Db2 11 for z/OS

Data Sharing: Planning and Administration



SC19-4055-05

Notes

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from PDF format manuals for Db2 11 for z/OS (Db2 Home).

2020-07-30 edition

This edition applies to Db2® 11 for z/OS® (product number 5615-DB2), Db2 11 for z/OS Value Unit Edition (product number 5697-P43), and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

© Copyright International Business Machines Corporation 1994, 2020.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information..... vii
Who should read this information..... vii
Db2 Utilities Suite for z/OS..... vii
Terminology and citations..... viii
Accessibility features for Db2 11 for z/OS..... viii
How to send your comments..... ix

Chapter 1. Introduction to Db2 data sharing..... 1
Advantages of Db2 data sharing..... 1
Improved data availability..... 1
Extended processing capacity..... 2
Configuration flexibility..... 3
Higher transaction rates..... 6
How Db2 protects data consistency..... 6
How an update happens..... 7
How Db2 writes changed data to disk..... 10
Implications of enabling Db2 data sharing..... 10
Communication options for data sharing groups..... 11
Database administration for data sharing..... 11
Options that affect data sharing performance..... 12
Commands for data sharing..... 12
Data recovery in data sharing environments..... 13
Maintenance of data sharing groups..... 14

Chapter 2. Consolidating data sharing members..... 17
Potential configuration changes when you consolidate data sharing members..... 19

Chapter 3. Removing members from the data sharing group..... 23
What data sets to keep when you quiesce a data sharing member..... 23
Quiescing a data sharing member..... 23
Deleting data sharing members..... 24
Deactivating data sharing members..... 24
Destroying data sharing members..... 25
Restoring deactivated data sharing members..... 26

Chapter 4. Communicating with data sharing groups..... 27
Ways to access data sharing groups..... 27
TCP/IP access methods..... 28
Group access..... 29
DNS network addressing..... 40
Member-specific access..... 41
Single-member access..... 45
Setting up Db2 for z/OS as a requester..... 45
Configuring data sharing groups as TCP/IP servers..... 47
Connecting distributed partners in a TCP/IP network..... 51
SNA access methods..... 56
Member-routing access..... 57
Group-generic access..... 58
Single-member access..... 58
Setting up Db2 for z/OS as a requester..... 59

Configuring data sharing groups as servers..... 61
Connecting distributed partners in an SNA network.....65
Preventing a member from processing requests.....69
Update the BSDS with the DSNJU003 utility..... 69

Chapter 5. Operating with data sharing..... 73
Commands for data sharing environments.....73
Command routing.....73
Command scope.....73
Commands issued from application programs..... 74
Command authorization.....74
Where messages are received..... 74
Effect of data sharing on sequence number caching.....74
Starting a data sharing member..... 75
Stopping a data sharing member..... 75
States of connections and structures after stopping Db2.....75
Normal shutdown.....75
Abnormal shutdown.....76
Submitting work to be processed.....76
Group attachments and subgroup attachments..... 76
CICS and IMS applications with Db2 data sharing.....77
Online utility jobs in data sharing environments.....77
Stand-alone utility jobs..... 78
Monitoring the group..... 78
Obtaining information about the group..... 78
Obtaining information about structures..... 80
Obtaining information about group buffer pools.....81
Database monitoring options.....82
Determining the data sharing member on which SQL statements run.....84
Controlling connections to remote systems in a data sharing environment..... 85
Starting and stopping DDF..... 85
Monitoring connections to remote systems..... 85
Resetting generic LU information..... 85
Logging environment for data sharing.....86
The impact of archiving logs in a data sharing group.....86
How to avoid using the archive log..... 87
Recovering data in data sharing..... 88
How recovery works in a data sharing group.....88
Improving recovery performance..... 89
Recovery options for data sharing environments.....90
System-level point-in-time recovery.....91
Recovering a data sharing group in case of a disaster..... 91
Recovery of pages on the logical page list.....94
How Db2 recovers from coupling facility failures..... 95
Resolution of transaction manager indoubt units of recovery.....109
Restarting Db2 after termination in a data sharing environment..... 111
Active and retained locks for Db2data sharing members.....112
Restart light.....113
Group restart phases..... 115
Protection of retained locks: failed-persistent connections..... 118
Handling coupling facility connections that hang..... 119
Postponed backout in a data sharing environment..... 121
Restarting a member with conditions..... 122
Deferring recovery during restart..... 123
Starting duplexing for a structure..... 123
Stopping duplexing for a structure.....124
Shutting down the coupling facility.....125

