage summary information can just be collected from one logical partition per physical.

Example 7-20 db2pd -fcm

```
rah 'db2pd -fcm | egrep
"Usage|==|Partition|Buffers:|Channels:|Sessions:|LWM:"|grep -v Status'
```

Database Partition 0 -- Active -- Up 0 days 00:13:39 -- Date 09/24/2010 19:35:58
FCM Usage Statistics
=====

Total Buffers:	131565
Free Buffers:	131542
Buffers LWM:	131287
Max Buffers:	1573410
Total Channels:	2685
Free Channels:	2644
Channels LWM:	2543
Max Channels:	1573410
Total Sessions:	895
Free Sessions:	860

```

Sessions LWM: 850
ISAS56R1D1: db2pd -fcm | egrep ... completed ok

Database Partition 1 -- Active -- Up 0 days 00:13:41 -- Date 09/24/2010 19:36:00
FCM Usage Statistics
=====
Total Buffers: 526260
Free Buffers: 526225
Buffers LWM: 526144
Max Buffers: 2097880
Total Channels: 10740
Free Channels: 10687
Channels LWM: 10611
Max Channels: 2097880
Total Sessions: 3580
Free Sessions: 3463
Sessions LWM: 3425
ISAS56R1D2: db2pd -fcm | egrep ... completed ok

Database Partition 5 -- Active -- Up 0 days 00:13:40 -- Date 09/24/2010 19:36:01
FCM Usage Statistics
=====
Total Buffers: 526260
Free Buffers: 526227
Buffers LWM: 526124
Max Buffers: 2097880
Total Channels: 10740
Free Channels: 10688
Channels LWM: 10612
Max Channels: 2097880
Total Sessions: 3580
Free Sessions: 3464
Sessions LWM: 3427
ISAS56R1D3: db2pd -fcm | egrep ... completed ok

```

In the output shown in Example 7-20 on page 224, we can see that the total number of buffers (526260) is below the FCM_NUM_BUFFERS initial configuration, which is 524288 logical database partitions (131072 * 4 = 524288). So, no automatic adjustment has occurred. The low watermark (LWM) for the buffers and the channels are also very close to the total, so we have not been any closer to running out of FCM resources. Because this parameter is set to AUTOMATIC, DB2 can increase or decrease this value depending on the workload requirements on the system.

This information is also available through the MON_GET_FCM relational monitoring function available starting with DB2 9.7 Fix Pack 2.

Example 7-21 shows an example of MON_GET_FCM usage that allows you to monitor the low watermark (bottom) for buffers and channels for each database partition (member).

Example 7-21 Using MON_GET_FCM function

```
# db2 "select substr(hostname,1,10) as hostname, member, buff_total,
buff_free_bottom, ch_total, ch_free_bottom from table(mon_get_fcm(-2)) order
by member"
```

HOSTNAME	MEMBER	BUFF_TOTAL	BUFF_FREE_BOTTOM	CH_TOTAL	CH_FREE_BOTTOM
ISAS56R1D1	0	131565	131303	2685	2648
ISAS56R1D2	1	526260	526193	10740	10631
ISAS56R1D2	2	526260	526193	10740	10631
ISAS56R1D2	3	526260	526193	10740	10631
ISAS56R1D2	4	526260	526193	10740	10631
ISAS56R1D3	5	526260	526187	10740	10632
ISAS56R1D3	6	526260	526187	10740	10632
ISAS56R1D3	7	526260	526187	10740	10632
ISAS56R1D3	8	526260	526187	10740	10632

9 record(s) selected.

On a well balanced environment with no data skew, particular queries with an inefficient access plan can also cause a sudden increase of FCM resource usage. To obtain detailed information about FCM resources usage per application, and identify the application consuming a high amount of FCM resources, you can use the **db2pd -fcm** full output, or the **MON_GET_CONNECTION** output.

Example 7-22 shows an output of **db2pd -fcm**. The FCM buffers and channels resource usage is low and well balanced between applications, with no application having an outstanding usage.

Example 7-22 db2pd -fmc output

db2pd -fcm

Database Partition 1 -- Active -- Up 0 days 00:02:38 -- Date 09/24/2010 20:21:21

FCM Usage Statistics

```
=====
Total Buffers: 526260
Free Buffers: 526212
Buffers LWM: 526145
Max Buffers: 2097880

Total Channels: 10740
Free Channels: 10655
Channels LWM: 10655
Max Channels: 2097880

Total Sessions: 3580
Free Sessions: 3451
Sessions LWM: 3451
```


Partition	BuFs Sent	BuFs Recv	Status
0	415734	99	Active
1	115	115	Active
2	1	1	Active
3	1	1	Active
4	1	1	Active
5	1	1	Active
6	1	1	Active
7	1	1	Active
8	1	1	Active

Buffers Current Consumption

```

-----
AppHandl [nod-index] TimeStamp Buffers In-use
0         [000-00000] 0          16
76        [000-00076] 3392975566 6
80        [000-00080] 3392975567 5
75        [000-00075] 3392975564 5
78        [000-00078] 3392975567 4
81        [000-00081] 3392975568 4
79        [000-00079] 3392975567 4
77        [000-00077] 3392975566 4

```

Channels Current Consumption

```

-----
AppHandl [nod-index] TimeStamp Channels In-use
0         [000-00000] 0          16
77        [000-00077] 3392975566 8
76        [000-00076] 3392975566 8
75        [000-00075] 3392975564 8
78        [000-00078] 3392975567 8
79        [000-00079] 3392975567 8
80        [000-00080] 3392975567 8
81        [000-00081] 3392975568 8
65601586 [1001-00050] 0          4
65546    [001-00010] 0          2
131082   [002-00010] 0          2
262154   [004-00010] 0          2
196618   [003-00010] 0          2
65587    [001-00051] 3392975446 1

```

Buffers Consumption HWM

```

-----
AppHandl [nod-index] TimeStamp Buffers Used

```

Channels Consumption HWM

```

-----
AppHandl [nod-index] TimeStamp Channels Used

```

Database configuration settings

In this section, we present the database configuration settings for the IBM Smart Analytics System 5600 V1/V2, and 7600/7700 environments. We then discuss further the parameters that have an impact on performance.

Table 7-5 contains the DB2 database configuration parameter settings set by default on these environments.

Table 7-5 DB2 database configuration parameters

Configuration parameter	5600 V1	5600 V2	7600	7700
LOCKLIST	16384	16384	16384	16384
MAXLOCKS	10	10	10	10
PCKCACHESZ	-1	-1	-1	-1
SORTHEAP	12000	12000, 35000 with SSD option	20000	35000
LOGBUFSZ	2048	2048	2048	2048
UTIL_HEAP_SZ	65536	65536	65536	65536
STMTHEAP	10000	10000	10000	10000
LOGFILSIZ	12800	12800	12800	12800
LOGPRIMARY	50	50	50	50
LOGSECOND	0	0	0	0
NEWLOGPATH	/db2fs/ bcu linux	/db2plog/ bcu linux	/db2path/ bcuaix	/db2plog/ bcuaix
MIRRORLOGPATH	Not available	/db2mlog/ bcu linux	Not available	/db2mlog/ bcuaix
CHNGPGS_THRESH	Default (60)	Default (60), 30 with SSD option	Default (60)	30
WLM_COLLECT_INIT	Not set	Not set	20	20
DFT_PREFETCH_SZ	AUTO	AUTO	AUTO	384
NUM_IO_SERVERS	AUTO(5)	AUTO(5)	AUTO(8)	12

Configuration parameter	5600 V1	5600 V2	7600	7700
NUM_IO_CLEANERS	AUTO(7)	AUTO(3) Explicitly set to 3 on the administration node	AUTO(1)	AUTO(7) Explicitly set to 7 on the administration node

LOCKLIST and MAXLOCKS

LOCKLIST represents the amount of memory in the database shared memory set that is used to store the locks for all applications connected to the database. It is currently set to 16384.

A high value for LOCKLIST can result in performance degradation associated with the traversal of the lock list by each application each time they request a lock. A value too low might result in premature lock escalations which can hurt the concurrency of the applications in the system. The *IBM Smart Analytics System User Guide* corresponding to your configuration contains detailed information about the sizing of the LOCKLIST. The LOCKLIST value set initially provides a good starting point.

You can use the following **db2pd** command to dump the contents of your locklist for a particular database partition:

```
db2pd -db bcukit -locks
```

MAXLOCKS is the percentage of the locklist contents that can be held by a single application before a lock escalation occurs, which consists in converting multiple row level locks on the same table into a single table level lock. This setting will result in saving space in the locklist.

Another important locking parameter that is set by default is LOCKTIMEOUT which is set to -1. This setting means that applications in lock-wait status will be waiting indefinitely for a lock, instead of timing out. This setting might not be appropriate for all environments and might need to be adjusted based on your specific applications behavior.

If your applications are experiencing locking issues (deadlocks or lock timeouts), it is necessary to identify:

- ▶ The various applications involved
- ▶ The SQL statements from the conflicting applications
- ▶ The database objects on which locking issues are occurring
- ▶ The nature and duration of the locks

This information allows you to understand the scenario involving the applications and utilities in conflict. Database snapshot or relational monitoring function SNAP_GET_DB provides a high level overview of deadlocks, and lock timeouts occurring in your database.

Example 7-23 shows an example of database snapshot excerpt with database level lock information.

Example 7-23 Snapshot of lock information

```
# db2_a11 'db2 "get snapshot for database on bcukit" | grep -i lock | egrep -vi
" MDC|Pool"'

Locks held currently                = 21
Lock waits                          = 5
Time database waited on locks (ms) = 953
Lock list memory in use (Bytes)     = 31872
Deadlocks detected                  = 0
Lock escalations                    = 0
Exclusive lock escalations          = 0
Agents currently waiting on locks   = 0
Lock Timeouts                       = 0
Internal rollbacks due to deadlock  = 0
ISAS56R1D1: db2 "get snapshot ... completed ok

Locks held currently                = 222
Lock waits                          = 1
Time database waited on locks (ms) = 5
Lock list memory in use (Bytes)     = 70656
Deadlocks detected                  = 0
Lock escalations                    = 0
Exclusive lock escalations          = 0
Agents currently waiting on locks   = 0
Lock Timeouts                       = 3
Internal rollbacks due to deadlock  = 0
ISAS56R1D2: db2 "get snapshot ... completed ok
...
```

Example 7-24 shows an example of the SNAP_GET_DB query to monitor locks usage. Note that the information returned is aggregated for all partitions.

Example 7-24 Monitor locks using SNAP_GET_DB

```
db2 "select APPLS_CUR_CONS, LOCKS_HELD, LOCK_WAITS, DEADLOCKS, LOCK_ESCALS,
LOCK_TIMEOUTS from TABLE(SNAP_GET_DB('BCUKIT',-2))"
```

A few monitoring tools are available with DB2 to further drill down the scenario of the lock escalation, or deadlocks. Consult the DB2 9.7 Information Center for more details about each of these options:

- ▶ **db2pd** utility: The DB2 9.7 Information Center contains detailed information about how to diagnose locking issues with **db2pd**:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.trb.doc/doc/c0054595.html>

- ▶ Lock and application snapshots.
- ▶ Relational monitoring functions: Provide aggregated information for all database partitions:
 - MON_GET_APPL_LOCKWAIT: Collects all locks information about what application is waiting for.
 - MON_GET_LOCKS: List all locks currently acquired for the applications connected to the database. The advantage of using this method is that you can specify search arguments to select locks for a particular table.
 - MON_FORMAT_LOCK_NAME: Can be used to format the binary name of a lock into a human readable format.
 - MON_GET_CONNECTION: Can give you locking information per application handle.
 - MON_LOCKWAITS: Useful administrative view which lists all applications on lockwait mode, along with lock details, and the application holding the lock.

For locking issues, it is essential to capture either information at the exact time the lock waits occur or historical information about the lock waits. It might be difficult to reproduce a scenario at will and collect diagnostics. The following ways can be used to get historical information or trigger diagnostic data collection when the problem is occurring:

- ▶ Event monitor:

DB2 9.7 provides a new event monitor for locking:

```
CREATE EVENT MONITOR ... FOR LOCKING
```

This new event monitor contains information about all locking related events including deadlocks and lock timeouts. Previously, the only event monitor available for locking was limited to monitor deadlocks occurrences. This event monitor provides more exhaustive information for both lock timeouts and deadlocks.

- ▶ DB2 callout script:

Another advanced option to understand the exact scenario leading to a deadlock or lock time is to enable a DB2 callout script (**db2pdcfg** setting to catch the error and trigger the diagnostic data collection and **db2cos** for the actual collection) to collect a customizable set of diagnostics at the exact time the deadlock or lock timeout occurs. This option is generally used by IBM Smart Analytics System support services to further narrow down complex or unclear locking scenarios.

PCKCACHESZ

IBM Smart Analytics System sets the package cache to -1, which corresponds to (MAXAPPLS*8). MAXAPPLS is set by default to AUTOMATIC, which can be adjusted by DB2 to allow more connections.

Example 7-25 shows how to check for your actual package cache value in effect for all your nodes.

Example 7-25 Check PCKCACHESZ value

```
db2_a11 'db2pd -db bcukit -dbcfg | egrep -i "value|pckcachesz"|grep -vi freq'
```

```
Description                Memory Value      Disk Value
PCKCACHESZ (4KB)           320              320
ISAS56R1D1: db2pd -db bcukit -dbcfg ... completed ok
```

```
Description                Memory Value      Disk Value
PCKCACHESZ (4KB)           320              320
ISAS56R1D2: db2pd -db bcukit -dbcfg ... completed ok
...
```

In order to check to see if this value is sufficient for your environment, you need to check for package cache overflows. This information can be found either with **db2pd** with **-dynamic** or **-static** flag, or a database snapshot. **SNAP_GET_DB** can provide you a high level overview for any package cache overflows.

Example 7-26 shows a usage example of **SNAP_GET_DB** on how to capture this information along with the output.

Example 7-26 Check package cache information

```
db2 "select PKG_CACHE_LOOKUPS, PKG_CACHE_INSERTS, PKG_CACHE_NUM_OVERFLOWS,
PKG_CACHE_SIZE_TOP from TABLE(SNAP_GET_DB('BCUKIT',-2))"
```

```
PKG_CACHE_LOOKUPS  PKG_CACHE_INSERTS  PKG_CACHE_NUM_OVERFLOWS  PKG_CACHE_SIZE_TOP
-----
                23                15                0                488697
```

```
1 record(s) selected.
```

In case of overflows or if the package cache high watermark size (PKG_CACHE_SIZE_TOP) is close to the PCKCACHESZ (after converting in bytes), you can increase your package cache.

You can also monitor the number of package cache inserts to identify an unusual pattern in package cache usage. In order to narrow down the application performing most of the package cache inserts, the relational monitoring function **MON_GET_CONNECTION** can provide you this information per application.

The value set by default with IBM Smart Analytics System is fairly conservative. You might need to increase it depending on your workload.

SORTHEAP

The IBM Smart Analytics System uses private sorts. The ratio of SHEAPTHRES divided by SORTHEAP determines the number of concurrent sorts supported before hitting the cap set by SHEAPTHRES. When the sum of the private sort allocations on each partition is close to SHEAPTHRES, DB2 starts limiting the amount of sorts allocations by allowing smaller allocations to the various applications to remain within the cap.

Table 7-6 shows a summary of the default sort concurrency level for particular IBM Smart Analytics System environments. This concurrency level represents the theoretical concurrent sort requests which can be ran before sort requests start being capped by the DB2 Database Manager. As explained next, these values can be adjusted to meet your specific needs.

Table 7-6 Summary of default concurrency level

Environment	SHEAPTHRES	SORTHEAP	Sort concurrency level
5600 V1/V2	600000	12000	50
5600 V1 with SSD	1200000	12000	100
5600 V2 with SSD	1400000	35000	40
7600	600000	20000	30
7700	1400000	35000	40

The IBM Smart Analytics System provides initial values for SHEAPTHRES and SORTHEAP which are a good starting point for most analytical query workloads, which are sort and hash join intensive. However, you can adjust these settings depending on your specific environment:

- ▶ If there are occurrences of post threshold sorts and hash joins (see “SHEAPTHRES” on page 222 for further details), you can try to decrease SORTHEAP to allow for more concurrency.
- ▶ You can monitor occurrences of sort and hash joins overflows. Overflows can happen with large amounts of data processed in IBM Smart Analytics System environments and might not necessarily be a problem. If you see:
 - An increase on the number of overflows.
 - The ratio of sort overflows on total number of sorts is increasing or is high.
 - The ratio of hash joins overflows on total number of hash joins is increasing or is high.

You can consider increasing SORTHEAP. However, you might see an increase of post threshold sorts or hash joins. Depending on the memory available on the system, you can still increase SHEAPTHRES proportionally to maintain the same level of concurrency in that case.

Example 7-27 shows how to get a global aggregated view of how many sort overflows and hash join overflows are occurring cluster wide. You can use the method shown in Example 7-18 on page 223 to further narrow down the applications performing the sorts.

Example 7-27 Aggregate view of sort and hash join overflows

```
SELECT sort_heap_allocated, total_sorts, total_sort_time, sort_overflows,
       hash_join_overflows, active_sorts
FROM TABLE(SNAP_GET_DB('BCUKIT',-2))
```

SORT_HEAP_ALLOCATED	TOTAL_SORTS	TOTAL_SORT_TIME	...
496865	59	2380...	
SORT_OVERFLOWS	HASH_JOIN_OVERFLOWS	ACTIVE_SORTS	
1	107	20	

1 record(s) selected.

In order to narrow down the statements performing the sorts or the hash joins, use the following methods:

- ▶ Application snapshots provide a drill down of sort and hash joins activity per application. The snapshots also provide information about the SQL being executed.
- ▶ MON_GET_CONNECTION can be used to obtain application level sort activity.
- ▶ The MON_GET_PKG_CACHE_STMT relational monitoring function can also be used to obtain statement level detailed metrics on sort processing.

Example 7-28 shows an example of MON_GET_PKG_CACHE_STMT usage to display sort summary information.

Example 7-28 Display sort summary information using MON_GET_PKG_CACHE_STMT

```
SELECT VARCHAR(SUBSTR(STMT_TEXT,1,50)) AS STMT,
       MEMBER, TOTAL_SORTS, SORT_OVERFLOWS, POST_THRESHOLD_SORTS
FROM TABLE(MON_GET_PKG_CACHE_STMT('D',NULL,NULL,-2))
WHERE TOTAL_SORTS > 0 ORDER BY STMT, MEMBER;
```

STMT	MEMBER	TOTAL_SORTS	...
------	--------	-------------	-----


```

-----
SELECT VARCHAR(SUBSTR(STMT_TEXT,1,100)) AS STMT,ME      0      2 ...
select  c_custkey, c_name, sum(l_extendedprice *      3      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     1      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     2      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     3      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     4      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     5      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     6      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     7      2 ...
select  l_orderkey, sum(l_extendedprice * (1 - l_     8      2 ...
select  l_returnflag, l_linestatus, sum(l_quanti      1      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      2      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      3      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      4      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      5      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      6      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      7      1 ...
select  l_returnflag, l_linestatus, sum(l_quanti      8      1 ...
...
...SORT_OVERFLOW      POST_THRESHOLD_SORTS
-----
...      0      0
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      1      2
...      0      0
...      0      0
...      0      0
...      0      0
...      0      0
...      0      0
...      0      1
...      0      0
...      0      0

```

LOGBUFSZ

LOGBUFSZ represents the size of the internal buffer used by DB2 logger to store transaction log records. The default DB2 value of 256 is quite small for IBM Smart Analytics System environments, and has been increased to a higher value of 2048. A higher value is necessary to ensure good performance during LOAD of multidimensional clustering (MDC) tables, as additional logging is performance for the MDC block maintenance. This value is sufficient for most workloads.

UTIL_HEAP_SZ

The utility heap is used by DB2 utilities such as BACKUP, RESTORE, or LOAD for allocating buffers. These utilities by default tune their heap usage themselves and adjust their individual heap usage based on the amount of memory available in UTIL_HEAP_SZ.

Allocate sufficient space to the UTIL_HEAP_SZ so that these utilities perform well. The value has been increased to 65536 from the default value and is sufficient for most workloads. See the *IBM Smart Analytics System User's Guide* for your respective version, for a detailed discussion about UTIL_HEAP_SZ sizing.

CHNGPGS_THRESH

CHNGPGS_THRESH represents the percentage of dirty pages in the buffer pool, at which page cleaners are triggered to flush these pages. In order to limit the overhead associated with handling large dirty list pages, the value has been reduced for the 7700 because it has a large buffer pool. Lowering the CHNGPGS_THRESH helps in triggering page cleaners earlier, and more proactively. This approach helps while running write intensive workloads and specific utilities or operations such as REORG, or CREATE INDEX.

DFT_PREFETCH_SZ and NUM_IOSERVERS

These parameters are related to prefetching. See “DB2_PARALLEL_IO” on page 219 for further discussion about these parameters.

7.1.3 DB2 buffer pool and table spaces

In the previous section, we discussed parameter configurations that are preset on the IBM Smart Analytics System. In this section, we discuss database objects such as the DB2 buffer pool and table spaces created on the IBM Smart Analytics System.

DB2 buffer pool

The default page size value chosen for all IBM Smart Analytics System is 16K. All IBM Smart Analytics System offerings have two buffer pools:

- ▶ Default IBMDEFAULTBP buffer pool with a size of 1000 pages for the catalog tables.
- ▶ BP16K: A large unified buffer pool for permanent and temporary table spaces.

Table 7-7 shows the buffer pool sizes and block area sizes for the various IBM Smart Analytics offerings.

Table 7-7 Buffer pool and block area sizes

BP16K buffer pool	5600 V1 (with SSD option)	5600 V2 (with SSD option)	7600	7700
Size (16k pages)	179200 (358400)	179200 (300000)	160000	300000
Size (GB)	2.73 (5.47)	2.73 (4.58)	2.44	4.58
Block area size (16k pages)	N/A	N/A	16000	100000
Block area size (GB)	N/A	N/A	0.24	1.53

One main difference between Linux-based and AIX-based offerings is the use of block areas. For analytical workloads, performance testing has shown that block I/O provides strong performance on the AIX platform, so the buffer pool has a dedicated block area for this purpose. Vectored I/O used by default for prefetching provides strong performance on the Linux platform.

The IBM Smart Analytics System 7700 buffer pool is larger than the buffer pool of IBM Smart Analytics System 7600, and needs a larger block area given a more aggressive prefetching, and higher I/O bandwidth available.

The IBM Smart Analytics System family uses an approach with a large unified buffer pool as a starting point. Managing a single buffer pool provides good performance in most cases. Results can vary depending on your actual workload. You might also have particular requirements in terms of page sizes (for example, need of 32K pages for large rows). You might then need to reduce the BP16K buffer pool and create additional buffer pools accordingly.

Buffer pool snapshot can be used for detailed metrics about buffer pool activity. A key metric to monitor the buffer pools is the buffer pool hit ratio. You can use the MON_GET_BUFFERPOOL table function to obtain these metrics buffer pool hit ratio data for all the nodes in the cluster. Buffer pool metrics are discussed in “DB2 I/O metrics” on page 174.

DB2 table spaces

In this section, we discuss aspects of the DB2 table space design for the IBM Smart Analytics System, as well as guidelines to create table spaces.

Regular table spaces

Recent IBM Smart Analytics System use automatic storage by default. The storage path is pointing on each platform to the file system and LUNs designed for table space data.

Table spaces not using automatic storage also need to have containers placed under the file systems designed for table space data. Create a single container per table space. For platforms where NUM_IOSERVERS is set to automatic, NUM_IOSERVERS is determined as:

MAX(number of containers in the same stripe set from all your tablespaces) x DB2_PARALLEL_IO disk setting

Note that, based on the previous formula, if you add any container to an existing table space or create a table space with multiple containers, these additional containers will be added to the same stripe set by default and will result in increasing the number of prefetchers. For example, in a 7600 configuration, which has two containers per table space, you might see 16 DB2 prefetchers per database partition on a 7600. The number of prefetchers will impact the prefetching performance on your system.

All table spaces are created as database managed space (DMS) table spaces with the default NO FILE SYSTEM CACHING, enabling the use of direct I/O (DIO) and concurrent I/O (CIO). DIO allows to bypass file system caching, and copies the data directly from the disk to the buffer pool. DIO provides strong performance for DMS table spaces. It eliminates the overhead of looking up the data in the file system cache. It also eliminates the cost of copying the data twice, the first time from the disk to the file system cache, and the second time from the file system cache to the buffer pool.

Concurrent I/O optimizes concurrent access to DMS container files. By default, JFS2 uses file level exclusive i-node locking mechanism to serialize concurrent write access, which impacts the performance of multiple DB2 threads trying to read and write data concurrently to the same single DMS container file. Concurrent I/O does not perform exclusive I-node locking systematically for all writes, but only on specific cases, allowing a greater level of concurrent access. Note that the use of DIO is implicit when using CIO.

Table 7-8 contains a summary of the table spaces parameters for various IBM Smart Analytics System platforms.

Table 7-8 Table space parameters

Table space parameters	5600 V1 and 5600 V2	7600	7700
EXTENT SIZE	32	16	16
PREFETCH SIZE	AUTO(160)	AUTO(128)	384
OVERHEAD	3.63	4.0	4.0
TRANSFERRATE	0.07	0.4	0.04

The EXTENT SIZE and PREFETCH SIZE parameters are discussed in “DB2_PARALLEL_IO” on page 219.

DB2 Optimizer uses the OVERHEAD and TRANSFERRATE parameters to estimate the I/O cost during query compilation. The DB2 9.7 Information Center contains information about these parameters:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0005051.html>

Any change to these parameters impacts the query I/O costing, which might change access plans for your queries. Do not change these parameters, unless there is strong evidence that a change will provide overall better performance for your entire query workload.

Temporary table spaces

For the IBM Smart Analytics System, use a DMS temporary table space because it provides good performance. IBM Smart Analytics System 5600 V1 and V2 with SSD, and 7700 use temporary table space containers located on SSD devices. The size of the SSD device might vary depending on your platform and configuration options.

All the DB2 table spaces are (by default) on a single container located on a dedicated file system per partition, except for the temporary table space when SSD is available on your specific environment.

By default, all DB2 containers are located on the same stripe set. When containers are located on the same stripe set, extents are allocated in a round robin fashion across the container. When DB2 containers are created on various stripe sets, extents are allocated sequentially (on one container first, then the other one).

Figure 7-1 shows an example of two unique table spaces:

- ▶ TEMP16K has two containers residing on the same stripe set. This stripe setting is the most commonly used and is used on the default table space creation. Extents are allocated on round robin fashion.
- ▶ TEMPB16K has two containers on two unique stripe sets. Extents are allocated sequentially. This stripe setting is used for DMS temporary table spaces with SSD containers on the IBM Smart Analytics System in order to use the SSD container first.

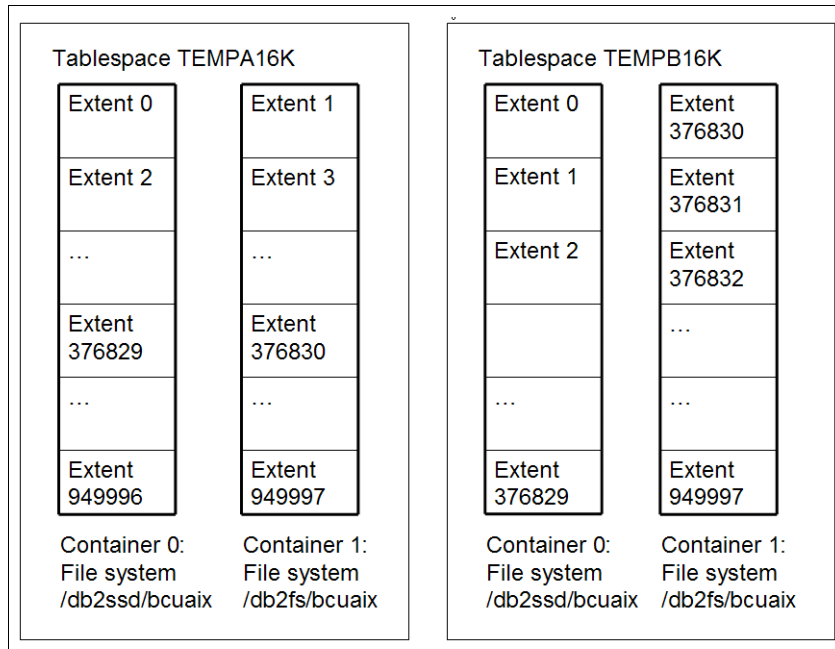


Figure 7-1 Table spaces created on the same and unique stripe sets

Example 7-29 shows the data definition language (DDL) to create these two table spaces.

Example 7-29 DDL to create table spaces

```
-- TABLESPACE TEMPA16K

CREATE TEMPORARY TABLESPACE "TEMPA16K"
IN DATABASE PARTITION GROUP IBMTEMPGROUP
PAGESIZE 16384 MANAGED BY DATABASE
USING (FILE '/db2ssd/bcuaix/ssd $N%8 /BCUDB/temp16k' 94208M,
        FILE '/db2fs/bcuaix/NODE000 $N /BCUDB/temp16k' 143292M)
ON DBPARTITIONNUMS (0 to 9)
USING (FILE '/db2ssd/bcuaix/ssd $N%8 /BCUDB/temp16k' 94208M,
        FILE '/db2fs/bcuaix/NODE00 $N /BCUDB/temp16k' 143292M)
ON DBPARTITIONNUMS (10 to 16)
EXTENTSIZE 16
PREFETCHSIZE 384
BUFFERPOOL BP16K
OVERHEAD 4.0
TRANSFERRATE 0.4
NO FILE SYSTEM CACHING
DROPPED TABLE RECOVERY OFF;
```

```

-- TABLESPACE TEMPB16K
CREATE TEMPORARY TABLESPACE TEMPB16K
IN DATABASE PARTITION GROUP IBMTEMPGROUP
PAGESIZE 16384 MANAGED BY DATABASE
USING (FILE '/db2ssd/bcuaix/ssd $N%8 /BCUDB/temp16k' 94208M) ON DBPARTITIONNUMS
(0 to 16)
EXTENTSIZE 16 PREFETCHSIZE 384
BUFFERPOOL BP16K OVERHEAD 4.0
NO FILE SYSTEM CACHING TRANSFERRATE 0.4;
COMMIT;

ALTER TABLESPACE TEMPB16K
BEGIN NEW STRIPE SET (FILE '/db2fs/bcuaix/NODE000 $N /BCUDB/temp16k' 143292M) ON
DBPARTITIONNUMS (0 to 9)
BEGIN NEW STRIPE SET (FILE '/db2fs/bcuaix/NODE00 $N /BCUDB/temp16k' 143292M) ON
DBPARTITIONNUMS (10 to 16);

```

The table space with the SSD container and the RAID disk container created on two unique stripe sets leverages better the SSD performance benefits, because DB2 will allocate all extents on the SSD container first, then the container on the RAID array. So, TEMP16K table space will have a DDL identical to TEMPB16K in the previous example.

Figure 7-2 shows the table space allocation on a 7700 platform with one 800 GB SSD RAID card.

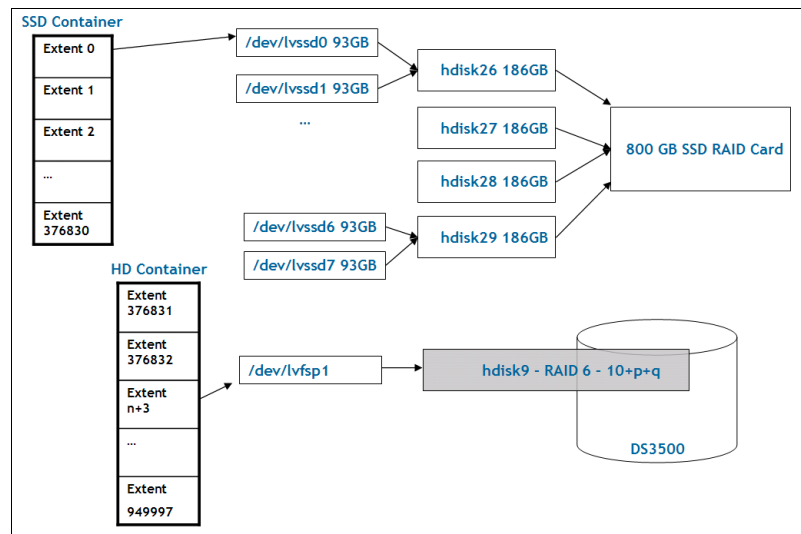


Figure 7-2 SSD container allocation

Note that table spaces on unique stripe sets can only be specified through a CREATE TABLESPACE statement, followed by an ALTER TABLESPACE. There is no syntax to create a table space on unique stripe sets with one statement.

db2look: Currently (up to DB2 9.7 Fix Pack 3a), **db2look** does not recognize, during DDL extraction, table spaces on unique stripe sets. The DDL generated by **db2look** will locate both containers on the same stripe set.

If you want to monitor your temporary table space spill usage, you can check the maximum high watermark for the temporary table space since the time the database was activated. Use the following methods to obtain the high water mark value:

- ▶ **db2pd -db <db-name> -tablespaces:** Column MaxHWM.
- ▶ **MON_GET_TABLESPACE:** This relational monitoring function contains a TBSP_MAX_PAGE_TOP column.

Example 7-30 shows an example of a **db2pd** output excerpt to capture the Max HWM statement. The output is limited to the data of interest. During the workload, the maximum HWM usage is 1 016 640 pages. The data listed under “Containers” shows that there are 6,029,312 pages in the first SSD container. So, the workload used about one sixth of the SSD container for spill purposes.

Example 7-30 Check high watermark using db2pd

```
# db2pd -db bcudb -tablespaces

Database Partition 1 -- Database BCUDB -- Active -- Up 0 days 00:50:34

Tablespace Configuration:
Address          Id      Type Content PageSz ExtentSz Auto Prefetch ...
0x07000001502683C0 260    DMS  SysTmp 16384 16      No   384      ...
...
...BufID BufIDDisk FSC NumCntrs MaxStripe LastConsecPg Name
...2      2          Off 2          1          15          TEMP16K

Tablespace Statistics:
Address          Id      TotalPgs UsablePgs UsedPgs PndFreePgs FreePgs ...
0x07000001502683C0 260    15200000 15199968 1016640 0          14183328...

...HWM          Max HWM      State      MinRecTime NQuiescers PathsDropped
...1016640      1016640     0x00000000 0          0          No

Containers:
Address          TspId ContainNum Type      TotalPgs UseablePgs ...
0x0700000150269880 260    0          File     6029312 6029296 ...
0x0700000150269A90 260    1          File     9170688 9170672 ...

...PathID      StripeSet Container
```



```

...-          0          /db2ssd/bcuaix/ssd1/BCUDB/temp16k
...-          1          /db2fs/bcuaix/NODE0001/BCUDB/temp16k

```

Example 7-31 shows an example of MON_GET_TABLESPACE relational monitoring function. The column TBSP_MAX_PAGE_TOP shows the maximum high watermark usage for TEMP16K table space.

Example 7-31 Check high watermark using MON_GET_TABLESPACE

```

SELECT SUBSTR(TBSP_NAME,1,20) AS TBSP_NAME,
TBSP_ID, MEMBER, TBSP_PAGE_TOP, TBSP_MAX_PAGE_TOP
FROM TABLE(MON_GET_TABLESPACE(' ', -2))
WHERE TBSP_NAME='TEMP16K' ORDER BY MEMBER

```

TBSP_NAME	TBSP_ID	MEMBER	TBSP_PAGE_TOP	TBSP_MAX_PAGE_TOP
TEMP16K	260	0	64	128
TEMP16K	260	1	601184	601184
TEMP16K	260	2	600896	600896
TEMP16K	260	3	634496	634496
TEMP16K	260	4	602144	602144
TEMP16K	260	5	628480	628480
TEMP16K	260	6	602688	602688
TEMP16K	260	7	672928	672928
TEMP16K	260	8	629728	629728

9 record(s) selected.

Note that these values represent the maximum HWM since the database was activated. During testing, if you want to see the maximum temporary table space size that a specific workload needs, perform the following steps:

1. Deactivate the database.
2. Activate the database.
3. Run the workload.
4. Collect the monitoring data prior to deactivating the database.

7.2 DB2 workload manager

The DB2 workload manager provides a powerful, low overhead capability in controlling the DB2 activities in execution based on the business priorities. You can use DB2 workload manager to tame system workload peaks to prevent overloading and control the priority of workloads. You can integrate the DB2 workload manager easily with AIX and Linux Workload Managers to have a cohesive management for the entire IBM Smart Analytics System.

In this section, we demonstrate how to progressively configure DB2 Workloads Manager to manage the workloads in an IBM Smart Analytics System. We use MARTS tables for our demonstration. The DDL scripts used in the examples for this section are provided in Appendix B, “Scripts for DB2 workload manager configuration” on page 299.

For the information about the best practice in using DB2 workload manager, see: <http://www.ibm.com/developerworks/data/bestpractices/workloadmanagement/>

7.2.1 Working with DB2 workload manager

There are two perspectives to take into account when designing a DB2 workload manager system:

- ▶ Business perspective: Considers business requirements about the process, applications, objectives, and performance expectations.
- ▶ System perspective: Reflects the realities of efficient system management.

The challenge of workload management is to map the business perspective to the system perspective as illustrated in Figure 7-3.

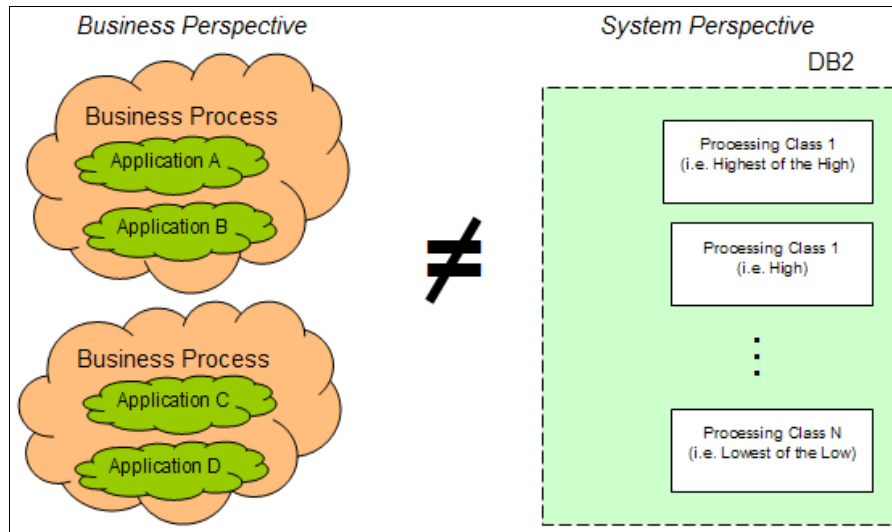


Figure 7-3 Business versus system perspective

A good strategy is to regulate the incoming workloads according to business priority and then manage the system capacity as efficiently as possible. The goal is to control the demands from business applications by managing the number of concurrent access and share of the resources among the applications.

The DB2 workload manager is able to identify *who* is submitting the workloads (by user or group, application, IP address, and other parameters). It can also determine *what* that user or application is to perform (for example, Data Manipulation Language (DML), DDL, stored procedures, or utilities).

Using this information, the DB2 workload manager can group together users, roles, or applications (*who*) with similar business priorities into *workloads*. And the type of operation to be carried out (*what*) can be used to instruct the *work action set* to take a pre-defined action. Figure 7-4 illustrates mapping the business function into workloads.

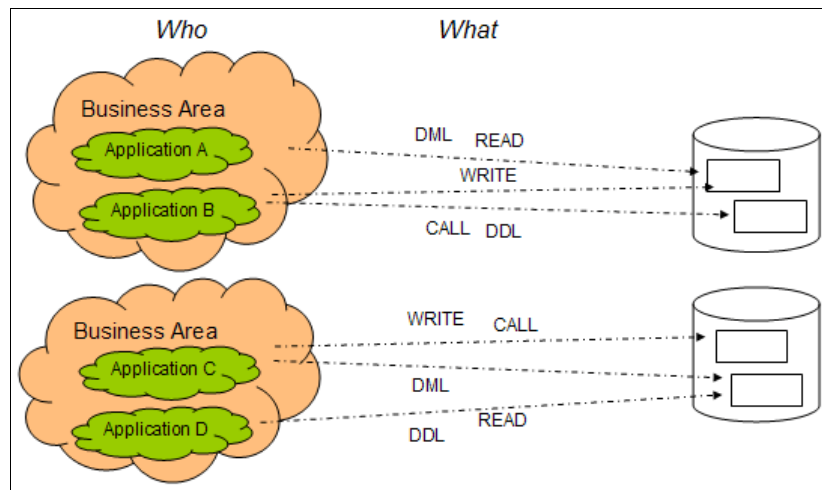


Figure 7-4 Mapping the workloads.

DB2 workload manager manages the work by using DB2 workloads and DB2 work action sets to place work into service classes where the work executes. The service class determines the priority and allocation of resources that the work receives during execution. Service classes have a two tier hierarchy; a service superclass contains one or more subclasses. A superclass always has a default subclass, and might have one to 64 user defined subclasses.

DB2 provides DDL statements for creating workload management objects. The following SQL statements are used exclusively to create, alter, drop, or manage workload management objects:

```
CREATE HISTOGRAM TEMPLATE, ALTER HISTOGRAM TEMPLATE or DROP
(HISTOGRAM TEMPLATE)
CREATE SERVICE CLASS, ALTER SERVICE CLASS, or DROP (SERVICE CLASS)
CREATE THRESHOLD, ALTER THRESHOLD, or DROP (THRESHOLD)
CREATE WORK ACTION SET, ALTER WORK ACTION SET, or DROP (WORK ACTION SET)
```

CREATE WORK CLASS SET, ALTER WORK CLASS SET, or DROP (WORK CLASS SET)
CREATE WORKLOAD, ALTER WORKLOAD, or DROP (WORKLOAD)
GRANT (Workload Privileges) or REVOKE (Workload Privileges)

Any workload management-exclusive SQL statements must be followed by a commit or rollback. Figure 7-5 shows DDL statements and workload management objects. You can use the GRANT and REVOKE statements to manage the privilege on a workload.

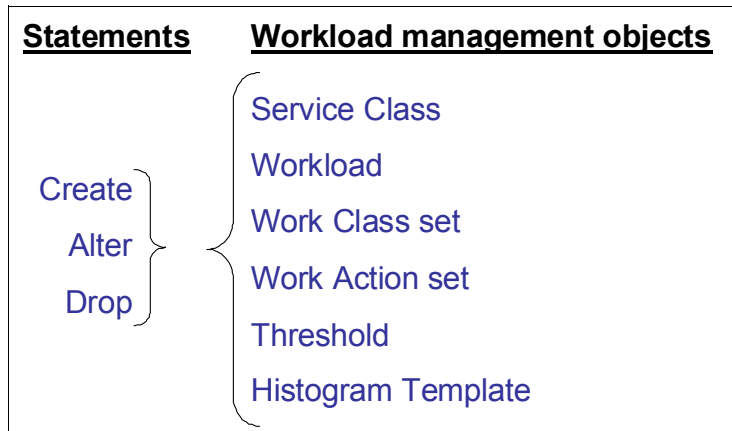


Figure 7-5 Managing DB2 workload manager objects

For more details about the DDL statements for DB2 workload manager, see DB2 Information Center at the following address:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.wlm.doc/doc/r0051422.html>

You can use **db2look** with the **-wlm** option to generate WLM specific DDL statements. See DB2 Information Center for details:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0002051.html>

DB2 provides table functions and routines for managing DB2 workload manager data, as follows:

- ▶ Workload management administrative SQL routines (see Table 18):

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.rtn.doc/doc/r0023485.html>

- ▶ Monitoring and intervention:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.wlm.doc/doc/c0052600.html>

These articles have good information about DB2 workload manager:

- ▶ *DB2 9.7: Using Workload Manager Features*
<http://www.ibm.com/developerworks/data/tutorials/dm-0908db2workload/index.html>
- ▶ *Smart Data Administration e-Kit: Article on DB2 Workload Management Histograms (3 Parts).*
<http://www.ibm.com/developerworks/data/kits/dbakit/index.html>

7.2.2 Configuring a DB2 workload manager for an IBM Smart Analytics System

In order to have a consistent plan for configuring the DB2 workload manager, perform the configuration process progressively starting from the monitor and understand the activities in the database. After you have learned the characteristics of the workloads, you can then gradually tune the DB2 workload manager configuration to set and enforce limits to each group of workloads to obtain system stability.

Default DB2 workload manager environment

Starting in Version 9.5, all DB2 server installations come with Workload Manager activated, although neither any action taken, nor any statistics captured for the work being executed. Any connection made to a DB2 database is assigned to a DB2 workload and any user request submitted by that connection is considered as part of that DB2 workload. DB2 considers all work within a workload as being from a common source and can be treated as a common set of work. If a connection does not match any user defined DB2 workloads, it is assigned to the default workload.

After a DB2 installation, there are three default service superclasses:

- ▶ **SYSDEFAULTUSERCLASS:** For user workloads
- ▶ **SYSDEFAULTSYSTEMCLASS:** For special system level tasks
- ▶ **SYSDEFAULTMAINTENANCECLASS:** For maintenance works such as statistics gathering or table reorganization

Initially, the DB2 workload manager is activated but not configured, so no user or application will be identified. Therefore, all requests will be handled by the default workload, which is mapped to the default user service subclass.

Figure 7-6 illustrates the default DB2 workload manager environment.

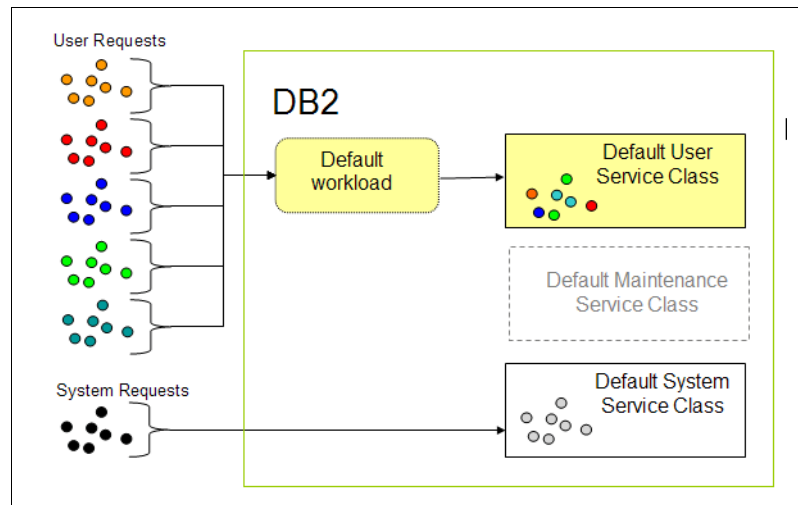


Figure 7-6 Default DB2 workload manager environment

For simplicity, we do not show the DefaultMaintenanceServiceClass and DefaultSystemServiceClass in the graphics shown in this section.

Untuned DB2 workload manager environment

In this section we describe how to set up a simple, untuned DB2 workload manager configuration that lets you monitor your workload. In a later section, we describe how to tune this configuration so that you can begin controlling your workload using the information that you obtained from monitoring it. The configuration of the untuned DB2 workload manager environment is as follows:

- ▶ One new superclass (named MAIN, for example)
- ▶ Six subclasses within the newly created user superclass (ETL, Trivial, Minor, Simple, Medium, and Complex)
- ▶ A work class set for redirecting the incoming workloads to the appropriate service subclass based on SQL cost and workload type.
- ▶ Remapping of the default workload from the default user service class to the new MAIN service superclass.

Figure 7-7 illustrates the untuned DB2 workload manager environment, where the workloads are assigned to the service superclass MAIN that has six subclasses. The default workload object is not changed.

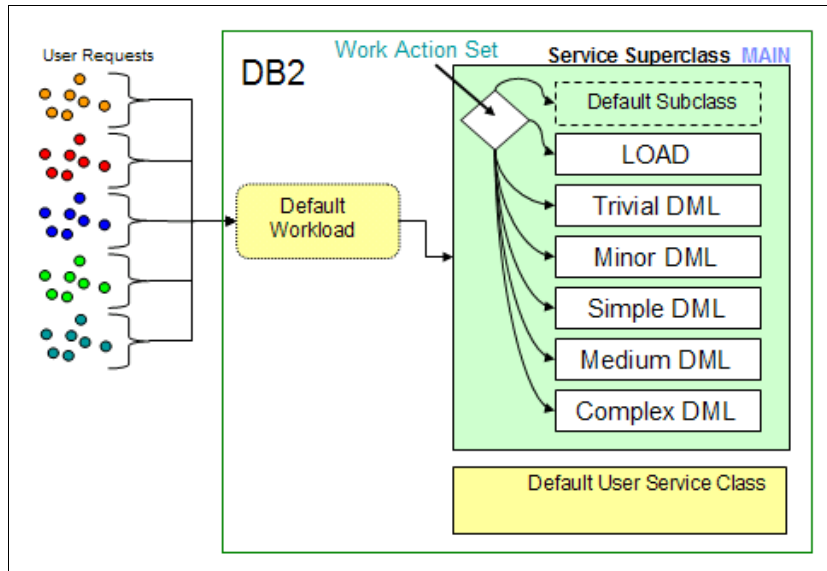


Figure 7-7 Untuned DB2 workload manager environment

Table 7-9 contains the suggested configuration for the workloads, regarding timeron ranges, execution time estimated, and the maximum allowable elapsed execution timeron for this environment.

Table 7-9 Initial configuration for workload management

Subclass	Expected time spread	Timeron range	Activity total time threshold criteria	Threshold actions
Default	Unknown	N/A	N/A	Collect Activity data & continue
ETL	Unknown	N/A	N/A	Collect Activity data & continue
Trivial	< 1 second	0 - 5000	1 minute	Collect Activity data & continue
Minor	< 60 seconds	5000 - 30,000	5 minutes	Collect Activity data & continue
Simple	< 5 minutes	30,000 - 300,00	30 minutes	Collect Activity data & continue
Medium	< 1 hour	300,000 - 5,000,000	60 minutes	Collect Activity data & continue
Complex	> 1 hour	5,000,000 - unbounded	240 minutes	Collect Activity data & continue

In this section, we demonstrate how to set up the untuned DB2 workload manager environment for an IBM Smart Analytics System with these parameters. All the scripts are provided in Appendix B, “Scripts for DB2 workload manager configuration” on page 299.