Chapter 6. Performance monitoring and tuning for data sharing environments.... 127
Monitoring tools for Db2 data sharing....127
Improving the performance of data sharing applications....128
Db2 address spaces involved in distributed data processing....128
Migration of batch applications....129
Resource limit facility implications for data sharing....130
Improving batch processing performance in data sharing....131
Resolving synchronous read I/O problems in data sharing environments....131
Physical open of a page set of partition....132
Concurrency and locks in data sharing environments....132
Explicit hierarchical locking in data sharing....133
A locking scenario....135
Traces that indicate whether locks have been propagated....135
Improving concurrency in data sharing environments....135
Deadlock detection and resource timeouts in data sharing environments....141
Ways to monitor Db2 locking activity....144
Changing the size of the lock structure....151
Tuning group buffer pools....154
Assigning page sets to group buffer pools....154
Inter-Db2 interest and GBP-dependency....156
Physical locks in data sharing....160
Read operations....164
Write operations....166
Group buffer pool thresholds....172
Ways to monitor group buffer pools....175
Determining the correct size and ratio of group buffer pools....180
Changing group buffer pools....189
Access path selection in a data sharing group....193
Effect of member configuration on access path selection....193
How EXPLAIN works in a data sharing group....194
How Db2 maintains in-memory statistics in data sharing....194

Information resources for Db2 11 for z/OS and related products....197

Notices....199
Programming interface information....200
Trademarks....200
Terms and conditions for product documentation....201
Privacy policy considerations....201

Glossary....203

Index....205

About this information

This information is the main source of information about using Db2 data sharing. You can use the information to learn about Db2 data sharing and to do many of the tasks that are associated with Db2 data sharing.

However, there are many tasks that are associated with Db2 data sharing, especially those of setting up the hardware and software environment for the Parallel Sysplex®, that require the use of other product libraries, such as z/OS. If you are installing Db2 and plan to use data sharing capabilities, use the *Db2 for z/OS Installation and Migration Guide* to do initial planning and develop your installation strategy. You can find detailed installation procedures in the *Db2 for z/OS Installation and Migration Guide*.

Throughout this information, "Db2" means "Db2 11 for z/OS". References to other Db2 products use complete names or specific abbreviations.

Important: To find the most up to date content for Db2 11 for z/OS, always use IBM® Knowledge Center or download the latest PDF file from PDF format manuals for Db2 11 for z/OS (Db2 Home).

This information assumes that Db2 11 is running in new-function mode, and that your application is running with the application compatibility value of 'V11R1'.

Availability of new function in Db2 11

The behavior of data definition statements such as CREATE, ALTER, and DROP, which embed data manipulation SQL statements that contain new capabilities, depends on the application compatibility value that is in effect for the application. An application compatibility value of 'V11R1' must be in effect for applications to use new capability in embedded statements such as SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, and SET *assignment-statement*. Otherwise, an application compatibility value of 'V10R1' can be used for data definition statements.

Generally, new SQL capabilities, including changes to existing language elements, functions, data manipulation statements, and limits, are available only in new-function mode with applications set to an application compatibility value of 'V11R1'.

Optimization and virtual storage enhancements are available in conversion mode unless stated otherwise.

SQL statements can continue to run with the same expected behavior as in DB2® 10 new-function mode with an application compatibility value of 'V10R1'.

Who should read this information

This information is primarily intended for system and database administrators who are responsible for planning and implementing Db2 data sharing. Many of the task descriptions in this information assume that the user is already familiar with administering Db2 in a non-Db2 data sharing environment. See *Db2 Administration Guide* for any concepts not explained in this information.

Db2 Utilities Suite for z/OS

Important: In Db2 11, the Db2 Utilities Suite for z/OS is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them.