Creating service classes

Here we create the service superclass MAIN and the service subclasses: ETL, Trivial, Minor, Simple, Medium and Complex. We connect to our database and run the DDL script **01_create_svc_classes.sql** using the following command:

```
db2 -vtf 01_create_svc_classes.sql
```

Example 7-32 shows the new service classes created.

Example 7-32 Service classes created

```
SELECT VARCHAR(serviceclassname,30) AS SvcClass_name,  
       VARCHAR(parentserviceclassname,30) AS Parent_Class_name  
FROM syscat.serviceclasses  
WHERE parentserviceclassname = 'MAIN'
```

SVCCCLASS_NAME	PARENT_CLASS_NAME
-----	-----
COMPLEX	MAIN
ETL	MAIN
MEDIUM	MAIN
SIMPLE	MAIN
SYSDEFAULTSUBCLASS	MAIN
MINOR	MAIN
TRIVIAL	MAIN

7 record(s) selected.

For more details about the **create service class** statement, see DB2 Information Center at this address:

<http://publib.boulder.ibm.com/infocenter/db21uw/v9r7/index.jsp?topic=/com.ibm.db2.1uw.sql.ref.doc/doc/r0050550.html>

After the new MAIN superclass and its subclasses are created, we will remap the default workload from SYSDEFAULTUSERCLASS to the new MAIN service superclass, by altering SYSDEFAULTUSERWORKLOAD.

Example 7-33 shows the workload name, the service subclass name, the service superclass name, and the workload evaluation order, before and after remapping. The script for this task is **02_remap_dft_wk1.sql**.

Example 7-33 Remapping the default workload

```
db2admin@node01:~/WLM> db2 -vtf 02_remap_dft_wk1.sql
-
----- Original defaultUSERworkload mapping -----

select varchar(workloadname,25) as Workload_name,
varchar(serviceclassname,20) as SvClass_name,
varchar(parentserviceclassname,20) as Parent_Class_name, EvaluationOrder
as Eval_Order FROM syscat.workloads ORDER by 4

WORKLOAD_NAME          SVCLASS_NAME          PARENT_CLASS_NAME     EVAL_ORDER
-----
SYSDEFAULTUSERWORKLOAD SYSDEFAULTSUBCLASS    SYSDEFAULTUSERCLASS   1
SYSDEFAULTADMWORKLOAD  SYSDEFAULTSUBCLASS    SYSDEFAULTUSERCLASS   2

2 record(s) selected.
```

alter workload SYSDEFAULTUSERWORKLOAD SERVICE CLASS MAIN
DB20000I The SQL command completed successfully.

```
commit
DB20000I The SQL command completed successfully.
```

----- Remapped defaultUSERworkload -----

```
select varchar(workloadname,25) as Workload_name,
varchar(serviceclassname,20) as SvClass_name,
varchar(parentserviceclassname,20) as Parent_Class_name, EvaluationOrder
as Eval_Order FROM syscat.workloads ORDER by 4

WORKLOAD_NAME          SVCLASS_NAME          PARENT_CLASS_NAME     EVAL_ORDER
-----
SYSDEFAULTUSERWORKLOAD MAIN                  -                       1
SYSDEFAULTADMWORKLOAD SYSDEFAULTSUBCLASS    SYSDEFAULTUSERCLASS   2

2 record(s) selected.
```

MAIN service class: Because the default workload is now mapped to the MAIN service class, do not disable or drop the MAIN service class, otherwise, all data access to the database will be interrupted. If you must disable or drop the MAIN service class, remap the default workload to the original SYSDEFAULTUSERCLASS first, using the following statement:

```
ALTER workload SYSDEFAULTUSERWORKLOAD SERVICE CLASS SysDefaultUserClass
```

Creating work class sets and work action sets

Work action sets analyze an incoming workload and send it to a pre-defined service subclass, based on a number of conditions:

- ▶ Work type (READ, WRITE, CALL, DML, DDL, LOAD, and ALL)
- ▶ Timeron cost
- ▶ Cardinality
- ▶ Schema names (for CALL statements only)

Work action sets work hand-in-hand with *work class* sets. A work class set defines the conditions to be evaluated and a work action set references a work class set to operate.

Example 7-34 shows the DDL statements (**03_create_wk_action_set.sql**) for creating work class sets and work action sets for the untuned environment configuration using the criteria set in Table 7-9 on page 249.

Example 7-34 Work class sets and work action sets DDL

```
CREATE WORK CLASS SET "WORK_CLASS_SET_1"
(
  WORK CLASS "WCLASS_TRIVIAL" WORK TYPE DML FOR TIMERONCOST FROM 0   to 5000POSITION AT 1,
  WORK CLASS "WCLASS_MINOR"  WORK TYPE DML FOR TIMERONCOST FROM 5000 to 30000POSITION AT 2,
  WORK CLASS "WCLASS_SIMPLE"  WORK TYPE DML FOR TIMERONCOST FROM 30000 to 300000POSITION AT 3,
  WORK CLASS "WCLASS_MEDIUM"  WORK TYPE DML FOR TIMERONCOST FROM 300000 to 5000000POSITION AT 4,
  WORK CLASS "WCLASS_COMPLEX"  WORK TYPE DML FOR TIMERONCOST FROM 5000000 to UNBOUNDEDPOSITION AT 5,
  WORK CLASS "WCLASS_ETL"     WORK TYPE LOAD POSITION AT 6,
  WORK CLASS "WCLASS_OTHER"   WORK TYPE ALL POSITION AT 7
);

commit ;

CREATE WORK ACTION SET "WORK_ACTION_SET_1" FOR SERVICE CLASS "MAIN" USING WORK CLASS SET
"WORK_CLASS_SET_1"
(
  WORK ACTION "WACTION_TRIVIAL" ON WORK CLASS "WCLASS_TRIVIAL" MAP ACTIVITY WITHOUT NESTED TO
  "TRIVIAL",
  WORK ACTION "WACTION_MINOR"   ON WORK CLASS "WCLASS_MINOR"   MAP ACTIVITY WITHOUT NESTED TO "MINOR",
  WORK ACTION "WACTION_SIMPLE"  ON WORK CLASS "WCLASS_SIMPLE"  MAP ACTIVITY WITHOUT NESTED TO "SIMPLE"
  ,
  WORK ACTION "WACTION_MEDIUM"  ON WORK CLASS "WCLASS_MEDIUM"  MAP ACTIVITY WITHOUT NESTED TO "MEDIUM"
  ,
  WORK ACTION "WACTION_COMPLEX" ON WORK CLASS "WCLASS_COMPLEX" MAP ACTIVITY WITHOUT NESTED TO
  "COMPLEX",
  WORK ACTION "WACTION_ETL"     ON WORK CLASS "WCLASS_ETL"     MAP ACTIVITY WITHOUT NESTED TO "ETL"
);

commit;
```

For details about the CREATE WORK CLASS SET statement, see the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0050577.html>

For details about the CREATE WORK ACTION SET statement, see the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0050576.html>

Workloads: So far we have created a framework only. We did not implement any workload controls. All we are going to do for now is to *identify* and *monitor* the workloads. Nothing will be prevented from executing. The controls will be implemented in the next stage, the tuned DB2 DB2 workload manager environment.

Preparing for monitoring

The untuned DB2 workload manager environment is ready and we can start collecting data. To monitor the system, use event monitors. There are three DB2 workload manager related event monitors:

- ▶ *Statistics* event monitor: For capturing histograms, counts, and high watermarks
- ▶ *Activity* event monitor: For capturing details about activities in a workload or service class
- ▶ *Threshold* event monitor: For capturing details about thresholds violations

Even though the data collected by event monitors can be sent to a pipe or to a file, the output type chosen for IBM Smart Analytics System is the table output, so you can easily access the data for historical analysis.

Create a separate, dedicated table space to store the event monitor tables. This table space must span all database partitions, otherwise, event monitor data will be lost in the partitions with no event monitor tables. In our example, we create a table space TS_WLM_MON as shown in Example 7-35 using the script **04_create_wlm_tablespace.sql**.

Example 7-35 Table space creation

```
db2admin@node01:~/WLM> db2 -vtf 04_create_wlm_tablespace.sql
CREATE TABLESPACE TS_WLM_MON MAXSIZE 2G
DB20000I The SQL command completed successfully.

COMMIT
DB20000I The SQL command completed successfully.
```

Use the script DB2 provided, `~/sql11ib/misc/wlmevmon.dd1`, to create and activate the event monitors. Modify the script to reflect the table space name created for this event monitoring.

Example 7-36 shows the script, **05_wlmevmon.ddl script**, which is used to create the three event monitors, DB2ACTIVITIES, DB2STATISTICS, and DB2THRESHOLDVIOLATIONS, as well as all the necessary tables for these DB2 workload manager related monitors. Table space TS_WLM_MON is used.

Example 7-36 05_wlmevmon.ddl script

```
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;

--
-- Define the activity event monitor named DB2ACTIVITIES
--

CREATE EVENT MONITOR DB2ACTIVITIES
  FOR ACTIVITIES
  WRITE TO TABLE
  ACTIVITY (TABLE ACTIVITY_DB2ACTIVITIES
            IN TS_WLM_MON
            PCTDEACTIVATE 100),
  ACTIVITYSTMT (TABLE ACTIVITYSTMT_DB2ACTIVITIES
                IN TS_WLM_MON
                PCTDEACTIVATE 100),
  ACTIVITYVALS (TABLE ACTIVITYVALS_DB2ACTIVITIES
                IN TS_WLM_MON
                PCTDEACTIVATE 100),
  CONTROL (TABLE CONTROL_DB2ACTIVITIES
            IN TS_WLM_MON
            PCTDEACTIVATE 100)
  AUTOSTART;

--
-- Define the statistics event monitor named DB2STATISTICS
--

CREATE EVENT MONITOR DB2STATISTICS
  FOR STATISTICS
  WRITE TO TABLE
  SCSTATS (TABLE SCSTATS_DB2STATISTICS
            IN TS_WLM_MON
            PCTDEACTIVATE 100),
  WCSTATS (TABLE WCSTATS_DB2STATISTICS
            IN TS_WLM_MON
            PCTDEACTIVATE 100),
  WLSTATS (TABLE WLSTATS_DB2STATISTICS
            IN TS_WLM_MON
            PCTDEACTIVATE 100),
  QSTATS (TABLE QSTATS_DB2STATISTICS
            IN TS_WLM_MON
            PCTDEACTIVATE 100),
  HISTOGRAMBIN (TABLE HISTOGRAMBIN_DB2STATISTICS
                 IN TS_WLM_MON
                 PCTDEACTIVATE 100),
  CONTROL (TABLE CONTROL_DB2STATISTICS
            IN TS_WLM_MON
            PCTDEACTIVATE 100)
  AUTOSTART;

--
-- Define the threshold violation event monitor named DB2THRESHOLDVIOLATIONS
--

CREATE EVENT MONITOR DB2THRESHOLDVIOLATIONS
  FOR THRESHOLD VIOLATIONS
  WRITE TO TABLE
  THRESHOLDVIOLATIONS (TABLE THRESHOLDVIOLATIONS_DB2THRESHOLDVIOLATIONS
```

```

                IN TS_WLM_MON
                PCTDEACTIVATE 100),
CONTROL (TABLE CONTROL_DB2THRESHOLDVIOLATIONS
        IN TS_WLM_MON
        PCTDEACTIVATE 100)
AUTOSTART;

--
-- Commit work
--
COMMIT WORK;

```

For more details about creating event monitors, see the DB2 Information Center:

- ▶ Creating event monitor for activities statement:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0055061.html>
- ▶ Creating event monitor for statistics statement:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0055062.html>
- ▶ Creating event monitor for threshold violations statement:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0055063.html>

Starting monitoring

To monitor the system, you need to activate the event monitors. Example 7-37 shows how to activate the event monitors.

Example 7-37 Starting the event monitors

```

db2admin@node01:~/WLM> db2 -vtf 06_start_evt_monitors.sql
.
----- Monitor switches status -----
SELECT substr(evmonname,1,30) as evmonname, CASE WHEN
event_mon_state(evmonname) = 0 THEN 'Inactive' WHEN event_mon_state(evmonname)
= 1 THEN 'Active' END as STATUS FROM syscat.eventmonitors

```

EVMONNAME	STATUS
DB2ACTIVITIES	Inactive
DB2DETAILDEADLOCK	Active
DB2STATISTICS	Inactive
DB2THRESHOLDVIOLATIONS	Inactive

4 record(s) selected.

```

set event monitor db2activities state 1
DB20000I The SQL command completed successfully.

```

```
set event monitor db2statistics state 1
DB20000I The SQL command completed successfully.
```

```
set event monitor db2thresholdviolations state 1
DB20000I The SQL command completed successfully.
```

```
----- Monitor switches status -----
SELECT substr(evmonname,1,30) as evmonname, CASE WHEN
event_mon_state(evmonname) = 0 THEN 'Inactive' WHEN event_mon_state(evmonname)
= 1 THEN 'Active' END as STATUS FROM syscat.eventmonitors
```

EVMONNAME	STATUS
DB2ACTIVITIES	Active
DB2DETAILDEADLOCK	Active
DB2STATISTICS	Active
DB2THRESHOLDVIOLATIONS	Active

4 record(s) selected.

The event monitors collect information about the workloads and store it in memory, not into tables. For the statistics event monitor, the statistics are flushed to the tables periodically. Set the number of minutes you want the in-memory statistics to be flushed to the tables using the database parameter `WLM_COLLECT_INT`. Typical values are 30 or 60 minutes. The default is 0, which means never write in-memory data to tables.

You can also write the in-memory statistics to table manually using the procedure `WLM_COLLECT_STATS()`. The `WLM_COLLECT_STATS` procedure gathers statistics for service classes, workloads, work classes, and threshold queues and writes them to the statistics event monitor. The procedure also resets the statistics for service classes, workloads, work classes, and threshold queues. If there is no active statistics event monitor, the procedure only resets the statistics.

For more information about `WLM_COLLECT_STATS` procedure, see:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.rtn.doc/doc/r0052005.html>

Testing the environment: Work action sets

In this section, we demonstrate how to verify if the service classes were created correctly. We use a script to display the existing service superclasses and subclasses, execute queries of various timeron costs, and list the workload executed by subclass so we can verify where each of the queries was executed.

Before running this verification test, we must reset the Workload Manager statistics so we can see the results easily. Because the **reset** command is asynchronous, wait a few seconds for the counters to be zeroed before running the script.

Example 7-38 shows the result of the verification script. In this example, to save the time, we commented out the complex query in the script. To include the complex query, uncomment the query in the script **07_execs_by_subclasses.sql**.

Example 7-38 Executions by subclasses script

```
db2admin@node01:~/WLM> db2 -vtf 07_execs_by_subclasses.sql

===== Workloads executed by Subclasses =====
SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 20) SUPERCLASS, VARCHAR( SERVICE_SUBCLASS_NAME,
20) SUBCLASS, COORD_ACT_COMPLETED_TOTAL FROM
TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97('','-',1)) AS T WHERE SERVICE_SUPERCLASS_NAME
like 'MAIN%'

SUPERCLASS          SUBCLASS              COORD_ACT_COMPLETED_TOTAL
-----
MAIN                 SYSDEFAULTSUBCLASS    0
MAIN                 TRIVIAL                0
MAIN                 MINOR                  0
MAIN                 SIMPLE                 0
MAIN                 MEDIUM                0
MAIN                 COMPLEX                0
MAIN                 ETL                    0

7 record(s) selected.

executing queries...
.
===== query to be mapped to the TRIVIAL service subclass =====
select count(*) from MARTS.PRODUCT

1
-----
35259

1 record(s) selected.

===== query to be mapped to the MINOR service subclass =====
select count(*) from MARTS.time, MARTS.time, MARTS.store

1
-----
18248384

1 record(s) selected.
```

```
===== query to be mapped to the EASY service subclass =====
select count(*) from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME, MARTS.STORE
```

```
1
-----
1560605200
```

1 record(s) selected.

```
===== query to be mapped to the MEDIUM service subclass =====
select count_big(*) from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.PRCHS_PRFL_ANALYSIS
```

```
1
-----
12133022500.
```

1 record(s) selected.

```
===== query to be mapped to the COMPLEX service subclass =====
===== Workloads executed by Subclasses =====
SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 20) SUPERCLASS, VARCHAR( SERVICE_SUBCLASS_NAME,
20) SUBCLASS, COORD_ACT_COMPLETED_TOTAL FROM
TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97('',' ',-1)) AS T WHERE SERVICE_SUPERCLASS_NAME
like 'MAIN%'
```

SUPERCLASS	SUBCLASS	COORD_ACT_COMPLETED_TOTAL
MAIN	SYSDEFAULTSUBCLASS	0
MAIN	TRIVIAL	2
MAIN	MINOR	1
MAIN	SIMPLE	1
MAIN	MEDIUM	1
MAIN	COMPLEX	1
MAIN	ETL	0

7 record(s) selected.

Timeron: A timeron is a DB2 internal measure of the cost of executing an SQL query. Because it takes into account the system characteristics such as CPU speed, hard disk speed, memory available, and many others, the timeron count might vary between systems, even for the same query.

We use another script (`08_et1_subclass.sql`) to run the load operation to verify the ETL service subclass. Reset the counters and wait a few seconds before executing the script. Example 7-39 shows the results of our test.

Example 7-39 Testing the ETL service subclass

```
db2admin@node01:~/WLM> db2 call wlm_collect_stats
```

```
Return Status = 0
```

```
db2admin@node01:~/WLM> db2 -vtf 08_etl_subclass.sql
```

```
create table db2admin.PRODUCT like marts.product
```

```
DB20000I The SQL command completed successfully.
```

```
declare mycursor cursor for select * from marts.product
```

```
DB20000I The SQL command completed successfully.
```

Load from mycursor of cursor replace into db2admin.product

```
SQL3501W The table space(s) in which the table resides will not be placed in backup pending state since forward recovery is disabled for the database.
```

```
SQL1193I The utility is beginning to load data from the SQL statement " select * from marts.product".
```

```
SQL3500W The utility is beginning the "LOAD" phase at time "10/25/2010 17:55:51.138355".
```

```
SQL3519W Begin Load Consistency Point. Input record count = "0".
```

```
SQL3520W Load Consistency Point was successful.
```

```
SQL3110N The utility has completed processing. "35259" rows were read from the input file.
```

```
SQL3519W Begin Load Consistency Point. Input record count = "35259".
```

```
SQL3520W Load Consistency Point was successful.
```

```
SQL3515W The utility has finished the "LOAD" phase at time "10/25/2010 17:55:51.385153".
```

```
Number of rows read          = 35259
Number of rows skipped       = 0
Number of rows loaded        = 35259
Number of rows rejected      = 0
Number of rows deleted       = 0
Number of rows committed    = 35259
```

```
drop table db2admin.product
```

```
DB20000I The SQL command completed successfully.
```

```
=
```

```
===== Executed workloads status =====
```

```
SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 30) SUPERCLASS, VARCHAR( SERVICE_SUBCLASS_NAME, 20)
SUBCLASS, COORD_ACT_COMPLETED_TOTAL FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(' ', '-1')) AS T
WHERE SERVICE_SUPERCLASS_NAME like 'MAIN%'
```

SUPERCLASS	SUBCLASS	COORD_ACT_COMPLETED_TOTAL
MAIN	SYSDEFAULTSUBCLASS	5
MAIN	ETL	1
MAIN	TRIVIAL	5
MAIN	MINOR	0
MAIN	SIMPLE	0
MAIN	MEDIUM	0
MAIN	COMPLEX	0

```
7 record(s) selected.
```

We can see that the work action set correctly redirected the workloads to the corresponding subclasses. You might also have noticed that during the load operation, other executions were made and were shown in the SYSDEFAULTSUBCLASS and TRIVIAL subclasses.

Testing the environment: Concurrency

Now let us see how to monitor how many concurrent queries were submitted on each subclass. You can use your own workloads or use the scripts we provide to send concurrent queries to the database, and then check the monitors to see what happened.

We create two files with the queries to be run on the database. Example 7-40 shows content of easy_query.sql.

Example 7-40 Query for EASY service subclass (query_minor.sql)

```
select count(*) as Easy from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME, MARTS.STORE ;
```

Example 7-41 is the query in minor_query.sql.

Example 7-41 Query for MINOR service subclass (query_easy_query.sql)

```
select count(*) as Minor from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME;
```

For testing in UNIX environments, the **db2batch** utility is used to run these queries. Example B-15 on page 311 shows the **db2batch** script. Do not use the RUNSTATS command to update the database statistics prior to running the test. Example 7-42 shows the output of our script (**09_conc_exec_Unix.sh**) that starts the foregoing queries concurrently.

Example 7-42 Running the queries

```
db2admin@node01:~/WLM> ./09_conc_exec_Unix.sh
db2admin@node01:~/WLM> * Timestamp: Tue Oct 26 2010 08:10:25 CDT
* Timestamp: Tue Oct 26 2010 08:10:26 CDT
* Timestamp: Tue Oct 26 2010 08:10:26 CDT
* Timestamp: Tue Oct 26 2010 08:10:26 CDT
* Timestamp: Tue Oct 26 2010 08:10:26 CDT
-----
* SQL Statement Number 1:

SELECT COUNT(*) as Easy FROM empmdc, empmdc, suppliers ;

* Timestamp: Tue Oct 26 2010 08:10:28 CDT
-----
```

* SQL Statement Number 1:

```
SELECT COUNT(*) as Minor FROM empmdc, empmdc ;
```

(execution continues...)

Even though the **db2batch** output seems to indicate that the queries are being run serially, they were actually run in parallel. This situation can be seen by checking the AIX or Linux process status while the script is being executed (Example 7-43).

Example 7-43 Parallel execution

```
db2admin@node01:~/WLM> ps -ef |grep db2admin
db2admin 29100      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_minor.sql -a db2admin/ibm2blue -time off
db2admin 29101      1  1 10:29 pts/0    00:00:00 db2batch -d sample -f
query_Minor.sql -a db2admin/ibm2blue -time off
db2admin 29102      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_Minor.sql -a db2admin/ibm2blue -time off
db2admin 29103      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_Minor.sql -a db2admin/ibm2blue -time off
db2admin 29104      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_easy.sql  -a db2admin/ibm2blue -time off
db2admin 29105      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_easy.sql  -a db2admin/ibm2blue -time off
db2admin 29106      1  0 10:29 pts/0    00:00:00 db2batch -d sample -f
query_easy.sql  -a db2admin/ibm2blue -time off
db2admin 29145 12653  0 10:30 pts/1    00:00:00 ps -ef
db2admin 29146 12653  0 10:30 pts/1    00:00:00 grep db2admin
```

On Windows, use **db2cmd** to run the test. We provide a set of scripts for the Windows environment in the `09a_conc_exec_Win.bat` file in Appendix B, “Scripts for DB2 workload manager configuration” on page 299.

Example 7-44 shows results of the script (**10_conc_check.sql**) that checks the workload executions per subclass and the high water mark for the number of concurrent queries. Because we have not defined new workload objects yet, all the connections are under the default WLM workload object `SYSDEFAULTUSERWORKLOAD` as shown in the first set of output. See the second set of output.

Example 7-44 Checking concurrency

```
db2admin@node01:~/WLM> db2 -vtf 10_conc_check.sql
=
===== Queries executed by workloads =====

SELECT CONCURRENT_WLO_TOP, SUBSTR (WORKLOAD_NAME,1,25) AS WORKLOAD_NAME FROM
TABLE(WLM_GET_WORKLOAD_STATS_V97(CAST(NULL AS VARCHAR(128)), -2)) AS WLSTATS WHERE DBPARTITIONNUM = 0
ORDER BY WORKLOAD_NAME

CONCURRENT_WLO_TOP WORKLOAD_NAME
-----
0 SYSDEFAULTADMWORKLOAD
8 SYSDEFAULTUSERWORKLOAD

2 record(s) selected.

===== Workloads executed by Subclasses =====

SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 27) SUPERCLASS, VARCHAR( SERVICE_SUBCLASS_NAME, 18)
SUBCLASS, COORD_ACT_COMPLETED_TOTAL as NUMBER_EXECS, CONCURRENT_ACT_TOP as CONC_HWM FROM
TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(' ', '-1')) AS T

SUPERCLASS          SUBCLASS          NUMBER_EXECS      CONC_HWM
-----
SYSDEFAULTSYSTEMCLASS  SYSDEFAULTSUBCLASS      0          0
SYSDEFAULTMAINTENANCECLASS  SYSDEFAULTSUBCLASS      0          0
SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS      0          0
MAIN                    SYSDEFAULTSUBCLASS      0          0
MAIN                    ETL                      0          0
MAIN                    TRIVIAL                  8          1
MAIN                    MINOR                    4          4
MAIN                    SIMPLE                   3          3
MAIN                    MEDIUM                  0          0
MAIN                    COMPLEX                  0          0

10 record(s) selected.

db2admin@node01:~/WLM>
```

Monitoring service subclasses

Now that we have the service subclasses created in the MAIN service superclass, and the work action set is sending the queries to the appropriate service subclass, we will monitor the activities in each service subclass. As mentioned before, service subclasses are in charge of actually executing the queries sent to the database.

Referring to Table 7-9 on page 249, you can see that we are preparing to limit the concurrent execution in the ETL, MEDIUM, and COMPLEX subclasses. In the untuned WLM configuration, we set up a limit on the number of concurrent query execution. Also in Table 7-9 on page 249, you can see that, with the exception of the SysDefaultSubclass and ETL subclasses, all the subclasses have a timeout threshold to prevent runaway queries. The CREATE THRESHOLD statement is used. Example 7-45 shows the result of the `11_create_timeout_thresholds.sql` script.

Example 7-45 Script 11_create_timeout_Thresholds

```
db2admin@node01:~/WLM> db2 -vtf 11_create_timeout_thresholds.sql
```

THRESHOLD_NAME	THRESHOLD_TYPE	MAXVALUE
TH_TIME_SC_TRIVIAL	TOTALTIME	60
TH_TIME_SC_MINOR	TOTALTIME	300
TH_TIME_SC_SIMPLE	TOTALTIME	1800
TH_TIME_SC_MEDIUM	TOTALTIME	3600
TH_TIME_SC_COMPLEX	TOTALTIME	14400

```
5 record(s) selected.
```

```
db2admin@node01:~/WLM>
```

Setting the automatic statistics collection time interval

The default setting for the WLM_COLLECT_INT DB2 database parameter is 0 (zero), which means that the WLM statistics data collected by the monitors will never be sent to the event monitor output tables, nor reset. To keep a history of the system workload, you must set this the WLM_COLLECT_INT parameter to the desired time interval, in minutes. Because we want to monitor the system workload closely to properly adjust the service classes concurrency levels, an initial setting of five minutes interval was selected. After the service classes concurrency has been determined, this interval can be altered to 30 or 60 minutes. See Example 7-46.

Example 7-46 Setting the wlm_collect_int parameter

```
db2admin@node01:~/WLM> db2 update db cfg using WLM_COLLECT_INT 5
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
db2admin@node01:~/WLM>
```

For information about the wlm_collect_int parameter, see the website:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.config.doc/doc/r0051457.html>

Operating system level monitoring with NMON

To help determine the appropriate service classes concurrency number, you might have to refer to the overall system monitoring. An excellent tool for this task is the NMON tool and its companion, the NMON_analyzer. After collecting operating system workload statistics, you can look for overload periods (such as 100% CPU), and cross reference it to the DB2 WLM statistics to check if the concurrency levels need to be turned down in order to avoid the CPUs from reaching 100% utilization for long periods of time.

The NMON tool can be used interactively or in the data-collect mode. We use the second option, and save the results to a file by specifying the -f option (Example 7-47). The default parameters for the data-collection mode is to collect data at a five-minute interval (-s 300) for 24 hours (-c 288), then stop. This is the same time interval setting set for the WLM_COLLECT_INT parameter. The default name of the output file is <hostname>_YYYYMMDD_HHMM.nmon.

Example 7-47 Collecting NMON statistics

```
nmon -f
```

You can send the output file to your Windows client and analyze the data with the NMON_Analyzer tool. Figure 7-8 shows a sample analyzing report from the NMON_analyzer tool.

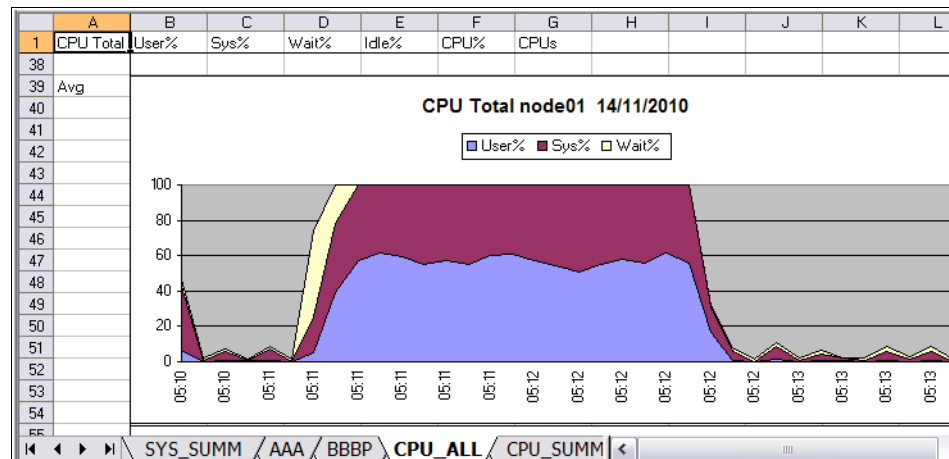


Figure 7-8 Analyzing data with NMON_analyzer tool

In Figure 7-8, we can see that the CPUs are spending almost 50% of the time processing *system* requests, instead of *user* requests. This indicates that the service sub-classes concurrency levels might need to be lowered. Running fewer queries at a time will leave more system resources to each of them, reducing *system* time and increasing *user* time. Indications of system overload are high process switch, high paging activity, and high run queues, shown in the PROC tab of the same NMON report.

In this configuration, the thresholds are not enforced yet, allowing any amount of queries to be executed simultaneously. So you can search for the actual concurrency level at the same time interval as the system peak observed in NMON and in the WLM control tables. Use those numbers as references when setting the threshold levels during next stage of WLM configuring.

To check the concurrency levels of queries already executed during a specific time period, you can use a query similar to the one in Example 7-48. If you use this script (**12_subclass_concurrency.sql**), be sure to adjust the date and time to the desired period before executing it. The concurrent activity top column of the report shows the number of queries executed during the last time interval.

Example 7-48 Checking service class concurrency during overload period

```
db2admin@node01:~/WLM> db2 -vtf 12_subclass_concurrency.sql
select concurrent_act_top , varchar(service_subclass_name,20) as subclass,
varchar(service_superclass_name,30) as superclass, statistics_timestamp from
scstats_db2statistics where statistics_timestamp between '2010-11-15-15.00.00'
and '2010-11-15-15.30.00'
```

CONCURRENT_ACT_TOP	SUBCLASS	SUPERCLASS	STATISTICS_TIMESTAMP
0	SYSDEFAULTSUBCLASS	SYSDEFAULTSYSTEMCLASS	2010-11-15-15.01.23.304973
2	SYSDEFAULTSUBCLASS	SYSDEFAULTMAINTENANCECLASS	2010-11-15-15.01.23.304973
0	SYSDEFAULTSUBCLASS	SYSDEFAULTUSERCLASS	2010-11-15-15.01.23.304973
0	SYSDEFAULTSUBCLASS	MAIN	2010-11-15-15.01.23.304973
0	ETL	MAIN	2010-11-15-15.01.23.304973
9	TRIVIAL	MAIN	2010-11-15-15.01.23.304973
4	MINOR	MAIN	2010-11-15-15.01.23.304973
3	SIMPLE	MAIN	2010-11-15-15.01.23.304973
7	MEDIUM	MAIN	2010-11-15-15.01.23.304973
4	COMPLEX	MAIN	2010-11-15-15.01.23.304973
0	SYSDEFAULTSUBCLASS	SYSDEFAULTSYSTEMCLASS	2010-11-15-15.06.23.812014
2	SYSDEFAULTSUBCLASS	SYSDEFAULTMAINTENANCECLASS	2010-11-15-15.06.23.812014
0	SYSDEFAULTSUBCLASS	SYSDEFAULTUSERCLASS	2010-11-15-15.06.23.812014
0	SYSDEFAULTSUBCLASS	MAIN	2010-11-15-15.06.23.812014
0	ETL	MAIN	2010-11-15-15.06.23.812014
8	TRIVIAL	MAIN	2010-11-15-15.06.23.812014
4	MINOR	MAIN	2010-11-15-15.06.23.812014
3	SIMPLE	MAIN	2010-11-15-15.06.23.812014
4	MEDIUM	MAIN	2010-11-15-15.06.23.812014
4	COMPLEX	MAIN	2010-11-15-15.06.23.812014
0	SYSDEFAULTSUBCLASS	SYSDEFAULTSYSTEMCLASS	2010-11-15-15.11.23.866814
2	SYSDEFAULTSUBCLASS	SYSDEFAULTMAINTENANCECLASS	2010-11-15-15.11.23.866814
0	SYSDEFAULTSUBCLASS	SYSDEFAULTUSERCLASS	2010-11-15-15.11.23.866814
0	SYSDEFAULTSUBCLASS	MAIN	2010-11-15-15.11.23.866814
0	ETL	MAIN	2010-11-15-15.11.23.866814
0	TRIVIAL	MAIN	2010-11-15-15.11.23.866814
0	MINOR	MAIN	2010-11-15-15.11.23.866814
0	SIMPLE	MAIN	2010-11-15-15.11.23.866814
0	MEDIUM	MAIN	2010-11-15-15.11.23.866814
0	COMPLEX	MAIN	2010-11-15-15.11.23.866814

30 record(s) selected.

```
db2admin@node01:~/WLM>
```

The NMON tool is included with AIX from 5.3 TL09 and AIX 6.1 TL02. For Linux, you can download it from this link:

<http://nmon.sourceforge.net/pmwiki.php>

For the NMON and NMON_Analyzer references, see:

<http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon>
http://www.ibm.com/developerworks/aix/library/au-nmon_analyser/

Monitoring the default workload

Though all users and applications are mapped to the default workload (SYSDEFAULTUSERWORKLOAD) only at this state, it is important to monitor and understand the queries sent through this default workload to understand the database activity. Monitoring workloads is useful when you have to gather information about who (or what) is sending the queries to the database. This monitoring can also be applied to user defined WLM workloads as well.

Example 7-49 shows how to collect the data in the default workload by altering the default workload with our script **13_alter_default_workload**.

Example 7-49 Altering the default workload

```
db2admin@node01:~/WLM> db2 -vtf 13_alter_default_workload.sql
alter workload sysdefaultuserworkload collect activity data on coordinator with
details
DB20000I The SQL command completed successfully.
```

The monitored data is collected each time the **call wlm_collect_stats()** statement is run. All the data collected is stored in the monitor tables created in Example 7-36 on page 254. There are a lot of details in those tables. As a starting point, you can use the SELECT statement provided in Example 7-50 to see what SQL statements were sent to database through default workload. The script used is **14_dftwkload_statements.sql**.

Example 7-50 Selecting default workload captured data

```
select varchar(session_auth_id,15) as user_name, varchar(appl_name,10) as
appl_name, varchar(workloadname,25) as workload_name,
varchar(service_superclass_name,10) as superclass,
varchar(service_subclass_name,10) as subclass, date(time_started) as date,
time(time_started) as time, varchar(stmt_text, 150) as statement_text from
wlm_event_stmt s, wlm_event e, syscat.workloads w where s.activity_id =
e.activity_id and s.appl_id = e.appl_id and s.uow_id = e.uow_id and
e.workload_id = 1 and e.workload_id = w.workloadid and date(e.time_started) =
date (current timestamp) fetch first 5 rows only
```

USER_NAME	APPL_NAME	WORKLOAD_NAME	SUPERCLASS	SUBCLASS	DATE
SLFERRARI	javaw.exe	SYSDEFAULTUSERWORKLOAD	MAIN	MINOR	11/02/2010
00:38:05	SELECT count(*), sum(length(packed_desc))/1024/4*2 from sysibm.systables				
SLFERRARI	javaw.exe	SYSDEFAULTUSERWORKLOAD	MAIN	MINOR	11/02/2010
01:05:01	VALUES (SUBSTR(CAST(? AS CLOB(56)), CAST(? AS INTEGER), CAST(? AS INTEGER)))				
SLFERRARI	javaw.exe	SYSDEFAULTUSERWORKLOAD	MAIN	MINOR	11/02/2010
00:38:08	SELECT count(*) from sysibm.systables where type='T' and creator <> 'SYSIBM'				
SLFERRARI	javaw.exe	SYSDEFAULTUSERWORKLOAD	MAIN	MINOR	11/02/2010
00:38:08	SELECT BPNAME, NPAGES, PAGESIZE FROM SYSIBM.SYSBUFFERPOOLS ORDER BY BPNAME				
SLFERRARI	javaw.exe	SYSDEFAULTUSERWORKLOAD	MAIN	MINOR	11/02/2010
00:39:00	select tb.bufferpoolid from syscat.tablespace tb where tb.tbspace = 'SYSCATSPACE' and not exists (select * from sysibm.systables st where st. SQL0445W Value "select tb.bufferpoolid from syscat.tablespace tb where t" has been truncated. SQLSTATE=01004				

5 record(s) selected with 1 warning messages printed.

Tuned DB2 workload manager environment

After monitoring the system for a period of time, you can start tuning DB2 workload manager parameters based on the monitor statistics to achieve a stable system operation. This can mean creating more workloads, implementing concurrency limits by workloads and by service classes, creating more service superclasses (2 to 5 total), and fine-tuning the SQL cost thresholds for the work action sets.

A tuned system can achieve an overall CPU utilization around 85-100%, with system CPU usage below 10%. Do not let the CPU work at 100% all the time. Turn down the concurrency levels for one or more service classes.

When applying changes to the system, make one change at a time and monitor the result. Then make another adjustment and monitor the result. This process can take a while to complete.

Attention: Try to keep the number of DB2 workload manager objects and configurations as simple as possible.

Here we demonstrate the process of adjusting the DB2 workload manager configuration. Figure 7-9 shows a high level picture of the database environment to be set up. Note that this configuration is for demonstration purposes only, and is not necessarily the preferable value. The adjustment can be based on the observations from the untuned configuration environment.

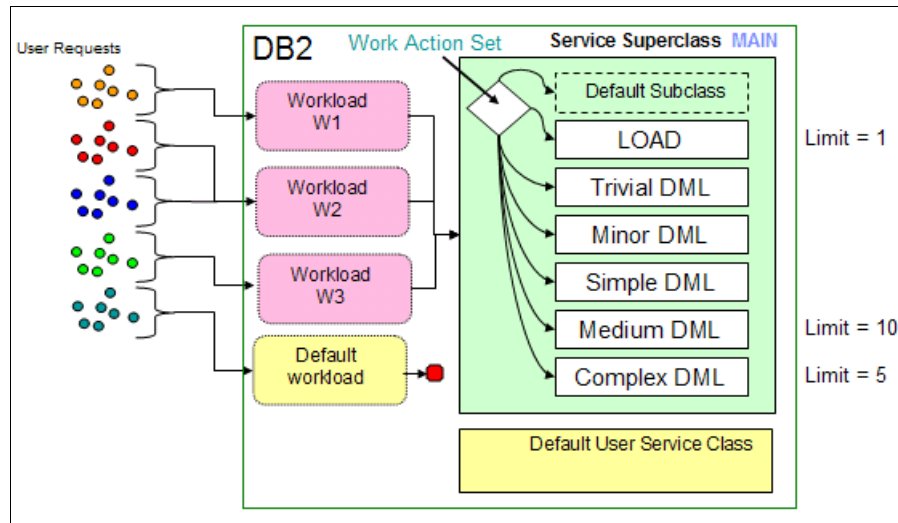


Figure 7-9 Tuned DB2 workload manager configuration

We categorized users and applications by roles and create new workloads to manage the workloads. With DB2 workload manager, you can set limits at the workload level to prevent a set of users monopolize the system, or to control the resources usages by applying limits to user sets based on the business objectives.

Administering a group as a workload allows you to apply particular monitor criteria to workloads including disable or enable workload monitoring separately.

Creating roles

Grouping users with similar workload behavior or business functions allows you to map their connections to a DB2 workload manager workload and apply unique rules to each workload. You can create DB2 roles to group users and applications. The rule of thumb in creating roles is to create only the roles needed and to keep the number roles as few as possible.

As an example, we create roles Adhoc, DBAs, PWRUSR, and Guest and user1 to user9 these roles. Example 7-51 shows the DDL statements. The script used is **50_create_roles.sql**.

Example 7-51 Creating DB2 roles

```
CREATE ROLE Adhoc;
GRANT ROLE Adhoc TO USER user1;
GRANT ROLE Adhoc TO USER user2;
GRANT ROLE Adhoc TO USER user3;
commit;
CREATE ROLE DBAs;
GRANT ROLE DBAs TO USER user4;
GRANT ROLE DBAs TO USER user5;
commit;
CREATE ROLE PWRUSR;
GRANT ROLE DBAs TO USER user6;
GRANT ROLE DBAs TO USER user7;
commit;
CREATE ROLE GUEST;
GRANT ROLE DBAs TO USER user8;
GRANT ROLE DBAs TO USER user9;
commit;
```

For more details about the CREATE ROLE statement, see the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0050615.html>

Creating additional workloads

Create new DB2 workload manager workloads and map the groups of users and applications to the workloads for a more granular control and monitoring. Creating new workloads allows you to map connections and unit of works that are to be monitored and controlled as peers while allowing the work being submitted to be treated in a common way by the system with work from all other workloads through evaluation and placement into the appropriate service subclass based on projected impact.

Any user or application not included in one of the new workloads is considered as an unknown user or application and will be handled by the default workload. Analyze these unclassified workloads and assign them to a proper workload. Do not create more workloads than necessary to group users or applications reasonably. Creating 5 to 20 workloads seems to be a good number.

A workload has to be mapped to a service class, either a superclass or a subclass. Mapping a workload to a service superclass allows the work action set to decide to which service subclass the SQL query will be set, based on the parameters such as timeron cost or SQL type.

Example 7-52 shows the DDL (**51_create_workloads.sql**) to create three workloads W1, W2, and W3 for the roles created in Example 7-51. You must grant the workload usage to the desired user, group, role, or public.

Example 7-52 Creating workloads

```
CREATE WORKLOAD W1 SESSION_USER ROLE ('DBAS') SERVICE CLASS MAIN POSITION AT 1
commit
GRANT USAGE on WORKLOAD W1 to public
commit
CREATE WORKLOAD W2 SESSION_USER ROLE ('ADHOC', 'PWRUSR') SERVICE CLASS MAIN POSITION AT 2
commit
GRANT USAGE on WORKLOAD W2 to public
commit
CREATE WORKLOAD W3 SESSION_USER ROLE ('GUEST') SERVICE CLASS MAIN POSITION AT 3
commit
GRANT USAGE on WORKLOAD W3 to public
commit
```

For more details about the CREATE WORKLOAD statement, see the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0050554.html>

Defining query concurrency using thresholds

In warehouse environments, with their typical long running queries, lowering query execution priority to avoid system overload by slowing them down, can potentially makes things even worse, because these queries will then be holding resources even longer than usual. Limit the number of concurrent workload executions to prevent the system from overloading in an IBM Smart Analytics System. This can be done at the workload level, at service subclass level, or at both, and is implemented by creating *thresholds*.

The idea is to execute fewer concurrent queries so they can finish faster, instead of trying to execute a large number of them at once. They will be competing for resources, which is not efficient. Limiting the number of concurrent queries has been proved to be a better solution in real-life systems as shown in Figure 7-10. Limiting the concurrency of large queries is far more effective than limiting smaller queries as well as (typically) more acceptable by the end-user population.

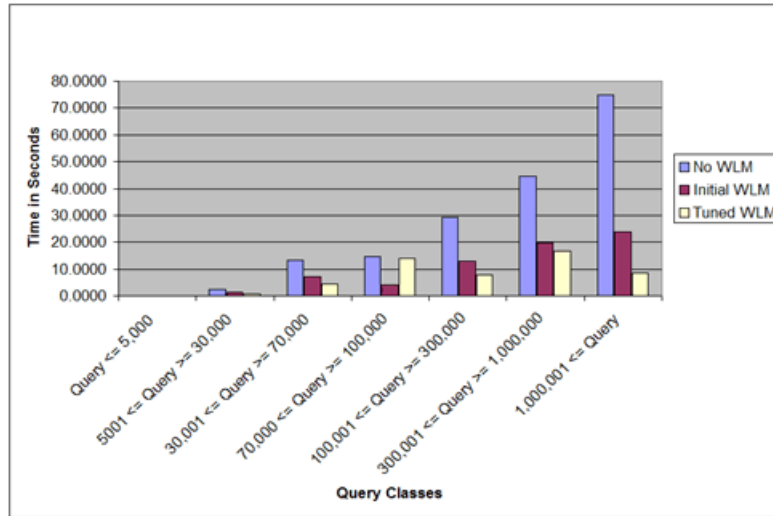


Figure 7-10 Real-life results

A threshold limit on a workload is about controlling the “share” of system resources that can be consumed by connections mapping to that workload. Depending on how the threshold is imposed, it can also control the share for particular classes of work submitted within that workload definition. Creating a threshold on workloads prevents a set of users or applications from using too many resources from the system. For example, if workload W3 has a limit of five concurrent executions, that will be the limit imposed, even if the system is idle and can handle more queries. This limit does not take into account the cost of a query. Any request beyond the first five workloads will be put on a queue.

A threshold limit on a service class is about controlling the “share” of system resources consumed by the class of work represented by that service class. Creating a threshold in the service class level (superclass or subclass) limits the execution at this particular level, regardless of who (or what) submitted it.

On the IBM Smart Analytics System, typically, the vast majority of queries are very small and quick and are executed in the Minor or Small service subclasses. They are usually left without a concurrency limit. Limiting the high cost subclass, or maybe the medium complexity subclasses, is far more effective. These few but demanding queries can overload the system. Another task that is often controlled is the load operation.

The appropriate values for the concurrency limits depends on your system workloads, as observed in the untuned DB2 workload manager configuration.

For each of the service subclasses (except for SYSDEFAULTSUBCLASS and ETL subclasses), also create a timeout threshold to prevent runaway queries from running much longer than expected for that service subclass.

Example 7-53 shows the script (**52_enforce_thresholds.sql**) to enforce the thresholds created earlier (untuned configuration) and its output. Use this script to change the concurrency levels of the thresholds to avoid system overload.

Example 7-53 Creating thresholds

```
ALTER THRESHOLD TH_TIME_SC_TRIVIAL WHEN    ACTIVITYTOTALTIME > 1 MINUTE COLLECT ACTIVITY
DATA on COORDINATOR WITH DETAILS STOP EXECUTION
```

```
ALTER THRESHOLD TH_TIME_SC_MINOR WHEN    ACTIVITYTOTALTIME > 5 MINUTES COLLECT ACTIVITY
DATA on COORDINATOR WITH DETAILS STOP EXECUTION
```

```
ALTER THRESHOLD TH_TIME_SC_SIMPLE WHEN    ACTIVITYTOTALTIME > 30 MINUTES COLLECT
ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION
```

```
ALTER THRESHOLD TH_TIME_SC_MEDIUM WHEN    ACTIVITYTOTALTIME > 60 MINUTES COLLECT
ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION
```

```
ALTER THRESHOLD TH_TIME_SC_COMPLEX WHEN    ACTIVITYTOTALTIME > 240 MINUTES COLLECT
ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION
```

```
select varchar(THRESHOLDNAME,25) as Threshold_name,
varchar(THRESHOLDPREDICATE,25) as Threshold_Type, maxvalue from
syscat.thresholds
```

THRESHOLD_NAME	THRESHOLD_TYPE	MAXVALUE
TH_TIME_SC_TRIVIAL	TOTALTIME	60
TH_TIME_SC_MINOR	TOTALTIME	300
TH_TIME_SC_SIMPLE	TOTALTIME	1800
TH_TIME_SC_MEDIUM	TOTALTIME	3600
TH_TIME_SC_COMPLEX	TOTALTIME	14400

8 record(s) selected.

Adjusting SQL cost range for a work action set

If required, you can readjust the attributes of a work action set. Example 7-54 (**53_alter_workclasses.sql**) shows how to change the boundary between the work classes SIMPLE and MEDIUM, from 300,000 to 400,000. The other thresholds remain unchanged.

Example 7-54 Change SQL cost thresholds command

```
db2admin@node01:~/WLM> db2 -vtf 53_alter_workclasses.sql
ALTER WORK CLASS SET "WORK_CLASS_SET_1" ALTER WORK CLASS "WCLASS_SIMPLE" FOR
TIMERONCOST FROM 30000 TO 40000 POSITION AT 3 ALTER WORK CLASS "WCLASS_MEDIUM"
FOR TIMERONCOST FROM 40000 TO 5000000 POSITION AT 4
DB20000I The SQL command completed successfully.
```

```
COMMIT WORK
DB20000I The SQL command completed successfully.
```

For more information about the CREATE THRESHOLD statement, see the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0050563.html>

Preventing unknown workload to execute

If you prefer to block any unidentified workload instead of monitoring them, you can do so by altering the default workload behavior. Although you cannot *disable* the default workload as with user-defined workloads, you can *disallow* it from accessing the database, using this command:

```
ALTER WORKLOAD sysdefaultuserworkload DISALLOW DB ACCESS
```

Warning: Make sure that the user ID to be used to disallow the default workload belongs to a WLM workload other than the default. Otherwise, after you disable the default workload, you will not be able to send any more commands to the database unless you are a *dbadm* or *wlmadm* authority and issue SET WORKLOAD TO SYSDEFAULTADMWORKLOAD.