Db2 11 utilities can use the DFSORT program regardless of whether you purchased a license for DFSORT on your system. For more information, see the following informational APARs:

- II14047
- II14213
- II13495

© Copyright IBM Corp. 1994, 2020

Db2 utilities can use IBM Db2 Sort for z/OS (5655-W42) as an alternative to DFSORT for utility SORT and MERGE functions. Use of Db2 Sort for z/OS requires the purchase of a Db2 Sort for z/OS license. For more information about Db2 Sort for z/OS, see Db2 Sort for z/OS.

Related concepts

Db2 utilities packaging (Db2 Utilities)

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.

About the Db2 brand change: IBM has re-branded DB2 to Db2, and Db2 for z/OS is now the name of the offering previously know as "DB2 for z/OS". During and after implementation of this change, you might sometimes still see references to the original names, such as "DB2 for z/OS" and "DB2", in different IBM web pages and documents. However, assume that these refer to the same "Db2 for z/OS" product when the PID, Entitlement Entity, version, modification, and release information match. For more information, see Revised naming for IBM Db2 family products on IBM z/OS platform.

The following terms are used as indicated:

Db2
Represents either the Db2 licensed program or a particular Db2 subsystem.

Tivoli® OMEGAMON® XE for Db2 Performance Expert on z/OS
Refers to any of the following products:

- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS®
Represents CICS Transaction Server for z/OS.

IMS
Represents the IMS Database Manager or IMS Transaction Manager.

MVS™
Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF®
Represents the functions that are provided by the RACF component of the z/OS Security Server.

Accessibility features for Db2 11 for z/OS

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in z/OS products, including Db2 11 for z/OS. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size

Tip: The IBM Knowledge Center (which includes information for Db2 for z/OS) and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

For information about navigating the Db2 for z/OS ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

IBM and accessibility

See the *IBM Accessibility Center* at http://www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other Db2 for z/OS documentation.

Send your comments by email to db2zinfo@us.ibm.com and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title or a help topic title).

Chapter 1. Introduction to Db2 data sharing

In Db2 *data sharing*, applications that reside in multiple Db2 subsystems in a Parallel Sysplex can read from and write to the same Db2 for z/OS data concurrently, with integrity, performance, scalability, and dynamic workload balancing.

A *Sysplex* is a group of z/OS systems that communicate and cooperate with one another using specialized hardware and software. They are connected and synchronized through a Sysplex Timer and enterprise systems connection channels. A *Parallel Sysplex* is a Sysplex that uses one or more coupling facilities, which provide high-speed caching, list processing, and lock processing for any applications on the Sysplex.

A collection of one or more Db2 subsystems that share Db2 data is called a *data sharing group*. Db2 subsystems that access shared Db2 data must belong to a data sharing group.

A Db2 subsystem that belongs to a data sharing group is a *member* of that group. Each member can belong to one, and only one, data sharing group. All members of a data sharing group share the same Db2 catalog and directory, and all members must reside in the same Parallel Sysplex. The maximum number of members in a data sharing group is 32.

All members of a data sharing group can read and update the same Db2 data simultaneously. Therefore, all data that different members of the group can access must reside on shared disks.

Some capabilities described in this information can be used in both data sharing and non-data sharing environments. This information uses the term *data sharing environment* to describe a situation in which a data sharing group has been defined with at least one member. In a non-data sharing environment, no group is defined.

Related concepts

Planning for Db2 data sharing (Db2 Installation and Migration)

Related tasks

Installing, migrating, and enabling Db2 data sharing (Db2 Installation and Migration)

Advantages of Db2 data sharing

Db2 data sharing improves the availability of Db2 data, extends the processing capacity of your system, provides more flexible ways to configure your environment, and increases transaction rates.

You do not need to change the SQL in your applications to use data sharing, although you might need to do some tuning for optimal performance. Because Db2 data sharing does not affect the application interface, it does not create additional work for application programmers and users. However, system programmers, operators, and database administrators must perform additional tasks in a data sharing environment.

Leaves application interface unchanged

Your investment in people and skills is protected because existing SQL interfaces and attachments remain intact when sharing data.

You can bind a package or plan on one member of a data sharing group and run that package or plan on any other member of the group.

Improved data availability

More users demand access to Db2 data every hour, every day. Data sharing helps you meet your service objectives by improving data availability during both planned and unplanned outages.