7.2.3 DB2 workload manager resources

The following resources provide more details about DB2 workload manager:

- ▶ DB2 9.5 documentation:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>
- ▶ DB2 9.7 documentation:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>
- ▶ DB2 workload manager best practices:
<http://www.ibm.com/developerworks/data/bestpractices/workloadmanagement/>

- ▶ DB2 9.7: Using Workload Manager features:
<http://www.ibm.com/developerworks/data/tutorials/dm-0908db2workload/index.html>
- ▶ DB2 WLM Hands on Tutorial, in DB2 9.5 documentation:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.wlm.doc/doc/c0053139.html>
- ▶ developerWorks site to download the tutorial scripts
<http://www-128.ibm.com/developerworks/forums/servlet/JiveServlet/download/1116-179878-14005115-301960/wlmiodlab.zip>
- ▶ Article on DB2 Workload Management Histograms (3 Parts) in the Smart Data Administration e-Kit:
<http://www.ibm.com/developerworks/data/kits/dbakit/index.html>
- ▶ White paper: Workload Management with MicroStrategy Software and IBM DB2 9.5:
http://www-01.ibm.com/software/sw-library/en_US/detail/G407381L49488H62.html
- ▶ Exploitation of DB2's Workload Management in an SAP Environment:
<https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/d046f3f5-13c5-2b10-179d-80b6ae7b9657>
- ▶ IBM Redbooks publication:
<http://www.redbooks.ibm.com/redpieces/abstracts/sg247524.html>

7.3 Capacity planning

In Chapter 6, “Performance troubleshooting” on page 127, we discuss ways to monitor the various resources of the system which includes CPU, I/O, memory, and network. In various cases, you might determine that your current system does not allow you to meet your Service Level Agreements anymore, due to an increase or change in data volume and workload.

IBM Smart Analytics System offerings are highly scalable and offer various options in terms of capacity planning. In this section, we give an overview on identifying resource requirements, then we review the various options available for capacity planning.

7.3.1 Identifying resource requirements

When considering capacity planning, it is important to understand the current resource bottlenecks on your system. This can only be achieved if you have a clear picture of the current performance of your system through ongoing long term performance monitoring.

Here are various performance considerations regarding monitoring:

- ▶ OS level monitoring: CPU, I/O, memory, and network utilization collected on a regular basis
- ▶ DB2 level monitoring: DB2 performance metrics, such as buffer pool utilization, temporary table space usage, FCM buffers usage, number of applications connected, and nature of the query workload (rows read/written/returned for example)

The data allows matching the system resources utilization (CPU, I/O, memory, and network) to a given DB2 workload.

Ongoing long term performance monitoring allows you establish a baseline for the current performance of your system. This baseline is essential for trend and pattern analysis to confirm if potential bottlenecks results from legitimate resource requirements due to changes in the nature and concurrency of your workload.

Tools are available to perform this type of monitoring, such as the Performance Analytics feature, or Optim Performance Manager.

In order to ensure that resources are used optimally by DB2, it is essential to check that no additional tweaking or configuration changes might help from various perspectives:

- ▶ From an application perspective, monitor all the applications and make sure that there is no resource misuse (for example, rogue query due to bad access plan or poorly written query). This situation is the first item to check when the resource usage pattern shows a sudden change.
- ▶ From the database perspective, make sure that the database is well designed and maintained:
 - Best practices in database design: This approach includes absence of data skew, collocation of most frequent joined tables, appropriate indexes, and MQT for dimension tables.
 - Best practices in database maintenance: This approach includes up-to-date complete table and index statistics, including distribution statistics, and reorganization of the most frequently accessed and large tables when needed.

- Tuning: Review to see if the bottleneck can be alleviated through the database tunables to improve the overall efficiency of your database (BUFFERPOOL, SORTHEAP, and SHEAPTHRES tunables, for example).
- ▶ From an overall workload management perspective, look at opportunities to manage the workload on the system using the DB2 workload manager, and prioritize your workload. This action can result in an optimal use of the system resources by preventing conflicting workloads to run in parallel.

In terms of resource bottlenecks such as CPU, I/O, or network, the resource usage is tied to the type and concurrency of query and utility workload you are running. CPU and I/O saturation can be caused by poorly written queries or applications, or a poorly maintained database. Best practices in database design such as ensuring proper distribution keys, join collocation and the use of MQTs for dimension tables might help in reducing the usage of network.

For a memory bottleneck, do a memory usage analysis to understand how the memory is being used:

- ▶ Memory usage: Main memory consumers have to be accounted for. You need to have a clear understanding for the DB2 memory usage, and know what is being used in DB2 shared memory, DB2 private memory, and OS and kernel (kernel memory, file system caching, network buffers, and so on). This information is discussed in 6.3.3, “DB2 memory usage” on page 186.
- ▶ After you have a clear picture of how the memory is being used, look for opportunities in tuning down part of the memory usage by reducing the largest memory consumers (such as buffer pool or SHEAPTHRES) without affecting the baseline for your current level of performance.
- ▶ Consider IBM Smart Analytics System capacity planning options.

In terms of capacity planning, the IBM Smart Analytics System is highly scalable:

- ▶ There are options to increase the capacity of your existing servers in terms of CPU, memory, and SSD.
- ▶ You can scale up your database by adding additional data modules.

In the following sections, we examine the various options available to increase the capacity of your existing system.

7.3.2 Increasing capacity of existing systems

All the IBM Smart Analytics System family have options to increase the capacity of existing systems in terms of memory, CPU, and SSD when applicable. Table 7-10 lists these options for the various systems.

For all the options described next, the number of processors and amount of memory per server must be the same for the nodes contained in each of the following groups (but does not need to be the same across the groups):

- ▶ All data, administration, user, and standby nodes
- ▶ All business intelligence nodes
- ▶ All warehouse applications module nodes

5600 V1 and 5600 V2 environments

For 5600 V1 and 5600 V2 environments, you have the possibility to increase the CPU, memory, and SSDs for base servers to specifications.

Table 7-10 describes these options.

Table 7-10 Options for 5600 models

	5600 V1		5600 V2	
	5600 V1	5600 with SSD V1	5600 V2	5600 with SSD V2
CPU	Quad-core Intel Xeon X5570 (4 cores)	2 Quad-core Intel Xeon X5570 (8 cores)	1 6-core Intel X5680 (6 cores)	2 6-core Intel X5680 (12 cores)
Memory	32 GB	64 GB	64 GB	128 GB
SSD	n/a	2 x FusionIO 320 GB	n/a	2 x FusionIO 320 GB

Additional options are available for upgrading the network:

- ▶ Upgrading switches to 10 GbE (5600 V2 only)
- ▶ LAN-free backup option: Addition of a FC HBA dedicated to backup
- ▶ LAN-based backup option: Addition of a quad-port ethernet NIC dedicated to backup

Not all combinations of these options are available due to hardware restrictions. Consult with IBM Smart Analytics system support for further details.

7600 environment

For the 7600 environment, the following upgrade options are available:

- ▶ Double the CPU: Two additional dual-core 5.0 GHz POWER6® processors, so 8 cores in total
- ▶ Double the memory: Additional 32 GB available, so 64 GB in total

7700 environment

The 7700 environment has the following options available per server:

- ▶ SSD: By default, one 800 GB PCIe solid-state drive (SSD) RAID card is installed on the 7700. There are various options available to increase your SSD capacity:
 - Add one additional 800 GB SSD card per server.
 - Add an EXP12x expansion drawer to add one, three, or five more PCIe SSD cards to increase the total capacity to respectively two, four, or six 800 GB PCIe SSD cards per data node.
- ▶ Network: For LAN backups, there are possibilities to add:
 - LAN-free backup: One dual-port 8 Gb Fiber Ethernet PCIe adapter for dedicated LAN-free backup.
 - LAN-based backup: One quad-port 1Gb Copper Ethernet PCIe adapter, where one port can be used for a 1Gb Ethernet corporate network and a second port can be used for dedicated 1Gbps LAN-based backup. The other two ports are left available for other uses.

Certain restrictions apply in the combination of the previous options due to hardware limitations. Consult the *IBM Smart Analytics System 7700 User's Guide*, SC27-3571-01 for the restrictions.

7.3.3 Adding additional modules

Another option to scale your IBM Smart Analytics System is to add additional modules:

- ▶ An additional user module can be added if the bottleneck resides in the user module. This can be the case in an environment with a high number of connections.
- ▶ Additional data modules of the same release can be added to an existing IBM Smart Analytics System: This approach is commonly used when the data volume increases, and CPU and I/O resources are getting saturated to satisfy your workload.

Adding additional data modules allows you to decrease the amount of active data per database partition, and increase parallelism by adding additional CPU and I/O storage for the same amount of data.

Integrating additional servers into an existing cluster requires thorough planning. Key aspects of this planning include these:

- ▶ Integrate the new servers into your existing server infrastructure in terms of floor space, which includes power and cooling requirements.
- ▶ Integrate the new servers into your existing network (cabling, IP addresses).
- ▶ Configure the new servers as well as other servers in the cluster to ensure a well balanced and consistent environment (create the same users with the same UID and GID, same OS and kernel parameters, same firmware, and same software levels as the existing servers of the cluster).
- ▶ From a DB2 perspective, add additional database partitions. This will require you to redistribute part or all your data across all the database partitions, depending on your database partition groups layout.


Data redistribution can be done using the REDISTRIBUTE PARTITION GROUP command. Other options might be available to redistribute the data, depending on your requirements. Due to the numerous prerequisites and restrictions, this step requires thorough planning and comprehensive testing. This procedure can be time consuming, depending on the amount of data to be redistributed.

From a high level, the steps to add new database partitions are as follows:

- Add the new partitions to DB2 using, for example, **db2start...**
ADD DBPARTITIONNUM.
- Expand existing database partition groups to the newly added partitions using the command ALTER DATABASE PARTITION GROUP.
- Alter all table spaces belonging to the previously extended database partition groups to add table space containers on the new partitions, using the command ALTER TABLESPACE... ADD clause. This step can be time consuming on Linux platforms, because Linux does not have fast file preallocation.
- Data can be redistributed on each expanded database partition group using the command REDISTRIBUTE DATABASE PARTITION GROUP. This particular step has many prerequisites and restrictions. An essential prerequisite is to have a full backup and recovery point before engaging in a data redistribution activity. Consult the DB2 Information Center for additional details:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.partition.doc/doc/t0005017.html>

- ▶ From a Tivoli System Automation high availability perspective, integrate the additional servers into a high availability group, or create a new high availability group. High availability is discussed in Chapter 3, “High availability” on page 31.



Smart Analytics global performance monitoring scripts

In this appendix we list the formatting scripts discussed in 6.1.1, “Running performance troubleshooting commands” on page 129.

Example A-1 shows the global CPU performance monitoring Perl script `sa_cpu_mon.pl`.

Example A-1 `sa_cpu_mon.pl`

```
#!/usr/bin/perl

use strict;

# Choose which of the following two methods applies
# 1) on Management Node as user 'root' pick the first method
# 2) on Admin Node as DB2 instance owner pick the second method
my @nodes = `lsnode -N BCUALL`;
my @nodes = `cat ~/db2nodes.cfg | tr -s ' ' | cut -d' ' -f2 | sort | uniq`;

my $row = $nodes[0];
chomp $row;
my ($nodegroup, $odelist) = split (/: /,$row);

my $continuousloop = 'Y';
my $scriptparm;
my $nbrparms;
```

```

$nbrparms = $#ARGV + 1;
if ($nbrparms == 1)
{
    $scriptparm = $ARGV[0];
    chomp $scriptparm;
    if ($scriptparm eq "-s") { $continousloop = 'N' }
}

if (($nbrparms > 1) || (($nbrparms == 1) && ($scriptparm ne "-s")))
{
    print "Usage is: sa_cpu_mon.pl -s\n";
    print "where the optional parameter -s indicates 'snapshot'\n";
    print "versus default of continous looping.\n";
    exit 1;
}

my $nbrnodes = $nodes + 1;
my @nodeoutputfiles;
my $specific_node_output_file;
my @node_info_array;
my ($n,$m,$p) = 0;
my $nodesleft;
my $firstnodeoutput = 'Y';
my $node_info_row;

my $nodename;
my ($tot_runq, $tot_blockq, $tot_swapin, $tot_swapout, $tot_usrcpu, $tot_syscpu, $tot_idlecpu);
my ($tot_iowaitcpu, $tot_loadavg1min, $tot_loadavg5min, $tot_loadavg15min);
my ($avg_runq, $avg_blockq, $avg_swapin, $avg_swapout, $avg_usrcpu, $avg_syscpu, $avg_idlecpu);
my ($avg_iowaitcpu, $avg_loadavg1min, $avg_loadavg5min, $avg_loadavg15min);

my @array_nodename;
my @array_runq;
my @array_blockq;
my @array_usrcpu;
my @array_syscpu;
my @array_idlecpu;
my @array_iowaitcpu;
my @array_loadavg1min;
my @array_loadavg5min;
my @array_loadavg15min;

do
{
    $n = 0;
    $nodesleft = $nbrnodes;
    $firstnodeoutput = 'Y';
    while ($nodesleft)
    {
        chomp $nodes[$n];
        local *NODEOUT;
        open (NODEOUT, "ssh $nodes[$n] 'echo `hostname`: `vmstat 2 2 | tail -1` `uptime`' & |")
            || die "fork error: $!";
        $nodeoutputfiles[$n] = *NODEOUT;
        $n = $n + 1;
        $nodesleft = $nodesleft - 1;
    }

    reset_counters();

    $m = 0;
    foreach $specific_node_output_file (@nodeoutputfiles)
    {
        while (<$specific_node_output_file>)

```



```

    $avg_blockq      = $tot_blockq / $nbrnodes;
    $avg_usrcpu     = $tot_usrcpu / $nbrnodes;
    $avg_syscpu     = $tot_syscpu / $nbrnodes;
    $avg_idlecpu    = $tot_idlecpu / $nbrnodes;
    $avg_iowaitcpu  = $tot_iowaitcpu / $nbrnodes;
    $avg_loadavg1min = $tot_loadavg1min / $nbrnodes;
    $avg_loadavg5min = $tot_loadavg5min / $nbrnodes;
    $avg_loadavg15min = $tot_loadavg15min / $nbrnodes;

    print "          -----\n";

    printf ("%11s %6.1f %6.1f %5.1f %5.1f %5.1f %5.1f %7.2f %7.2f %7.2f\n",
            $nodename, $avg_runq, $avg_blockq, $avg_usrcpu, $avg_syscpu, $avg_idlecpu,
            $avg_iowaitcpu, $avg_loadavg1min, $avg_loadavg5min, $avg_loadavg15min);

    print "          -----\n";
}

sub format_and_print
{
    my $j = shift;
    printf ("%11s %6.1f %6.1f %5.1f %5.1f %5.1f %5.1f %5s %5s %5s\n",
            $array_nodename[$j], $array_runq[$j], $array_blockq[$j], $array_usrcpu[$j],
            $array_syscpu[$j],
            $array_idlecpu[$j], $array_iowaitcpu[$j], $array_loadavg1min[$j], $array_loadavg5min[$j],
            $array_loadavg15min[$j]);
}

sub reset_counters
{
    ($tot_runq, $tot_blockq, $tot_swapin, $tot_swapout, $tot_usrcpu, $tot_syscpu, $tot_idlecpu) = 0;;
    ($tot_iowaitcpu, $tot_loadavg1min, $tot_loadavg5min, $tot_loadavg15min) = 0;
    ($avg_runq, $avg_blockq, $avg_swapin, $avg_swapout, $avg_usrcpu, $avg_syscpu, $avg_idlecpu) = 0;;
    ($avg_iowaitcpu, $avg_loadavg1min, $avg_loadavg5min, $avg_loadavg15min) = 0;
}

```

Example A-2 shows the global I/O performance monitoring Perl script `sa_io_mon.pl`.

Example A-2 `sa_io_mon.pl`

```

#!/usr/bin/perl

# Script Name: sa_io_mon.pl
# Author      : Patrick Thoreson
# Company     : IBM
# Date        : Oct 7th, 2010

use strict;

# Choose which of the following two methods applies
# 1) on Management Node as user 'root' pick the first method
# 2) on Admin Node as DB2 instance owner pick the second method
my @nodes = `lsnode`;
#my @nodes = `cat ~/db2nodes.cfg | tr -s ' ' | cut -d' ' -f2 | sort | uniq`;

my $row = $nodes[0];
chomp $row;

```

```

my ($nodegroup, $odelist) = split (/: /,$row);

my $continousloop = 'Y';
my $scriptparm;
my $nbrparms;

$nbrparms = $#ARGV + 1;
if ($nbrparms == 1)
{
    $scriptparm = $ARGV[0];
    chomp $scriptparm;
    if ($scriptparm eq "-s") { $continousloop = 'N' }
}

if (($nbrparms > 1) || (($nbrparms == 1) && ($scriptparm ne "-s")))
{
    print "Usage is: sa_io_mon.pl -s\n";
    print "where the optional parameter -s indicates 'snapshot'\n";
    print "versus default of continous looping.\n";
    exit 1;
}

my $na;

my $nbrnodes = $nodes + 1;
my @nodeoutputfiles;
my $specific_node_output_file;
my ($n,$m,$p) = 0;
my $nodesleft;
my $firstnodeoutput = 'Y';
my $node_info_row;
my @node_info_array;

my $nodename;
my $blockq;
my $blockq_info;
my ($usrcpu,$syscpu,$idlecpu,$iowaitcpu);
my $iostat_output;
my $iostat_output2;
my $cpu_info;
my $io_info;
my $iodev;
my ($tps, $rtps, $wtps, $rKBps, $wKBps);
my $iodevinfo;
my $iodevremainder;
my $iodevnewremainder;
my $device_count;
my $devutil;
my $cumulative_devutil;
my $node_devutil;
my ($cumulative_rtps, $cumulative_wtps, $cumulative_rKBps, $cumulative_wKBps);
##my ($node_navg_rKBps, $node_ntot_rKBps, $node_navg_rtps, $node_ntot_rtps, $node_navg_wtps,
$node_ntot_wtps);
my ($node_ntot_tps, $node_ntot_rtps, $node_ntot_wtps);
##my ($node_navg_rKBps, $node_ntot_rKBps, $node_navg_rKB, $node_ntot_rKB, $node_navg_wKBps,
$node_ntot_wKBps, $node_navg_wKB, $node_ntot_wKB);
my ($node_ntot_rKBps, $node_ntot_rKB, $node_ntot_wKBps, $node_ntot_wKB);
my $device_in_use;
my $device_light_use;
my $device_medium_use;
my $device_heavy_use;
my $device_near_max_use;

my ($tot_blockq, $tot_usrcpu, $tot_syscpu, $tot_idlecpu, $tot_iowaitcpu, $tot_device_count,
$tot_devutil);

```

```

my ($tot_device_in_use, $tot_device_light_use, $tot_device_medium_use, $tot_device_heavy_use,
    $tot_device_near_max_use);
##my ($tot_navg_tps, $tot_navg_rKB, $tot_navg_wKB, $tot_ntot_tps, $tot_ntot_rKB, $tot_ntot_wKB);
my ($tot_ntot_tps, $tot_ntot_rKB, $tot_ntot_wKB);
my ($savg_blockq, $savg_usrcpu, $savg_syscpu, $savg_idlecpu, $savg_iowaitcpu, $savg_device_count,
    $savg_devutil);
my ($savg_device_in_use, $savg_device_light_use, $savg_device_medium_use, $savg_device_heavy_use,
    $savg_device_near_max_use);
##my ($savg_navg_tps, $savg_navg_rKB, $savg_navg_wKB, $savg_ntot_tps, $savg_ntot_rKB,
    $savg_ntot_wKB);
my ($savg_ntot_tps, $savg_ntot_rKB, $savg_ntot_wKB);

my @array_nodename;
my @array_runq;
my @array_blockq;
my @array_usrcpu;
my @array_syscpu;
my @array_idlecpu;
my @array_iowaitcpu;
my @array_device_count;
my @array_devutil;
##my @array_navg_tps;
##my @array_navg_rKB;
##my @array_navg_wKB;
my @array_ntot_tps;
my @array_ntot_rKB;
my @array_ntot_wKB;
my @array_device_in_use;
my @array_device_light_use;
my @array_device_medium_use;
my @array_device_heavy_use;
my @array_device_near_max_use;

do
{
    $n = 0;
    $nodesleft = $nbrnodes;
    $firstnodeoutput = 'Y';
    while ($nodesleft)
    {
        chomp $nodes[$n];
        local *NODEOUT;
        open (NODEOUT, "ssh $nodes[$n] `echo `hostname`:AAAAA`iostat -k -x 5 2`AAAAA`grep
procs_blocked /proc/stat` & |")
            || die "fork error: $!";
        $nodeoutputfiles[$n] = *NODEOUT;
        $n = $n + 1;
        $nodesleft = $nodesleft - 1;
    }

    reset_counters();

    $m = 0;
    foreach $specific_node_output_file (@nodeoutputfiles)
    {
        while (<$specific_node_output_file>)
        { if ("{$firstnodeoutput}" eq "Y")
            { header(); $firstnodeoutput = "N"; } $node_info_row = $_ ; extract_info($m); }
        close $specific_node_output_file || die "child cmd error: $! $?";
        $m = $m + 1;
    }

    compute_and_print_system_summary();

    for ($p = 0; $p < $nbrnodes; $p++)
    {

```

```

        format_and_print($p);
    }
} while ($Continuousloop eq 'Y');

sub extract_info
{
    my $i = shift;
    chomp $node_info_row;

    ($device_in_use, $device_light_use, $device_medium_use, $device_heavy_use, $device_near_max_use)
    = 0;
    ($tps, $rtps, $wtps, $rKBps, $wKBps) = 0;
    $blockq_info = '';

    # The $na variable is 'not applicable', i.e. we don't need it's value (it's simply a placeholder):
    ($nodename, $iostat_output, $blockq_info)
    = split('AAAAA',$node_info_row,3);
    ($na, $blockq)
    = split(' ', $blockq_info, 2);
    ($na, $na, $iostat_output2)
    = split('avg-cpu: ', $iostat_output, 3);
    ($cpu_info, $io_info)
    = split('Device: ', $iostat_output2, 2);
    ($na, $na, $na, $na, $na, $na, $usrcpu, $na, $syscpu, $iowaitcpu, $na, $idlecpu, $na)
    = split(' ', $cpu_info, 13);
    ($na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na, $na)
    = split(' ', $io_info, 20);

    $device_count = 0;
    $node_devutil = 0;
    $cumulative_devutil = 0;
    $node_ntot_tps = 0;
    ## $node_navg_tps = 0;
    $node_ntot_rtps = 0;
    ## $node_navg_rtps = 0;
    $cumulative_rtps = 0;
    $node_ntot_wtps = 0;
    ## $node_navg_wtps = 0;
    $cumulative_wtps = 0;
    $node_ntot_rKBps = 0;
    ## $node_navg_rKBps = 0;
    $node_ntot_rKB = 0;
    ## $node_navg_rKB = 0;
    $cumulative_rKBps = 0;
    $node_ntot_wKBps = 0;
    ## $node_navg_wKBps = 0;
    $node_ntot_wKB = 0;
    ## $node_navg_wKB = 0;
    $cumulative_wKBps = 0;
    while ($iodevremainder)
    {
        ($iodev, $na, $na, $na, $rtps, $wtps, $rKBps, $wKBps, $na, $na, $na, $na, $devutil,
        $iodevnewremainder)
        = split(' ', $iodevremainder, 13);
        $iodevremainder = $iodevnewremainder;
        $iodevnewremainder = '';
    }
    #
    # On the IBM Smart Analytics 5600 in our lab, the "sdb", "sdc", "sdd" and "sde" devices IO stats are
    # covered by using the IO stats of "dm-0", "dm-1", "dm-2" and "dm-3".
    # since they are mapped to the very same physical LUN device; for example, if we were to count the
    # stats of both "sdc" and "dm-1" we would be counting
    # the real IO stats twice for that real physical device.

```

```

# Hence, to avoid this redundant "double-counting" of io device statistics, we skip collecting stat
# for "sdb", "sdc", "sdd" and "sde"
# as they will be collected already under "dm-0", "dm-1", "dm-2" and "dm-3".
#
#   if ( ("${iodev}" eq "sdb") || ("${iodev}" eq "sdc") || ("${iodev}" eq "sdd") || ("${iodev}" eq "sde")
) { next; }
    $device_count = $device_count + 1;
    $cumulative_devutil = $cumulative_devutil + $devutil;
    $cumulative_rtps = $cumulative_rtps + $rtps;
    $cumulative_wtps = $cumulative_wtps + $wtps;
    $cumulative_rKBps = $cumulative_rKBps + $rKBps;
    $cumulative_wKBps = $cumulative_wKBps + $wKBps;
    if ( $devutil > 0 ) { $device_in_use = $device_in_use + 1; };
    if ((($devutil > 0) && ($devutil < 30)) { $device_light_use = $device_light_use + 1; };
    if ((($devutil >= 30) && ($devutil < 60)) { $device_medium_use = $device_medium_use + 1; };
    if ((($devutil >= 60) && ($devutil < 90)) { $device_heavy_use = $device_heavy_use + 1; };
    if ( $devutil >= 90 ) { $device_near_max_use = $device_near_max_use + 1; };
}

$node_devutil = $cumulative_devutil / $device_count;
$node_ntot_rtps = $cumulative_rtps;
## $node_navg_rtps = $cumulative_rtps / $device_count;
$node_ntot_wtps = $cumulative_wtps;
## $node_navg_wtps = $cumulative_wtps / $device_count;
$node_ntot_tps = $node_ntot_rtps + $node_ntot_wtps;
## $node_navg_tps = $node_navg_rtps + $node_navg_wtps;

$node_ntot_rKBps = $cumulative_rKBps;
## $node_navg_rKBps = $cumulative_rKBps / $device_count;
$node_ntot_wKBps = $cumulative_wKBps;
## $node_navg_wKBps = $cumulative_wKBps / $device_count;
#
# Multiply the rKBps (read KB per second) by 5 seconds since that is the interval we used in the
parallel ssh commands in this script to obtain the rKB;
# do the same with wKBps to obtain the wKB.

# If that time interval changes to another number, adjust in both
multiplication lines below:

# (for example: iostat -k -x 5 2 --> iostat -k -x 10 2, change the '$node_rKBps * 5' to
'$node_rKBps * 10', and do the same for $node_wKBps)
#
$node_ntot_rKB = $node_ntot_rKBps * 5;
$node_ntot_wKB = $node_ntot_wKBps * 5;
## $node_navg_rKB = $node_navg_rKBps * 5;
## $node_navg_wKB = $node_navg_wKBps * 5;

$tot_blockq = $tot_blockq + $blockq;
$tot_usrcpu = $tot_usrcpu + $usrcpu;
$tot_syscpu = $tot_syscpu + $syscpu;
$tot_idlecpu = $tot_idlecpu + $idlecpu;
$tot_iowaitcpu = $tot_iowaitcpu + $iowaitcpu;
$tot_device_count = $tot_device_count + $device_count;
$tot_devutil = $tot_devutil + $node_devutil;
$tot_device_in_use = $tot_device_in_use + $device_in_use;
$tot_device_light_use = $tot_device_light_use + $device_light_use;
$tot_device_medium_use = $tot_device_medium_use + $device_medium_use;
$tot_device_heavy_use = $tot_device_heavy_use + $device_heavy_use;
$tot_device_near_max_use = $tot_device_near_max_use + $device_near_max_use;
## $tot_navg_tps = $tot_navg_tps + $node_navg_tps;
## $tot_navg_rKB = $tot_navg_rKB + $node_navg_rKB;
## $tot_navg_wKB = $tot_navg_wKB + $node_navg_wKB;
$tot_ntot_tps = $tot_ntot_tps + $node_ntot_tps;
$tot_ntot_rKB = $tot_ntot_rKB + $node_ntot_rKB;
$tot_ntot_wKB = $tot_ntot_wKB + $node_ntot_wKB;

```

```

$array_nodename[$i] = $nodename;
$array_blockq[$i] = $blockq;
$array_usrcpu[$i] = $usrcpu;
$array_syscpu[$i] = $syscpu;
$array_idlecpu[$i] = $idlecpu;
$array_iowaitcpu[$i] = $iowaitcpu;
$array_device_count[$i] = $device_count;
$array_devutil[$i] = $node_devutil;
$array_device_in_use[$i] = $device_in_use;
$array_device_light_use[$i] = $device_light_use;
$array_device_medium_use[$i] = $device_medium_use;
$array_device_heavy_use[$i] = $device_heavy_use;
$array_device_near_max_use[$i] = $device_near_max_use;
## $array_navg_tps[$i] = $node_navg_tps;
## $array_navg_rKB[$i] = $node_navg_rKB;
## $array_navg_wKB[$i] = $node_navg_wKB;
$array_ntot_tps[$i] = $node_ntot_tps;
$array_ntot_rKB[$i] = $node_ntot_rKB;
$array_ntot_wKB[$i] = $node_ntot_wKB;
}

sub header
{
    if ($continuousloop eq 'Y') { system ("clear"); }
    ## print "sa_io_mon ----- CPU -----"
    ----- IO Device Usage -----
    ## print "
    Block Tot Tot Tot Tot Tot Tot Avg/dev #Active --
    Nbr devices in %util range -- --- Avg/device for node ---- --- Tot all devices on node --\n";
    ## print "
    Queue usr sys idle wio #devices %util devices
    0-30% 30-60% 60-90% 90-100% tps readKB writeKB tps readKB
    writeKB\n";

    print "sa_io_mon ----- CPU -----"
    ----- IO Device Usage -----
    ## print "
    Block Tot Tot Tot Tot Tot Avg/dev #Active --
    Nbr devices in %util range -- --- Tot all devices on node --\n";
    ## print "
    Queue usr sys idle wio #devices %util devices
    0-30% 30-60% 60-90% 90-100% tps readKB writeKB\n";
}

sub compute_and_print_system_summary
{
    $nodename = "System Avg:";
    $savg_blockq = $tot_blockq / $nbrnodes;
    $savg_usrcpu = $tot_usrcpu / $nbrnodes;
    $savg_syscpu = $tot_syscpu / $nbrnodes;
    $savg_idlecpu = $tot_idlecpu / $nbrnodes;
    $savg_iowaitcpu = $tot_iowaitcpu / $nbrnodes;
    $savg_device_count = $tot_device_count / $nbrnodes;
    $savg_devutil = $tot_devutil / $nbrnodes;
    $savg_device_in_use = $tot_device_in_use / $nbrnodes;
    $savg_device_light_use = $tot_device_light_use / $nbrnodes;
    $savg_device_medium_use = $tot_device_medium_use / $nbrnodes;
    $savg_device_heavy_use = $tot_device_heavy_use / $nbrnodes;
    $savg_device_near_max_use = $tot_device_near_max_use / $nbrnodes;
    ## $savg_navg_tps = $tot_navg_tps / $nbrnodes;
    ## $savg_navg_rKB = $tot_navg_rKB / $nbrnodes;
    ## $savg_navg_wKB = $tot_navg_wKB / $nbrnodes;
    $savg_ntot_tps = $tot_ntot_tps / $nbrnodes;
    $savg_ntot_rKB = $tot_ntot_rKB / $nbrnodes;
    $savg_ntot_wKB = $tot_ntot_wKB / $nbrnodes;
}

```

```

## print "          -----
-----\n";
print "          -----\n";

## printf ("%11s %6.1f %6.2f %6.2f %6.2f %6.2f %6.0f %6.2f %5.1f %5.1f %5.1f
%5.1f %5.1f %8.1f%11.1f%11.1f %8.1f%12.1f%12.1f\n",
printf ("%11s %6.1f %6.2f %6.2f %6.2f %6.2f %6.0f %6.2f %5.1f %5.1f %5.1f
%5.1f %5.1f %8.1f%12.1f%12.1f\n",
        $nodename, $savg_blockq, $savg_usrcpu, $savg_syscpu, $savg_idlecpu, $savg_iowaitcpu,
        $savg_device_count, $savg_devutil, $savg_device_in_use,
        $savg_device_light_use, $savg_device_medium_use, $savg_device_heavy_use,
        $savg_device_near_max_use,
##        $savg_navg_tps, $savg_navg_rKB, $savg_navg_wKB, $savg_ntot_tps, $savg_ntot_rKB,
$savg_ntot_wKB);
        $savg_ntot_tps, $savg_ntot_rKB, $savg_ntot_wKB);

## print "          -----
-----\n";
print "          -----\n";
}

sub format_and_print
{
    my $j = shift;
##    printf ("%11s %6.1f %6.2f %6.2f %6.2f %6.2f %6.0f %6.2f %5.1f %5.1f %5.1f
%5.1f %5.1f %8.1f%11.1f%11.1f %8.1f%12.1f%12.1f\n",
    printf ("%11s %6.1f %6.2f %6.2f %6.2f %6.2f %6.0f %6.2f %5.1f %5.1f %5.1f
%5.1f %5.1f %8.1f%12.1f%12.1f\n",
        $array_nodename[$j], $array_blockq[$j], $array_usrcpu[$j], $array_syscpu[$j],
        $array_idlecpu[$j], $array_iowaitcpu[$j],
        $array_device_count[$j], $array_devutil[$j], $array_device_in_use[$j],
        $array_device_light_use[$j], $array_device_medium_use[$j], $array_device_heavy_use[$j],
        $array_device_near_max_use[$j],
##        $array_navg_tps[$j], $array_navg_rKB[$j], $array_navg_wKB[$j], $array_ntot_tps[$j],
        $array_ntot_rKB[$j], $array_ntot_wKB[$j]);
        $array_ntot_tps[$j], $array_ntot_rKB[$j], $array_ntot_wKB[$j]);
}

sub reset_counters
{
    ($tot_blockq, $tot_usrcpu, $tot_syscpu, $tot_idlecpu, $tot_iowaitcpu, $tot_device_count,
    $tot_devutil) = 0;
    ($tot_device_in_use, $tot_device_light_use, $tot_device_medium_use, $tot_device_heavy_use,
    $tot_device_near_max_use) = 0;
##    ($tot_navg_tps, $tot_navg_rKB, $tot_navg_wKB, $tot_ntot_tps, $tot_ntot_rKB, $tot_ntot_wKB) = 0;
    ($tot_ntot_tps, $tot_ntot_rKB, $tot_ntot_wKB) = 0;
    ($savg_blockq, $savg_usrcpu, $savg_syscpu, $savg_idlecpu, $savg_iowaitcpu, $savg_device_count,
    $savg_devutil) = 0;
    ($savg_device_in_use, $savg_device_light_use, $savg_device_medium_use, $savg_device_heavy_use,
    $savg_device_near_max_use) = 0;
##    ($savg_navg_tps, $savg_navg_rKB, $savg_navg_wKB, $savg_ntot_tps, $savg_ntot_rKB, $savg_ntot_wKB)
    = 0;
    ($savg_ntot_tps, $savg_ntot_rKB, $savg_ntot_wKB) = 0;
    ($device_in_use, $device_light_use, $device_medium_use, $device_heavy_use, $device_near_max_use )
    = 0;
    ($tps, $rtps, $wtps, $rKBps, $wKBps) = 0;
    $iodevremainder = '';
    $iodevnewremainder = '';
}

```


Example A-3 shows the global paging and memory resources performance monitoring Perl script `sa_paging_mon.pl`.

Example A-3 sa_paging_mon.pl

```
#!/usr/bin/perl

# Script Name: sa_paging_mon.pl
# Author      : Patrick Thoreson
# Company     : IBM
# Date        : Oct 11th, 2010

use strict;

# Choose which of the following two methods applies
# 1) on Management Node as user 'root' pick the first method
# 2) on Admin Node as DB2 instance owner pick the second method
my @nodes = `lsnode -N BCUALL`;
my @nodes = `lsnode`;
my @nodes = `cat ~/db2nodes.cfg | tr -s ' ' | cut -d' ' -f2 | sort | uniq`;

my $row = $nodes[0];
chomp $row;
my ($nodegroup, $nodelist) = split (/: /,$row);

my $continousloop = 'Y';
my $scriptparm;
my $nbrparms;

$nbrparms = $#ARGV + 1;
if ($nbrparms == 1)
{
    $scriptparm = $ARGV[0];
    chomp $scriptparm;
    if ($scriptparm eq "-s") { $continousloop = 'N' }
}

if ((($nbrparms > 1) || (($nbrparms == 1) && ($scriptparm ne "-s")))
{
    print "Usage is: sa_cpu_mon.pl -s\n";
    print "where the optional parameter -s indicates 'snapshot'\n";
    print "versus default of continous looping.\n";
    exit 1;
}

my $nbrnodes = $nodes + 1;
my @nodeoutputfiles;
my $specific_node_output_file;
my @node_info_array;
my ($n,$m,$p) = 0;
my $nodesleft;
my $firstnodeoutput = 'Y';
my $node_info_row;

my $nodename;
my ($tot_runq, $tot_blockq, $tot_swapin, $tot_swapout, $tot_usrpcpu, $tot_syscpu, $tot_idlecpu);
my ($tot_iowaitcpu, $tot_node_total_mem, $tot_node_used_mem, $tot_node_free_mem);
my ($tot_node_total_swap, $tot_node_used_swap, $tot_node_free_swap);
my ($avg_runq, $avg_blockq, $avg_swapin, $avg_swapout, $avg_usrpcpu, $avg_syscpu, $avg_idlecpu);
my ($avg_iowaitcpu, $avg_node_total_mem, $avg_node_used_mem, $avg_node_free_mem);
my ($avg_node_total_swap, $avg_node_used_swap, $avg_node_free_swap);

my @array_nodename;
my @array_runq;
```

```

my @array_blockq;
my @array_swapin;
my @array_swapout;
my @array_usrcpu;
my @array_syscpu;
my @array_idlecpu;
my @array_iowaitcpu;
my @array_node_total_mem;
my @array_node_used_mem;
my @array_node_free_mem;
my @array_node_total_swap;
my @array_node_used_swap;
my @array_node_free_swap;

do
{
    $n = 0;
    $nodesleft = $nbrnodes;
    $firstnodeoutput = 'Y';
    while ($nodesleft)
    {
        chomp $nodes[$n];
        local *NODEOUT;
        open (NODEOUT, "ssh $nodes[$n] 'echo `hostname`: `vmstat 5 2 | tail -1` `vmstat -s`' & |")
            || die "fork error: $!";
        $nodeoutputfiles[$n] = *NODEOUT;
        $n = $n + 1;
        $nodesleft = $nodesleft - 1;
    }

    reset_counters();

    $m = 0;
    foreach $specific_node_output_file (@nodeoutputfiles)
    {
        while (<$specific_node_output_file>)
        {
            if ("{$firstnodeoutput}" eq "Y")
            {
                header(); $firstnodeoutput = "N"; } $node_info_row = $_ ; extract_info($m); }
#           { header(); $firstnodeoutput = "N"; } $node_info_row = $_ ; print; }
        close $specific_node_output_file || die "child cmd error: $! $?";
        $m = $m + 1;
    }

    compute_and_print_system_summary();

    for ($p = 0; $p < $nbrnodes; $p++)
    {
        format_and_print($p);
    }

} while ($continuousloop eq 'Y');

sub header
{
    if ($continuousloop eq 'Y') { system ("clear"); }
    print "sa_paging_mon    Run    Block    ----- CPU -----    -- Page Swapping --
----- Real Memory -----    ----- Swap Space -----\n";
    print "
Total    Used    Queue    Queue    usr    sys    idle    wio    in    out
# print "
-----    -----    -----    -----    -----    -----    -----    -----    -----
-----\n";
}

sub extract_info

```

```

{
  my $i = shift;
  chomp $node_info_row;

#
# The $na variable is 'not applicable', i.e. we don't need it's value (it's simply a placeholder):
my ($nodename, $runq, $blockq, $na, $na, $na, $na, $swpin, $swpout, $na, $na, $na, $na, $usrcpu,
    $syscpu, $idlecpu, $iowaitcpu, $na, $node_mem_info )
    = split(' ', $node_info_row, 19);
#
    = split(' ', $node_info_row);
my ($node_total_mem, $na, $na, $node_used_mem, $na, $na,
    $na, $na, $na, $na, $na, $na,
    $node_free_mem, $na, $na, $na, $na, $na,
    $na, $na, $na, $node_total_swap, $na, $na,
    $node_used_swap, $na, $na, $node_free_swap, $na)
    = split(' ', $node_mem_info, 29);

$tot_runq           = $tot_runq + $runq;
$tot_blockq        = $tot_blockq + $blockq;
$tot_swapin        = $tot_swapin + $swpin;
$tot_swapout       = $tot_swapout + $swpout;
$tot_usrcpu        = $tot_usrcpu + $usrcpu;
$tot_syscpu        = $tot_syscpu + $syscpu;
$tot_idlecpu       = $tot_idlecpu + $idlecpu;
$tot_iowaitcpu     = $tot_iowaitcpu + $iowaitcpu;
$tot_node_total_mem = $tot_node_total_mem + $node_total_mem;
$tot_node_used_mem  = $tot_node_used_mem + $node_used_mem;
$tot_node_free_mem  = $tot_node_free_mem + $node_free_mem;
$tot_node_total_swap = $tot_node_total_swap + $node_total_swap;
$tot_node_used_swap = $tot_node_used_swap + $node_used_swap;
$tot_node_free_swap = $tot_node_free_swap + $node_free_swap;

$array_nodename[$i] = $nodename;
$array_runq[$i]     = $runq;
$array_blockq[$i]  = $blockq;
$array_swapin[$i]  = $swpin;
$array_swapout[$i] = $swpout;
$array_usrcpu[$i]  = $usrcpu;
$array_syscpu[$i]  = $syscpu;
$array_idlecpu[$i] = $idlecpu;
$array_iowaitcpu[$i] = $iowaitcpu;
$array_node_total_mem[$i] = $node_total_mem;
$array_node_used_mem[$i] = $node_used_mem;
$array_node_free_mem[$i] = $node_free_mem;
$array_node_total_swap[$i] = $node_total_swap;
$array_node_used_swap[$i] = $node_used_swap;
$array_node_free_swap[$i] = $node_free_swap;
}

sub compute_and_print_system_summary
{
  $nodename = "System Avg:";
  $avg_runq = $tot_runq / $nbrnodes;
  $avg_blockq = $tot_blockq / $nbrnodes;
  $avg_swapin = $tot_swapin / $nbrnodes;
  $avg_swapout = $tot_swapout / $nbrnodes;
  $avg_usrcpu = $tot_usrcpu / $nbrnodes;
  $avg_syscpu = $tot_syscpu / $nbrnodes;
  $avg_idlecpu = $tot_idlecpu / $nbrnodes;
  $avg_iowaitcpu = $tot_iowaitcpu / $nbrnodes;
  $avg_node_total_mem = $tot_node_total_mem / $nbrnodes;
  $avg_node_used_mem = $tot_node_used_mem / $nbrnodes;
  $avg_node_free_mem = $tot_node_free_mem / $nbrnodes;
  $avg_node_total_swap = $tot_node_total_swap / $nbrnodes;
  $avg_node_used_swap = $tot_node_used_swap / $nbrnodes;
}

```

```

    $avg_node_free_swap = $tot_node_free_swap / $nbrnodes;
    print "
-----
-----\n";

    printf ("%11s %6.1f %6.1f %5.1f %5.1f %5.1f %5.1f %5d %5d %10d %10d
%10d %10d %10d %10d\n",
           $nodename, $avg_runq, $avg_blockq, $avg_usrcpu, $avg_syscpu, $avg_idlecpu,
           $avg_iowaitcpu, $avg_swapin, $avg_swapout,
           $avg_node_total_mem, $avg_node_used_mem, $avg_node_free_mem,
           $avg_node_total_swap, $avg_node_used_swap, $avg_node_free_swap );

    print "
-----
-----\n";
}

sub format_and_print
{
    my $j = shift;
    printf ("%11s %6.1f %6.1f %5.1f %5.1f %5.1f %5.1f %5d %5d %10d %10d
%10d %10d %10d %10d\n",
           $array_nodename[$j], $array_runq[$j], $array_blockq[$j], $array_usrcpu[$j],
           $array_syscpu[$j],
           $array_idlecpu[$j], $array_iowaitcpu[$j], $array_swapin[$j], $array_swapout[$j],
           $array_node_total_mem[$j], $array_node_used_mem[$j], $array_node_free_mem[$j],
           $array_node_total_swap[$j], $array_node_used_swap[$j], $array_node_free_swap[$j]);
}

sub reset_counters
{
    ($tot_runq, $tot_blockq, $tot_swapin, $tot_swapout, $tot_usrcpu, $tot_syscpu, $tot_idlecpu) = 0;;
    ($tot_iowaitcpu, $tot_node_total_mem, $tot_node_used_mem, $tot_node_free_mem) = 0;
    ($tot_node_total_swap, $tot_node_used_swap, $tot_node_free_swap) = 0;
    ($avg_runq, $avg_blockq, $avg_swapin, $avg_swapout, $avg_usrcpu, $avg_syscpu, $avg_idlecpu) = 0;;
    ($avg_iowaitcpu, $avg_node_total_mem, $avg_node_used_mem, $avg_node_free_mem) = 0;
    ($avg_node_total_swap, $avg_node_used_swap, $avg_node_free_swap) = 0;
}

```

Example A-4 shows the disk device to file system mapping Korn shell script `disk2fs.ksh`.

Example A-4 `disk2fs.ksh`

```

#!/bin/ksh

# Script Name: disk2fs.ksh
# Author      : Patrick Thoreson
# Company     : IBM
# Date       : Sep 24th, 2010

if [ $# != 1 ]
then
    echo "Usage is: $0 device"
    exit 1
fi

device=${1};export device

###echo "DEBUG Device: >${device}<"

if [ ! -e /dev/${device} ]

```

```

then
    echo "Device ${device} does not exist."
    exit 2
fi

device_long_desc=`ls -l /dev/${device}`;export device_long_desc
device_type=`echo ${device_long_desc} | cut -d' ' -f4`;export device_type

if [ "${device_type}" != "disk" ]
then
    echo "Device is not a disk: >${device_type}<"
    exit 3
fi

###echo "DEBUG Device type: >${device_type}<"

device_major_nbr=`echo ${device_long_desc} | cut -d' ' -f5|cut -d',' -f1`;export device_major_nbr
###echo "DEBUG Device Major number: >${device_major_nbr}<"

device_minor_nbr=`echo ${device_long_desc} | cut -d' ' -f6`;export device_minor_nbr
###echo "DEBUG Device Minor number: >${device_minor_nbr}<"

lvs -o lv_name,vg_name,lv_kernel_major,lv_kernel_minor,devices --separator : > /tmp/lvs.txt 2>
/dev/null

lvsinfo='';export lvsinfo
lvsinfo=`grep ':'${device_major_nbr}':'${device_minor_nbr}':' /tmp/lvs.txt | cut -c3-`
if [ "${lvsinfo}x" = "x" ]
then
    lvsinfo=`grep '/dev/'${device}'(' /tmp/lvs.txt | cut -c3-`
    if [ "${lvsinfo}x" = "x" ]
    then
        echo "lvsinfo not found."
        exit 4
    else
        ### echo "DEBUG lvsinfo: >${lvsinfo}<"
    fi
else
    ### echo "DEBUG lvsinfo: >${lvsinfo}<"
fi

lvname=`echo ${lvsinfo}| cut -d: -f1`;export lvname
###echo "DEBUG lvname: >${lvname}<"
vgname=`echo ${lvsinfo}| cut -d: -f2`;export vgname
###echo "DEBUG vgname: >${vgname}<"
lvdevice='/dev/'${vgname}'/'${lvname};export lvdevice
###echo "DEBUG lvdevice: >${lvdevice}<"

fstabinfo='';export fstabinfo
fstabinfo=`grep ${lvdevice} /etc/fstab`
###echo "DEBUG fstab info: >${fstabinfo}<"

fsmountdir=`echo ${fstabinfo} | cut -d' ' -f2`;export fsmountdir
echo `hostname`: I/O device ${device} --> filesystem mountdir: ${fsmountdir} (LV: ${lvdevice})"

```

Example A-5 shows the file system to disk device mapping Korn shell script `fs2disk.ksh`.

Example A-5 fs2disk.ksh

```
#!/bin/ksh

# Script Name: fs2disk.ksh
# Author      : Patrick Thoreson
# Company     : IBM
# Date       : Sep 23th, 2010

if [ $# != 1 ]
then
    echo "Usage is: $0 <filesystem mount directory>"
    echo "Ex: $0 /stage2"
    exit 1
fi

fsmountdir=${1};export fsmountdir

#echo "DEBUG fsmountdir: >${fsmountdir}<"

if [ ! -e ${fsmountdir} ]
then
    echo "filesystem mount directory ${fsmountdir} does not exist."
    exit 2
fi

fstabinfo='';export fstabinfo
tmpfsdir='';export tmpfsdir
lvdevice='';export lvdevice
cat /etc/fstab | while read fstabinfo
do
    tmpfsdir=`echo ${fstabinfo} | tr -s ' ' | cut -d' ' -f2`
    # if [ "${tmpfsdir}" = "${fsmountdir}" ]
    # if [ "${tmpfsdir}" = "${fsmountdir}" -o "${tmpfsdir}" = "${fsmountdir}/" ]
    then
        lvdevice=`echo ${fstabinfo} | tr -s ' ' | cut -d' ' -f1`
        break
    fi
done

if [ "${tmpfsdir}" != "${fsmountdir}" -a "${tmpfsdir}" != "${fsmountdir}/" ]
then
    echo "File system ${fsmountdir} not found in /etc/fstab."
    exit 1
fi

#echo "DEBUG lvdevice: >${lvdevice}<"

lvname=`echo ${lvdevice}| cut -d'/' -f4`;export lvname
#echo "DEBUG lvname: >${lvname}<"
vgname=`echo ${lvdevice}| cut -d'/' -f3`;export vgname
#echo "DEBUG vgname: >${vgname}<"

lvs --noheadings -o lv_name,vg_name,lv_kernel_major,lv_kernel_minor,devices --separator :
${lvdevice} > /tmp/lvs.txt 2> /dev/null

lvsinfo='';export lvsinfo
device_major_nbr='';export device_major_nbr
device_minor_nbr='';export device_minor_nbr
device='';export device
majorminor='';export majorminor
otherdeviceinfo='';export otherdeviceinfo
```

```

otherdevicelist='';export otherdevicelist
cat /tmp/lvs.txt | cut -c3- | while read lvsinfo
do
    device_major_nbr=`echo ${lvsinfo} | cut -d: -f3`
    # echo "DEBUG Device Major number: >${device_major_nbr}<"
    device_minor_nbr=`echo ${lvsinfo} | cut -d: -f4`
    # echo "DEBUG Device Minor number: >${device_minor_nbr}<"
    majorminor=${device_major_nbr}', '${device_minor_nbr}
    # echo "DEBUG majorminor : >${majorminor}<"
    device=`ls -ld /dev/* | tr -s ' ' | grep ' disk ' | grep " ${majorminor} " | cut -d/' -f3`
    # echo "DEBUG Device : >${device}<"
    otherdeviceinfo=`echo ${lvsinfo} | cut -d: -f5`
    # echo "DEBUG Device info: >${otherdeviceinfo}<"
    otherdevicelist=`echo ${otherdeviceinfo} | sed '1,$s/\/dev/\/g' | sed '1,$s/([0-9])/g'`
    # echo "DEBUG Device list: >${otherdevicelist}<"
    echo `hostname`: filesystem mountdir ${fsmountdir} (LV: ${lvdevice}) ==> I/O device ${device},
other device(s) ${otherdevicelist}."
done

```



B

Scripts for DB2 workload manager configuration

In this appendix we provide the scripts used in 7.2, “DB2 workload manager” on page 243, which show how to configure a DB2 workload manager for an IBM Smart Analytics System.