Because data sharing provides multiple paths to data, a member can be down, and applications can still access the data through other members of the data sharing group. As the following figure illustrates,

© Copyright IBM Corp. 1994, 2020

when an outage occurs and one member is down, transaction managers are informed that the member is unavailable, and they can direct new application requests to another member of the group.

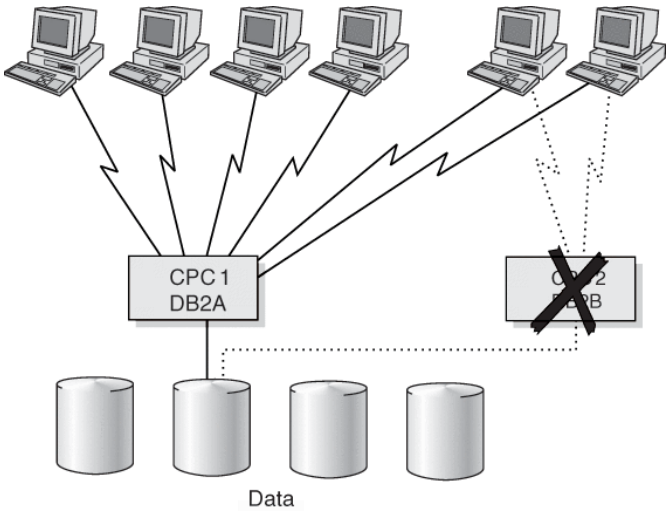


Figure 1. Data sharing improves data availability during outages

While increasing data availability has some performance cost, the overhead for interprocessor communication and caching changed data is minimal. Db2 provides efficient locking and caching mechanisms and uses coupling facility hardware. A *coupling facility* is a special logical partition that runs the coupling facility control program. It provides high-speed caching, list processing, and locking functions in a Parallel Sysplex. The Db2 structures in the coupling facility benefit from high availability.

Extended processing capacity

As you move more data processing onto Db2, your processing needs can exceed the capacity of a single system. Using data sharing can help meet your ever-increasing capacity needs.

Without Db2 data sharing

Without Db2 data sharing, you have the following options for addressing increased capacity needs:

- Copy the data, or split the data between separate Db2 subsystems.

  This approach requires that you maintain separate copies of the data. There is no communication between or among Db2 subsystems, and no shared Db2 catalog or directory.
- Install another Db2 subsystem and rewrite applications to access the data as distributed data.

  This approach might relieve the workload on the original Db2 subsystem, but it requires changes to your applications and has performance overhead of its own. Nevertheless, if Db2 subsystems are separated by great distance or Db2 needs to share data with another system, the distributed data facility is still your only option.
- Install a larger processor and move the data and applications to that machine.

  This option can be expensive. In addition, this approach demands that your system come down while you move to the new machine.

With Db2 data sharing

With Db2 data sharing, you get the following benefits:

Support for incremental growth: A Parallel Sysplex can grow incrementally, allowing you to add processing power in granular units and in a non-disruptive manner. The coupling technology of Parallel Sysplex along with the additional CPU power results in more throughput for users' applications. You no longer need to manage multiple copies of data, and all members of the data sharing group share a single Db2 catalog and directory.

Workload balancing: Db2 data sharing provides workload balancing so that when the workload increases or you add a new member to the group, you do not need to distribute your data or rewrite your applications. Db2 data sharing is unlike the partitioned-data approach to parallelism (sometimes called *shared-nothing* architecture), in which a one-to-one relationship exists between a database management system (DBMS) and a segment of data. When you add a new Db2 subsystem onto another central processor complex (CPC) in a data sharing environment, applications can access the same data through the new member just as easily as through any of the existing members.

Db2 works closely with the Workload Manager (WLM) component of z/OS to ensure that incoming requests are optimally balanced across the members of a data sharing group. All members of the data sharing group have the same concurrent and direct read-write access to the data.

Capacity when you need it: A data sharing configuration can handle peak workloads well, such as end-of-quarter processing. You can have data sharing members in reserve, bring them online to handle peak loads, and then stop them when the peak passes.

Configuration flexibility

Data sharing lets you configure your system environment much more flexibly with operational systems, decision support systems, and shared data management.

As the following figure shows, you can have more than one data sharing group on the same Parallel Sysplex. You might, for example, want one group for testing and another group for production data. This example also shows a single, non-data sharing Db2 subsystem.