B.1 Creating MARTS tables

This section describes how to create the tables used to test the DB2 workload manager work action set.

For our workload management scripts, we modify the DB2 provided scripts under the *<DB2 home directory>/samples/data* to create and populate four tables under the MARTS schema:

- ▶ Fact table: PRCHS_PRFL_ANALYSIS
- ▶ Dimension tables: STORE, TIME, and PRODUCT

Example B-1 shows the modified script of createMartTables.sql to create the tables. We also change the table space definition from USERSPACE1 to TS_SMALL, the table space for non-partitioned tables. Do not run RUNSTATS on these tables after populating data, otherwise, the timeron cost will be much lower and the work action set will not redirect the queries to the intended service subclasses during the exercises.

Example B-1 Script MARTS_create_tables.sql

```
--
-- MARTS_create_tables.sql
--
-- This script creates the sample tables used to optionally test the
-- Work Action Set timeron ranges
--

DROP TABLE MARTS.TIME;
DROP TABLE MARTS.STORE;
DROP TABLE MARTS.PRCHS_PRFL_ANALYSIS;
DROP TABLE MARTS.PRODUCT;

DROP SCHEMA MARTS RESTRICT;
CREATE SCHEMA MARTS;

CREATE TABLE MARTS.TIME (
    TIME_ID          SMALLINT NOT NULL,
    UNQ_ID_SRC_STM  CHAR(20),
    TIME_TP_ID      SMALLINT NOT NULL,
    CDR_YR          SMALLINT,
    CDR_QTR         SMALLINT,
    CDR_MO          SMALLINT,
    DAY_OF_CDR_YR  SMALLINT,
    DAY_CDR_QTR    SMALLINT,
    DAY_CDR_MO     SMALLINT,
    FSC_YR         SMALLINT,
    FSC_QTR        SMALLINT,
    FSC_MO         SMALLINT,
    NBR_DYS        SMALLINT,
    NBR_BSN_DYS    SMALLINT,
```

```

        PBLC_HOL_F      SMALLINT,
        BSN_DAY_F      SMALLINT,
        LAST_BSN_DAY_MO_F SMALLINT,
        SSON_ID        SMALLINT,
        MONTH_LABEL    VARCHAR(20) ,
        QTR_LABEL      VARCHAR(10)
    )
    IN TS_SMALL;

CREATE TABLE MARTS.STORE (
    STR_IP_ID INTEGER NOT NULL,
    STR_TP_NM VARCHAR(64) NOT NULL,
    ORG_IP_ID INTEGER NOT NULL,
    PRN_OU_IP_ID INTEGER,
    MGR_EMPE_ID INTEGER,
    NR_CPTR_PRX_NM VARCHAR(64),
    SALE_VOL_RNG_NM VARCHAR(64),
    FLRSP_AREA_RNG_NM VARCHAR(64),
    STR_CODE CHAR(6) NOT NULL,
    STR_SUB_DIV_NM VARCHAR(64) NOT NULL,
    STR_REG_NM VARCHAR(64) NOT NULL,
    STR_DIS_NM VARCHAR(64) NOT NULL,
    STR_NM VARCHAR(64) NOT NULL
)
    IN TS_SMALL;

CREATE TABLE MARTS.PRCHS_PRFL_ANALYSIS (
    STR_IP_ID INTEGER NOT NULL,
    PD_ID INTEGER NOT NULL,
    TIME_ID SMALLINT NOT NULL,
    NMBR_OF_MRKT_BSKTS INTEGER,
    NUMBER_OF_ITEMS INTEGER,
    PRDCT_BK_PRC_AMUNT DECIMAL(14,2),
    CST_OF_GDS_SLD_CGS DECIMAL(14,2),
    SALES_AMOUNT DECIMAL(14,2)
)
    IN TS_SMALL;

CREATE TABLE MARTS.PRODUCT (
    PD_ID INTEGER NOT NULL,
    UNQ_ID_SRC_STM CHAR(20),
    PD_TP_NM VARCHAR(64) NOT NULL,
    BASE_PD_ID INTEGER,
    NM VARCHAR(64),
    PD_IDENT CHAR(25),
    DSC VARCHAR(256),
    PD_DEPT_NM VARCHAR(64) NOT NULL,
    PD_SUB_DEPT_NM VARCHAR(64) NOT NULL,
    PD_CL_NM VARCHAR(64) NOT NULL,
    PD_SUB_CL_NM VARCHAR(64) NOT NULL
)
    IN TS_SMALL;

```

To load data into the MARTS tables, use the MARTS_load_tables.sql script shown in Example B-2.

Example B-2 Script MARTS_load_tables.sql

```
--
-- MARTS_load_tables.sql
--
-- This script loads data into the 4 MARTS tables used to help
--   in configuring the WLM.
--

LOAD from MartPrchProfAnalysis.txt of del REPLACE into MARTS.PRCHS_PRFL_ANALYSIS ;
LOAD from MartPD.txt                of del REPLACE into MARTS.PRODUCT ;
LOAD from MartStore.txt              of del REPLACE into MARTS.STORE ;
LOAD from MartTime.txt               of del REPLACE into MARTS.TIME ;

select 'PRCHS_PRFL_ANALYSIS' , count(*) from MARTS.PRCHS_PRFL_ANALYSIS UNION
select 'PRODUCT' , count(*) from MARTS.PRODUCT UNION
select 'STORE' , count(*) from MARTS.STORE UNION
select 'TIME' , count(*) from MARTS.TIME;
```

Use MARTS_count_tables.sql shown in Example B-3 to count the rows of the MARTS tables.

Example B-3 Script MARTS_count_tables.sql

```
--
-- MARTS_count_tables.sql
--
-- This script count the rows in all MARTS tables
--

select 'PRCHS_PRFL_ANALYSIS' , count(*) from MARTS.PRCHS_PRFL_ANALYSIS UNION
select 'PRODUCT' , count(*) from MARTS.PRODUCT UNION
select 'STORE' , count(*) from MARTS.STORE UNION
select 'TIME' , count(*) from MARTS.TIME;
```

To drop the MART tables, use the script shown in Example B-4.

Example B-4 Script MARTS_drop_tables.sql

```
--
-- MART_drop_tables.sql
--
-- This script creates the sample tables used to optionally test the
--   Work Action Set timeron ranges
--

DROP TABLE MARTS.TIME;
DROP TABLE MARTS.STORE;
DROP TABLE MARTS.PRCHS_PRFL_ANALYSIS;
```

```
DROP TABLE MARTS.PRODUCT;

DROP SCHEMA MARTS RESTRICT;
```

B.2 Untuned DB2 workload manager configuration

These scripts are use in the untuned DB2 workload manager environment exercise.

Example B-5 shows the script to create services classes.

Example B-5 01_create_svc_classes.sql

```
--
-- Script 01_create_svc_classes.sql
--
-- This script creates:
--     service superclass MAIN
--     service subclasses ETL, Trivial, Minor, Simple, Medium and Complex
--
--
-- To delete a service superclass you need to drop every dependent object:
--     remap the SYSDEFAULTUSERWORKLOAD back to SYSDEFAULTUSERCLASS
-- (if applicable)
--
--     disable the service subclasses
--     drop work action sets
--     drop work class sets
--     drop service classes' thresholds
--     drop service subclasses
--     drop service superclass
--
CREATE SERVICE CLASS MAIN ;
commit;

CREATE SERVICE CLASS    ETL    under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
commit;

CREATE SERVICE CLASS    Trivial under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
commit;

CREATE SERVICE CLASS    Minor under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
commit;

CREATE SERVICE CLASS    Simple under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
commit;

CREATE SERVICE CLASS    Medium under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
commit;

CREATE SERVICE CLASS    Complex under MAIN COLLECT AGGREGATE ACTIVITY DATA EXTENDED;
```

```

commit;

-- Verify existing super and sub service classes

select
    varchar(serviceclassname,30)      as SvcClass_name,
    varchar(parentserviceclassname,30) as Parent_Class_name
from
    syscat.serviceclasses
where parentserviceclassname = 'MAIN' ;

```

Example B-6 shows the script to remap the DEFAULTUSERWORKLOAD out from SYSDEFAULTUSERCLASS and into the MAIN superclass.

Example B-6 02_remap_dft_wkl.sql.

```

--
-- Script 02_remap_dft_wkl.sql
--
-- This script will remap the DEFAULTUSERWORKLOAD out from SYSDEFAULTUSERCLASS
-- and into the newly created MAIN superclass
--

echo -;
echo ----- Original defaultUSERworkload mapping ----- ;

select
    varchar(workloadname,25)          as Workload_name,
    varchar(serviceclassname,20)      as SvcClass_name,
    varchar(parentserviceclassname,20) as Parent_Class_name,
    EvaluationOrder                   as Eval_Order
FROM
    syscat.workloads
ORDER
    by 4;

alter workload SYSDEFAULTUSERWORKLOAD
    SERVICE CLASS MAIN ;
commit;

echo ----- Remapped defaultUSERworkload ----- ;

select
    varchar(workloadname,25)          as Workload_name,
    varchar(serviceclassname,20)      as SvcClass_name,
    varchar(parentserviceclassname,20) as Parent_Class_name,
    EvaluationOrder                   as Eval_Order
FROM
    syscat.workloads
ORDER
    by 4;

```

Example B-7 shows the script to create new work class sets and work action sets.

Example B-7 03_create_wk_action_set.sq

```
--
-- Script 03_create_wk_action_set.sql
--
-- This script creates the WORK_CLASS_SET and the WORK_ACTION_SET
-- with the starting values for the services subclasses
-- as described earlier, in "Untuned DB2 workload manager environment"
--

CREATE WORK CLASS SET "WORK_CLASS_SET_1"
(
  WORK CLASS "WCLASS_TRIVIAL" WORK TYPE DML FOR TIMERONCOST FROM 0    to 5000POSITION AT
  1,
  WORK CLASS "WCLASS_MINOR" WORK TYPE DML FOR TIMERONCOST FROM 5000    to 30000POSITION
  AT 2,
  WORK CLASS "WCLASS_SIMPLE" WORK TYPE DML FOR TIMERONCOST FROM 30000    to 300000POSITION
  AT 3,
  WORK CLASS "WCLASS_MEDIUM" WORK TYPE DML FOR TIMERONCOST FROM 300000    to 5000000
  POSITION AT 4,
  WORK CLASS "WCLASS_COMPLEX" WORK TYPE DML FOR TIMERONCOST FROM 5000000 to UNBOUNDED
  POSITION AT 5,
  WORK CLASS "WCLASS_ETL" WORK TYPE LOAD POSITION AT 6,
  WORK CLASS "WCLASS_OTHER" WORK TYPE ALL POSITION AT 7
) ;

commit ;

echo ===== ;
echo ===== SYSCAT.WORKCLASSETS table contents ===== ;
SELECT varchar(workclasssetname,40) as Work_Class_Set_name from SYSCAT.WORKCLASSETS ;

echo ===== ;
echo ===== SYSCAT.WORKCLASSES table contents ===== ;
SELECT varchar(workclassname,20) as Work_Class_name, varchar(workclasssetname,20) as
Work_Class_Set_name, int(fromvalue) as From_value, int(tovalue) as To_value,
evaluationorder as Eval_order from SYSCAT.WORKCLASSES order by evaluationorder ;

CREATE WORK ACTION SET "WORK_ACTION_SET_1" FOR SERVICE CLASS "MAIN" USING WORK CLASS SET
"WORK_CLASS_SET_1"
(
  WORK ACTION "WACTION_TRIVIAL" ON WORK CLASS "WCLASS_TRIVIAL" MAP ACTIVITY WITHOUT
  NESTED TO "TRIVIAL",
  WORK ACTION "WACTION_MINOR" ON WORK CLASS "WCLASS_MINOR" MAP ACTIVITY WITHOUT
  NESTED TO "MINOR",
  WORK ACTION "WACTION_SIMPLE" ON WORK CLASS "WCLASS_SIMPLE" MAP ACTIVITY WITHOUT
  NESTED TO "SIMPLE" ,
  WORK ACTION "WACTION_MEDIUM" ON WORK CLASS "WCLASS_MEDIUM" MAP ACTIVITY WITHOUT
  NESTED TO "MEDIUM" ,
```

```

WORK ACTION "WACTION_COMPLEX" ON WORK CLASS "WCLASS_COMPLEX" MAP ACTIVITY WITHOUT
NESTED TO "COMPLEX",
WORK ACTION "WACTION_ETL"      ON WORK CLASS "WCLASS_ETL"      MAP ACTIVITY WITHOUT
NESTED TO "ETL"
) ;

commit;

echo ===== ;
echo ===== SYSCAT.WORKACTIONSETS table contents ===== ;
SELECT varchar(actionsetname,30) as Work_Action_Set_name, varchar(objectname,30) as
Object_name from SYSCAT.WORKACTIONSETS ;

echo ===== ;
echo ===== SYSCAT.WORKACTIONS table contents ===== ;
SELECT varchar(actionname,25) as Work_Action_name, varchar(actionsetname,25) as
Work_Action_Set_name, varchar(workclassname,25) as Work_Class_name from
SYSCAT.WORKACTIONS ;

```

Example B-8 shows the script to create DB2 workload manager table space.

Example B-8 04_create_wlm_tablespace.sql

```

--
-- Script 04_create_wlm_tablespace.sql
--
-- This script creates the table pace for the WLM tables over
-- all DB2 database partitions.
-- WLM data gathered for DB database partitions whose tablespace/WLM control tables
-- are nonexistent will be discarded!

CREATE TABLESPACE TS_WLM_MON MAXSIZE 2G;

commit;

```

Example B-9 shows the script **05_wlmevmon.ddl** to create event monitors.

Example B-9 05_wlmevmon.ddl

```

--
-- Script 05_wlmevmon.ddl
--
-- -*- sql -*-
--
-- Sample DDL to create three workload management
-- event monitors.
--
-- -> assumes db2start issued
-- -> assumes connection to a database exists
-- -> assumes called by "db2 -tf wlmevmon.ddl"
-- -> Other notes:
--     - All target tables will be created in the table space named

```



```

--      TS_WLM_MON.  Change this if necessary.
--      - Any specified table spaces must exist prior to executing this DDL.
--      Furthermore they should reside across all partitions; otherwise
--      monitoring information may be lost.  Also, make sure they have space
--      to contain data from the event monitors.
--      - If the target table spaces are DMS table spaces, the PCTDEACTIVATE parameter
--      specifies how full the table space must be before the event monitor
--      automatically deactivates.  Change the value if necessary.  When the
--      target table space has auto-resize enabled, set PCTDEACTIVATE to 100.
--      Remove PCTDEACTIVATE for any specified System Managed (SMS) table
--      spaces.
--      - If AUTOSTART is specified, the event monitor will automatically
--      activate when the database activates.  If MANUALSTART is specified
--      instead, the event monitor must be explicitly activated through
--      a SET EVENT MONITOR statement after the database is activated.
--
--
-- To remind users how to use this file!
--
ECHO                                     ;
ECHO ***** IMPORTANT *****          ;
ECHO                                     ;
ECHO USAGE: db2 -tf wlmevmon.ddl       ;
ECHO                                     ;
ECHO ***** IMPORTANT *****          ;
ECHO                                     ;
ECHO                                     ;

--
--
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;

--
-- Define the activity event monitor named DB2ACTIVITIES
--
CREATE EVENT MONITOR DB2ACTIVITIES
FOR ACTIVITIES
WRITE TO TABLE
ACTIVITY (TABLE ACTIVITY_DB2ACTIVITIES
          IN TS_WLM_MON
          PCTDEACTIVATE 100),
ACTIVITYSTMT (TABLE ACTIVITYSTMT_DB2ACTIVITIES
              IN TS_WLM_MON
              PCTDEACTIVATE 100),
ACTIVITYVALS (TABLE ACTIVITYVALS_DB2ACTIVITIES
               IN TS_WLM_MON
               PCTDEACTIVATE 100),
CONTROL (TABLE CONTROL_DB2ACTIVITIES
          IN TS_WLM_MON
          PCTDEACTIVATE 100)
AUTOSTART;

```

```

--
-- Define the statistics event monitor named DB2STATISTICS
--
CREATE EVENT MONITOR DB2STATISTICS
FOR STATISTICS
WRITE TO TABLE
SCSTATS (TABLE SCSTATS_DB2STATISTICS
         IN TS_WLM_MON
         PCTDEACTIVATE 100),
WCSTATS (TABLE WCSTATS_DB2STATISTICS
         IN TS_WLM_MON
         PCTDEACTIVATE 100),
WLSTATS (TABLE WLSTATS_DB2STATISTICS
         IN TS_WLM_MON
         PCTDEACTIVATE 100),
QSTATS (TABLE QSTATS_DB2STATISTICS
        IN TS_WLM_MON
        PCTDEACTIVATE 100),
HISTOGRAMBIN (TABLE HISTOGRAMBIN_DB2STATISTICS
              IN TS_WLM_MON
              PCTDEACTIVATE 100),
CONTROL (TABLE CONTROL_DB2STATISTICS
         IN TS_WLM_MON
         PCTDEACTIVATE 100)
AUTOSTART;

--
-- Define the threshold violation event monitor named DB2THRESHOLDVIOLATIONS
--
CREATE EVENT MONITOR DB2THRESHOLDVIOLATIONS
FOR THRESHOLD VIOLATIONS
WRITE TO TABLE
THRESHOLDVIOLATIONS (TABLE THRESHOLDVIOLATIONS_DB2THRESHOLDVIOLATIONS
                        IN TS_WLM_MON
                        PCTDEACTIVATE 100),
CONTROL (TABLE CONTROL_DB2THRESHOLDVIOLATIONS
         IN TS_WLM_MON
         PCTDEACTIVATE 100)
AUTOSTART;

--
-- Commit work
--
COMMIT WORK;

```

Example B-10 shows the script to activate event monitors.

Example B-10 06_start_evt_monitors.sql

```

--
-- Script 06_start_evt_monitors.sql
--

```

```

-- This script turns WLM monitors on
--

echo .;
echo ----- Monitor switches status ----- ;

SELECT substr(evmonname,1,30) as evmonname,
CASE
  WHEN event_mon_state(evmonname) = 0 THEN 'Inactive'
  WHEN event_mon_state(evmonname) = 1 THEN 'Active'
END as STATUS
FROM syscat.eventmonitors ;

set event monitor db2activities state 1 ;

set event monitor db2statistics state 1 ;

set event monitor db2thresholdviolations state 1 ;

echo ----- Monitor switches status ----- ;

SELECT substr(evmonname,1,30) as evmonname,
CASE
  WHEN event_mon_state(evmonname) = 0 THEN 'Inactive'
  WHEN event_mon_state(evmonname) = 1 THEN 'Active'
END as STATUS
FROM syscat.eventmonitors ;

```

Example B-11 shows the script to test the work action set setting.

Example B-11 07_execs_by_subclasses.sql

```

--
-- Script 07_execs_by_subclasses.sql
--
-- This script will display existing superclasses and subclasses,
-- and will execute some queries.
-- These queries have increasing timeron cost, so the Work Ation Set
-- This will send each of them to a particular service class.
--

echo = ;

echo ===== Workloads executed by Subclasses ===== ;

SELECT
  VARCHAR( SERVICE_SUPERCLASS_NAME, 20) SUPERCLASS,
  VARCHAR( SERVICE_SUBCLASS_NAME, 20) SUBCLASS,
  COORD_ACT_COMPLETED_TOTAL
FROM

```

```

        TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97('','",-1)) AS T
WHERE
    SERVICE_SUPERCLASS_NAME like 'MAIN%'
;

echo executing queries... ;
echo .;

echo ===== query to be mapped to the TRIVIAL service subclass ===== ;
select count(*) from MARTS.PRODUCT;

echo ===== query to be mapped to the MINOR service subclass ===== ;
select count(*) from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME;

echo ===== query to be mapped to the EASY service subclass ===== ;
select count(*) from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME, MARTS.STORE;

echo ===== query to be mapped to the MEDIUM service subclass ===== ;
select count_big(*) from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.PRCHS_PRFL_ANALYSIS;

echo ===== query to be mapped to the COMPLEX service subclass ===== ;
-- select count_big(*) from MARTS.PRODUCT, MARTS.Time, MARTS.Time;

echo ===== Workloads executed by Subclasses ===== ;

SELECT
    VARCHAR( SERVICE_SUPERCLASS_NAME, 20) SUPERCLASS,
    VARCHAR( SERVICE_SUBCLASS_NAME, 20) SUBCLASS,
    COORD_ACT_COMPLETED_TOTAL
FROM
    TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97('','",-1)) AS T
WHERE
    SERVICE_SUPERCLASS_NAME like 'MAIN%' ;

```

Example B-12 shows the script for verifying the ETL service class.

Example B-12 08_etl_subclass.sql

```

--
-- Script 08_etl_subclass.sql
--
-- This script creates a table and load data into it
--

create table db2admin.PRODUCT like marts.product;
declare mycursor cursor for select * from marts.product ;
load from mycursor of cursor replace into db2admin.product ;
drop table db2admin.product ;

echo = ;
echo ===== Executed workloads status ===== ;

```

```

SELECT
    VARCHAR( SERVICE_SUPERCLASS_NAME, 30) SUPERCLASS,
    VARCHAR( SERVICE_SUBCLASS_NAME, 20) SUBCLASS,
    COORD_ACT_COMPLETED_TOTAL
FROM
    TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(' ','',-1)) AS T
WHERE
    SERVICE_SUPERCLASS_NAME like 'MAIN%' ;

```

Example B-13 and Example B-14 show the queries for verifying the concurrency workloads in an UNIX environment.

Example B-13 query_minor.sql (for UNIX)

```

select count(*) as Minor from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME ;

```

Example B-14 query_easy.sql (for UNIX)

```

select count(*) as Easy from MARTS.PRCHS_PRFL_ANALYSIS, MARTS.TIME, MARTS.STORE ;

```

Example B-15 shows the script for running the queries for verifying the concurrency workloads on an UNIX environment. Replace db_name with your database name.

Example B-15 09_conc_exec_Unix.sh

```

db2batch -d db_name -f query_minor.sql -a db2admin/ibm2blue -time off &
db2batch -d db_name -f query_minor.sql -a db2admin/ibm2blue -time off &
db2batch -d db_name -f query_minor.sql -a db2admin/ibm2blue -time off &
db2batch -d db_name -f query_minor.sql -a db2admin/ibm2blue -time off &

db2batch -d db_name -f query_easy.sql -a db2admin/ibm2blue -time off &
db2batch -d db_name -f query_easy.sql -a db2admin/ibm2blue -time off &
db2batch -d db_name -f query_easy.sql -a db2admin/ibm2blue -time off &

```

Example B-16 shows the script for checking concurrency on an UNIX environment.

Example B-16 10_conc_check.sql

```

echo = ;
echo ===== Highest number of concurrent workload occurrences ===== ;
echo ===== (since last reset) ===== ;

SELECT CONCURRENT_WLO_TOP,
        SUBSTR (WORKLOAD_NAME,1,25) AS WORKLOAD_NAME
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97(' ','',-1)) AS T
WHERE DBPARTITIONNUM = 0
ORDER BY WORKLOAD_NAME ;

```

```

echo ===== Workloads executed by Subclasses ===== ;

SELECT
    VARCHAR( SERVICE_SUPERCLASS_NAME, 27) SUPERCLASS,
    VARCHAR( SERVICE_SUBCLASS_NAME, 18) SUBCLASS,
    COORD_ACT_COMPLETED_TOTAL as NUMBER_EXECS,
    CONCURRENT_ACT_TOP as CONC_HWM
FROM
    TABLE(WLM_GET_SERVICE_SUBCLASS_STATS_V97('','",-1)) AS T

--WHERE
--    SERVICE_SUPERCLASS_NAME like 'MAIN%'
;

```

Example B-17 and Example B-18 show the queries for verifying the concurrency workloads in a Windows environment.

Example B-17 query_medium.txt (for Windows)

```

connect to sample2 USER user4 USING password;
set schema schema_name ;
select count(*) as medium from empmdc, empmdc ;

```

Example B-18 query_easy.txt (for Windows)

```

connect to sample2 USER USER4 USING password;
set schema schema_name ;
Select count(*) as easy from empmdc, staff, staff ;

```

Example B-19 shows the **09a_conc_exec_Win.bat** script to run the queries for the concurrency test on a Windows environment. Use the same script to see the results.

Example B-19 Script 09a_conc_exec_Win.bat

```

REM 09a_conc_exec_Win.bat
REM Starts 4 concurrent medium workloads
db2cmd -c db2 -tf query_medium.txt
db2cmd -c db2 -tf query_medium.txt
db2cmd -c db2 -tf query_medium.txt
db2cmd -c db2 -tf query_medium.txt

REM Starts 3 concurrent easy workloads
db2cmd -c db2 -tf query_easy.txt
db2cmd -c db2 -tf query_easy.txt
db2cmd -c db2 -tf query_easy.txt

```

Example B-20 shows the commands to create timeout thresholds for service subclasses.

Example B-20 Script 11_create_timeout_thresholds

```
--
-- Script 11_create_timeout_threshold
--
-- This script creates elapsed time thresholds for service subclasses
--

-- Create threshold for TRIVIAL subclass -----
--
CREATE THRESHOLD TH_TIME_SC_TRIVIAL FOR SERVICE CLASS TRIVIAL UNDER MAIN ACTIVITIES
ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 1 MINUTE
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS CONTINUE ;

-- Create threshold for MINOR subclass -----
--
CREATE THRESHOLD TH_TIME_SC_MINOR FOR SERVICE CLASS MINOR UNDER MAIN ACTIVITIES
ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 5 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS CONTINUE ;

-- Create threshold for SIMPLE subclass -----
--
CREATE THRESHOLD TH_TIME_SC_SIMPLE FOR SERVICE CLASS SIMPLE UNDER MAIN ACTIVITIES
ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 30 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS CONTINUE ;

-- Create threshold for MEDIUM subclass -----
--
CREATE THRESHOLD TH_TIME_SC_MEDIUM FOR SERVICE CLASS MEDIUM UNDER MAIN ACTIVITIES
ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 60 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS CONTINUE ;

-- Create threshold for COMPLEX subclass -----
--
-- elapsed time:
CREATE THRESHOLD TH_TIME_SC_COMPLEX FOR SERVICE CLASS COMPLEX UNDER MAIN ACTIVITIES
ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 240 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS CONTINUE ;
```

Example B-21 is an example of how to check concurrency in the event monitor tables.

Example B-21 Script 12_subclass_concurrency.sql

```
--
-- Script 12_subclass_concurrency.sql
--
-- This script queries the wlm statistic tables to show the
-- number of cocurrent query execution by subclass and by time
--
-- Change the timestamps below to the desired period
--

SELECT
    concurrent_act_top,
    varchar(service_subclass_name,20) as subclass,
    varchar(service_superclass_name,30) as superclass,
    statistics_timestamp
FROM
    scstats_db2statistics
WHERE
    date(statistics_timestamp) = current date
--    statistics_timestamp between
--    '2010-11-15-15.00.00' and
--    '2010-11-15-15.30.00'
;
```

In Example B-22, **13_alter_default_workload** is the script to start collecting statistics in the default workload.

Example B-22 Script 13_alter_default_workload

```
--
-- Script 13_alter_default_workload.sql
--
-- This script starts collecting default workload statistics
--

alter workload sysdefaultuserworkload collect activity data on coordinator with details
;
```

Example B-23 shows the script to obtain data stored in the statistics tables.

Example B-23 14_dftwkload_statements.sql

```
--
-- Script 14_dftwkload_statements.sql
--
-- This script selects the statements captured at the default workload
```



```

-- (e.workload_id = 1, which is the defaultuserworkload)
-- along with some other details, like: user, application, date, time,
-- superclass and subclass for the current day.
--
SELECTvarchar(session_auth_id,15) as user_name,
      varchar(appl_name,10) as appl_name,
      varchar(workloadname,25) as workload_name,
      varchar(service_superclass_name,10) as superclass,
      varchar(service_subclass_name,10) as subclass,
      date(time_started) as date,
      time(time_started) as time,
      varchar(stmt_text, 150) as statement_text
FROM  ACTIVITYSTMT_DB2ACTIVITIES s, ACTIVITY_DB2ACTIVITIES e, syscat.workloads w
WHEREs.activity_id = e.activity_id
AND s.appl_id = e.appl_id
AND s.uow_id = e.uow_id
AND e.workload_id = 1
AND e.workload_id = w.workloadid
----- uncomment next row to obtain queries captured today
AND date(e.time_started) = date (current timestamp)
----- or adjust date and uncomment next row to obtain queries captured at selected day
-- and date(e.time_started) = date ('11/02/2010')
FETCH first 50 rows only ;

```

B.3 Tuned DB2 workload manager configuration

These scripts are used for the tuned DB2 workload manager environment exercise.

Example B-24 shows the script to create DB2 roles.

Example B-24 Script 50_create_roles.sql

```

--
-- Script 50_create_roles.sql
--
-- This script create DB2 roles.
-- The idea is to create groups of similar users into one of the roles
--

CREATE ROLE Adhoc ;
GRANT  ROLE Adhoc TO USER user1 ;
GRANT  ROLE Adhoc TO USER user2 ;
GRANT  ROLE Adhoc TO USER user3 ;
commit;

CREATE ROLE DBAs ;
GRANT  ROLE DBAs TO USER user4 ;
GRANT  ROLE DBAs TO USER user5 ;
commit;

```

```

CREATE ROLE PWRUSR ;
GRANT  ROLE DBAs TO USER user6 ;
GRANT  ROLE DBAs TO USER user7 ;
commit;

CREATE ROLE GUEST ;
GRANT  ROLE DBAs TO USER user8 ;
GRANT  ROLE DBAs TO USER user9 ;
commit;

```

Example B-25 shows the script to create DB2 workload manager workload objects.

Example B-25 51_create_workloads.sql

```

--
-- Script 51_create_workloads.sql
--
-- This script creates DB2 WLM workloads
--

--alter workload w1 disable ;
--drop workload w1 ;

CREATE WORKLOAD W1
    SESSION_USER ROLE ('DBAS')
    SERVICE CLASS MAIN
    POSITION AT 1;
commit;

GRANT USAGE on WORKLOAD W1 to public ;
commit;

CREATE WORKLOAD W2
    SESSION_USER ROLE ('ADHOC', 'PWRUSR')
    SERVICE CLASS MAIN
    POSITION AT 2;
commit;

GRANT USAGE on WORKLOAD W2 to public ;
commit;

CREATE WORKLOAD W3
    SESSION_USER ROLE ('GUEST')
    SERVICE CLASS MAIN
    POSITION AT 3;
commit;

GRANT USAGE on WORKLOAD W3 to public ;
commit;

```

Example B-26 shows the script for altering the defined thresholds.

Example B-26 Script 52_enforce_thresholds.sql

```
--
-- 52_enforce_thresholds
--

-- This script alter the thresholds defined in WLM configuration phase 1
--
-- For concurrency thresholds, queries exceeding the limit will be
-- put on a queue.
-- For timeout thresholds, queries exceeding the limit will be
-- terminated.
--
-- Create threshold for TRIVIAL subclass -----
--
ALTER THRESHOLD TH_TIME_SC_TRIVIAL
WHEN ACTIVITYTOTALTIME > 1 MINUTE
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION ;

-- Create threshold for MINOR subclass -----
--
ALTER THRESHOLD TH_TIME_SC_MINOR
WHEN ACTIVITYTOTALTIME > 5 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION ;

-- Create threshold for SIMPLE subclass -----
--
ALTER THRESHOLD TH_TIME_SC_SIMPLE
WHEN ACTIVITYTOTALTIME > 30 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION ;

-- Create threshold for MEDIUM subclass -----
--
ALTER THRESHOLD TH_TIME_SC_MEDIUM
WHEN ACTIVITYTOTALTIME > 60 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION ;

-- Create threshold for COMPLEX subclass -----
--
-- elapsed time:
ALTER THRESHOLD TH_TIME_SC_COMPLEX
WHEN ACTIVITYTOTALTIME > 240 MINUTES
COLLECT ACTIVITY DATA on COORDINATOR WITH DETAILS STOP EXECUTION ;

-- Lists the existing thresholds and corresponding types
--
```

```
select varchar(THRESHOLDNAME,25) as Threshold_name, varchar(THRESHOLDPREDICATE,25) as
Threshold_Type, maxvalue from syscat.thresholds ;
```

Example B-27 shows the script to change the work class set definitions.

Example B-27 53_alter_workclasses.sql

```
--
-- Script 53_alter_workclasses.sql
--
-- This script changes the Work Class Set definitions
--
ALTER WORK CLASS SET "WORK_CLASS_SET_1"
-- ALTER WORK CLASS "WCLASS_TRIVIAL" FOR TIMERONCOST FROM 0      to 5000      POSITION
AT 1
-- ALTER WORK CLASS "WCLASS_MINOR"   FOR TIMERONCOST FROM 5000  to 30000    POSITION
AT 2
ALTER WORK CLASS "WCLASS_SIMPLE"     FOR TIMERONCOST FROM 30000 to 400000   POSITION
AT 3
ALTER WORK CLASS "WCLASS_MEDIUM"     FOR TIMERONCOST FROM 400000 to 5000000  POSITION
AT 4
-- ALTER WORK CLASS "WCLASS_COMPLEX"  FOR TIMERONCOST FROM 5000000 to UNBOUNDED
POSITION AT 5 ;
;
COMMIT ;
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

The following IBM Redbooks publication provides additional information about the topic in this document. Note that publications referenced in this list might be available in softcopy only.

- ▶ *DB2 Performance Expert for Multiplatforms V2.2*, SG24-6470

You can search for, view, or download Redbooks publications, Redpaper publications, Technotes, draft publications, and Additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Smart Analytics System:
<http://www.ibm.com/software/data/infosphere/smart-analytics-system/>
- ▶ Database Management:
<http://www.ibm.com/software/data/management/>
- ▶ DB2 9.7 Manuals:
<http://www1.ibm.com/support/docview.wss?rs=71&uid=swg27015148>
- ▶ DB2 9.7 Features and benefits:
<http://www-01.ibm.com/software/data/db2/9/features.html>

Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services

Index

A

- active-passive configuration 34
- activity consuming
 - CPU 169
- administration node 5
- administrative IP interface 126
- administrative view 111–112
- AIX command
 - ioo 207
 - no 208
 - vmo 205
- antijoin 219
- application consuming
 - CPU 155
- application control shared heap segment 190
- application level shared memory 192
- application shared memory set 190
- asynchronous I/O 211

B

- backing up data
 - guidelines 100
- backup and recovery
 - database 97
 - Linux-based Smart Analytics System 94
- bcudomain 38
- BI type 1 node 71
- BI type 2 node 71
- block I/O 237
- buffer pool 236–237
- buffer pool activity 119
- buffer pool snapshot 184
- business intelligence 32, 59
 - module 6
 - moving database resource 73
 - node type 60
 - start and stop module 65

C

- cache 118
- capacity planning 275
 - Smart Analytics System 274

- cardinality 179
- catalog table space 100
- changing date and time 80
- chrg command 58
- chuser command 213
- Cognos component 60
- Cognos Content Manager 72
- command
 - db2inspf 103
 - db2start 45
 - db2stop 45
 - hafailover 47
 - inspect 103
 - lsrpnode 45
 - mmmount 45
 - mmumount 45
 - smactrl 47
- communication group 33
- concurrency level 233
- configuration parameter 204
- connections event 111
- core size 212
- core warehouse node 36
- CPU
 - application consuming 155
 - other activity consuming 169
 - usage high 152
- CPU resources
 - global system level 137
 - lifetime usage 140
 - node level 138
 - process level 141
- CPU usage elapsed time 153
- cpu_parallelism 168
- Cristie Bare Machine Recovery 96
- Customer Solution Center (CSC)
 - activities for IBM Smart Analytics System 26
- customer worksheet
 - describe 20

D

- data center and delivery information 23
- data corruption

- recovery 103
- data module 5, 278
- data size 212
- data skew 169
- data warehouse database
 - design considerations for recovery 98
- database
 - backup and recovery 97
 - table 176
- database and operating system configuration information 22
- database configuration parameters
 - chnpggs_thresh 228
 - dft_prefetch_sz 228
 - locklist 228
 - logbufsz 228
 - logfilsiz 228
 - logprimary 228
 - logsecond 228
 - maxlocks 228
 - mirrorlogpath 228
 - newlogpath 228
 - num_io_cleaners 229
 - num_io_servers 228
 - pckcachesz 228
 - self_tuning_mem 187
 - sortheap 228
 - stmtheap 228
 - util_heap_sz 228
 - wlm_collect_init 228
- database manager configuration parameters
 - comm_bandwidth 222
 - cpuspeed 222
 - database_memory 186
 - fcm_num_buffers 222
 - instance_memory 186
 - numdb 222
 - sheapthres 222
- database manager configuration setting 222
- database manager global snapshot 223
- database manager shared memory set 188
- database manager configuration parameters
 - diagpath 222
- database partition group 170
- database shared memory 186, 229
- database shared memory set 189
- data-collect mode 264
- date and time change 80
- DB2
 - buffer pool hit ratio 175
 - I/O metrics 174
 - memory allocation 187
 - memory usage 186
 - monitoring 196
 - network usage 193
 - performance troubleshooting
 - DB2 152
 - utility 168
 - DB2 callout script 231
 - DB2 EDU 153
 - DB2 level monitoring 275
 - DB2 message log management 76
 - DB2 registry variables
 - b2comm 219
 - db2_antijoin 219
 - db2_extended_optimization 219
 - db2_parallel_io 219
 - db2rshcmd 219
 - DB2 relational monitoring interface 114
 - DB2 roles
 - creating 268
 - DB2 table space 237
 - DB2 thread 152
 - DB2 Workload Manager 243
 - DB2 workload manager
 - configuring for IBM Smart Analytics System 247
 - creating new workload 269
 - tuning environment 267
 - working with 244
 - db2_all command 133
 - db2_parallel_io 221
 - db2_parallel_io registry variable 218
 - db2_parallel_io setting 220
 - db2advis utility 177
 - db2dart command
 - db2dart 103
 - db2dback shell script 76
 - db2diag utility 77
 - db2diag.log 76, 153
 - db2inspf command 103
 - db2lfrmX thread 166
 - db2loggw thread 169
 - db2lrid process 166
 - db2mtrk command 116
 - db2pd 128
 - db2pd command 116, 166, 229
 - db2pd -dbptnmem output 191
 - db2pd -edus command 152

- db2pd -fcm command 224
- db2pd utility 224
- db2start command 45
- db2stop command 45
- db2support -archive 76
- db2sysc 149
- db2sysc process 156
- db2sysc UNIX process 130
- db2top 128
- db2top utility 115, 155
- deadlocks 229
- dimension table 275
- direct read 184
- direct write 184
- disk mirroring 32
- dmesg -c command 143
- dsh utility on 129
- dual port network adapter 32
- dynamic memory growth 186

E

- engine dispatchable unit 153
- equivalency 34, 38
- equivalency status 44
- event monitor 111, 231
- external storage 32

F

- failover module 5
- FCM buffers usage 275
- FCM resources 225
- FCM shared memory set 188
- FCM_NUM_BUFFERS configuration parameter 224
- Fibre Channel device setting
 - lg_term_dma 209
 - max_xfer_size 209
 - num_cmd_elems 210
- Fibre Channel parameters 209
- file descriptor 212
- file size 212
- file system 58
- file system caching 276
- floor diagram and specification 25
- FMP shared memory set 188
- for each physical node 129
- formatting output
 - performance command 135

- foundation module 4

G

- gateway Service IP 71
- general parallel file system (GPFS) 38
- get_db_snap 232
- global memory parameter 186
- guidelines
 - backup data 100
 - recovery 103

H

- hafailback command 40
- hafailover command 40, 47
- hals command 39
- hard limit 212
- Hardware Management Console (HMC) 24
- haret command 40
- hash 223
- hash join 181, 233
- hash join overflow 179
- Hash join spill 179
- hastartdb2 command 39, 43
- hastartnfs command 40
- hastopdb2 command 39, 44
- hastopnfs command 40
- hdisk device settings
 - algorithm 211
 - max_transfer 210
 - queue_depth 211
 - reserve_policy 211
- high availability 31
- high availability group 35
 - for IBM Smart Analytics System 5600 35
 - for IBM Smart Analytics System 7600 36
 - for IBM Smart Analytics System 7700 37
- high availability management
 - OLAP nodes 54
 - warehouse applications 54
- high availability management toolkit 39
- high availability resource
 - monitoring 41
- high water mark 118
- HMC (Hardware Management Console) 24
- host bus adapter 98

I

- I/O activity 169
- I/O Completion Port 211
- I/O consumption 143
 - check node level 143
 - hardware level 145
 - identify node 142
- I/O metrics
 - DB2 174
- I/O usage 169, 174
 - application usage 170
- IBM Smart Analytics System
 - architecture 4
 - contains 2
 - description 2
 - installation 26
 - installation at customer site 27
 - installation report 29
- IBM Smart Analytics System installation report 29
- ibmdefaultbp 236
- ifconfig command 150
- index
 - usage 177
- ineligible list 48
- information server modules 7
- inspect command 103
- installation
 - IBM Smart Analytics System 26
 - IBM Smart Analytics System at the customer site 27
- installation report for IBM Smart Analytics System 29
- internal application network 209
- internal cluster network 124
- internal communication 193
- internal heap 188
- iostat 128
- ipcs -l command 214

J

- j2_maxPageReadAhead 207
- j2_minPageReadAhead 207
- jumbo frames 209

K

- kernel IPCS parameters 213
 - kernel.msgmax 213
 - kernel.msgmnb 213

- kernel.msgmni 213
- kernel.sem 213–214
- kernel.shmall 214
- kernel.shmmax 214
- kernel.shmmni 214
- kernel memory 276
- kernel parameters 204
 - Linux 213
- kernel TID 153

L

- Linux kernel parameters
 - kernel.suid_dumpable 215
 - randomize_va_space 215
 - vm.dirty_background_ratio 216
 - vm.swappiness 215
- list utility show detail statement 169
- location relationship 34
- lock timeouts 229
- locklist 229
- logical database node 131
- logical database partition 149
- logical partition 224
- long-term history data 119
- low watermark 225
- lsattr command 210–211
- lsrpnod
 - command 45
- LUN identifier 147

M

- mail relay server 123
- management modules 4
- management node 4, 126
- manual failover
 - business intelligence node 69
- manual node fallback 48
- manual node failover
 - for maintenance 46
- maxappls 232
- maximum high water mark usage 181
- maxuproc parameter 212
- mbufs kernel memory buffer 208
- memory pool allocation 116
- memory usage calculation 191
- mksysb 91
- mksysb backup
 - restore 90

- mmmount command 45
- mmumount command 45
- mon_format_lock_name 231
- mon_get_appl_lockwait 231
- mon_get_bufferpool 182, 184
- mon_get_connection 231–232
- mon_get_fcm 225
- mon_get_locks 231
- mon_get_table 176
- monitor heap 188
- monitoring high availability resource 41
- mpio_get_config -AR command 145
- mppUtil -a command 147
- mppUtil -S command 146
- multidimensional clustering 235

N

- netstat command 150
- network
 - usage DB2 193
- network buffers 276
- network information 21
- network interface 209
- network parameters
 - ipqmaxlen 208
 - rfc1323 207
 - sb_max 207
 - tcp_recvspace 208
 - tcp_sendspace 208
 - udp_recvspace 208
 - udp_sendspace 208
- network utilization 275
- NFS file system 90
- NFS service 40
- non-logged data 99
- num_ioservers 218, 220, 238

O

- OLAP node 51
- OLAP nodes
 - high availability management 54
- operating system
 - backup and recovery 89
 - monitoring 196
 - parameters 204
 - performance troubleshooting 137
- OS level monitoring 275
- overflows 233

P

- package cache 232
- page cleaning 178
- paging
 - most consuming 148
- pattern analysis 275
- pckcachesz 232
- peer domain 33
- performance command
 - db2_all 132
 - dsh 129
 - for multiple physical nodes 129
 - formatting output 135
 - rah 130
- performance degradation 218
- performance issue 128
- performance trouble shooting command 129
- performance troubleshooting
 - CPU resource 137
 - operating system 137
 - Smart Analytics System 128
- performance troubleshooting commands
 - running 129
- physical node 130
- physical-level UNIX process 130
- pidstat command 140
- pkg_cache_size_top 232
- point-in-time recovery 100
- post threshold sorts 234
- prefetch 178
- prefetch ratio 178
- primary node 46
- private memory 186, 191
- private memory allocation 187, 191
- private sorts 233
- process model 153
- product object tree 92
- ps aux command 149
- ps command 138, 166

Q

- query workload 275
- queue spills 183

R

- rack diagram 25
- rah utility 130
- RAID disk arrays 32

- RAID disk container 241
- RAID segment 221
- recovery
 - data corruption 103
 - guidelines 103
- recovery function 94
- recovery point objective 97
- recovery scope 103
- recovery time objective 97
- Redbooks publications website 319
- Redbooks Web site
 - Contact us xii
- redundant network switch 32
- redundant SAN switch 32
- regular table space 237
- relational monitoring function 184
- Reliable Scalable Cluster Technology (RSCT) 33
- Remote Support Manager (RSM) 24
- reorgchk_tb_stats 176
- resource bottleneck 275
- Resource group 33
- restore
 - mksysb backup 90
 - table space 104
- round robin 239
- RSM (Remote Support Manager) 24

S

- sar 128
- service classes 245, 247
 - creating 250
 - verifying 256
- service IP 37, 58
- set util_impact_priority command 168
- shared memory 186
- shared memory allocation 187
- shared memory set 188
- shared memory sets
 - application 190
 - application level 190
 - database 189
- sheaphres 233
- shell script 76
- smactrl command 47
- Smart Analytics System
 - application server environment 51
 - backup and recovery 88
 - building block example 7

- business intelligence 59
- business intelligence node high availability strategy 62
- capacity planning 274
- data center and delivery information 24
- data skew 195
- documentation and support 28
- expanding 7
- I/O activity 169
- information required for building by category 20
- Linux-based backup and recovery 94
- network information 21
- performance troubleshooting 128
- planning for 20
- portfolio 7
- product comparison 13
- redundant components 32
- server information 21
- software and firmware stacks 82
- software levels 83
- user and group information 23
- Smart Analytics System 1050 8
- Smart Analytics System 2050 8
- Smart Analytics System 5600 8
- Smart Analytics System 7600 10
- Smart Analytics System 9600 13
- snap_get_db 230
- snapshot monitor 111, 115
- snapshot option 115
- SNMP trap 124
- sort spill 179, 183
- sorthead 222, 233
- space management 118
- SQL statement 160
- SQL workload 128
- SSD container 241
- stack size 212
- sysdefaultmaintenanceclass 247
- sysdefaultsystemclass 247
- sysdefaultuserclass 247
- System Analytics System 7700 10
- system CPU usage 153
- system error log 124
- system temporary table 183

T

- table compression 183
- table function 111

- mon_get_activity_details 114
- mon_get_bufferpool 114
- mon_get_connection 114
- mon_get_connection_details 114
- mon_get_container 114
- mon_get_extent_movement_status 114
- mon_get_index 114
- mon_get_pkg_cache_stmt 114
- mon_get_pkg_cache_stmt_details 114
- mon_get_service_subclass 114
- mon_get_service_subclass_details 114
- mon_get_table 114
- mon_get_tablespace 114
- mon_get_unit_of_work 114
- mon_get_unit_of_work_details 114
- mon_get_workload 114
- mon_get_workload_details 114
- table queue buffer 179
- table queue overflow 181
- table queue spill 179
- table reorganization 174
- table space
 - I/O 179
 - restore 104
 - temporary 179
- table space container 239
- table space extent size 220
- table space level backup 100
- table space paramters
 - extent size 238
 - overhead 238
 - prefetch size 238
 - ransferrate 238
- table space prefetch size 220
- table spaces parameter 238
- tcp_recvspace 207
- tcp_sendspace 207
- temp operator 179
- temporary table space 179, 239
- temporary table spaces 239
- temporary tables compression 183
- text-based GUI interface 115
- threshold 223, 270
- threshold queues 256
- Tivoli Storage Manager (TSM) 94
 - backup 95
 - restore 95
- Tivoli System Automation for Multiplatforms (SA MP) 32

- configuration 33
 - for IBM Smart Analytics System 33
 - server group 32
- top 128
- topaz 128
- topaz command 139
- trace shared memory segment 188
- transferrate parameters 239
- troubleshooting connections
 - business intelligence node 74

U

- udp_recvspace 207–208
- udp_sendspace 207
- uptime 128
- uptime command 130
- user module 5, 278
- user node 5
- utilities
 - consuming CPU 166
 - high I/O 184
 - usage 179

V

- vectored I/O 237
- virtual memory manager 205
- vmstat 135

W

- warehouse application module 6
- warehouse application node 51
 - manual failback 57
 - manual failover 56
- warehouse applications
 - high availability management 54
- warehouse applications module 5
- wlm_collect_int 256, 263
- wlm_collect_stats 256
- work action set 252, 300
- work action sets
 - creating 252
- work class 256
- work class set 248, 252
- work class sets
 - creating 252
- workload 256
 - monitoring default 266



IBM Smart Analytics System

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



IBM Smart Analytics System



Understand IBM Smart Analytics System configuration

The IBM Smart Analytics System is a fully-integrated and scalable data warehouse solution that combines software, server, and storage resources to offer optimal business intelligence and information management performance for enterprises.

Learn how to administer IBM Smart Analytics System

This IBM Redbooks publication introduces the architecture and components of the IBM Smart Analytics System family. We describe the installation and configuration of the IBM Smart Analytics System and show how to manage the systems effectively to deliver an enterprise class service.

Integrate with existing IT systems

This book explains the importance of integrating the IBM Smart Analytics System with the existing IT environment, as well as how to leverage investments in security, monitoring, and backup infrastructure. We discuss the monitoring tools for both operating systems and DB2. Advance configuration, performance troubleshooting, and tuning techniques are also discussed.

This book is targeted at the architects and specialists who need to know the concepts and the detailed instructions for a successful Smart Analytics System implementation and operation.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks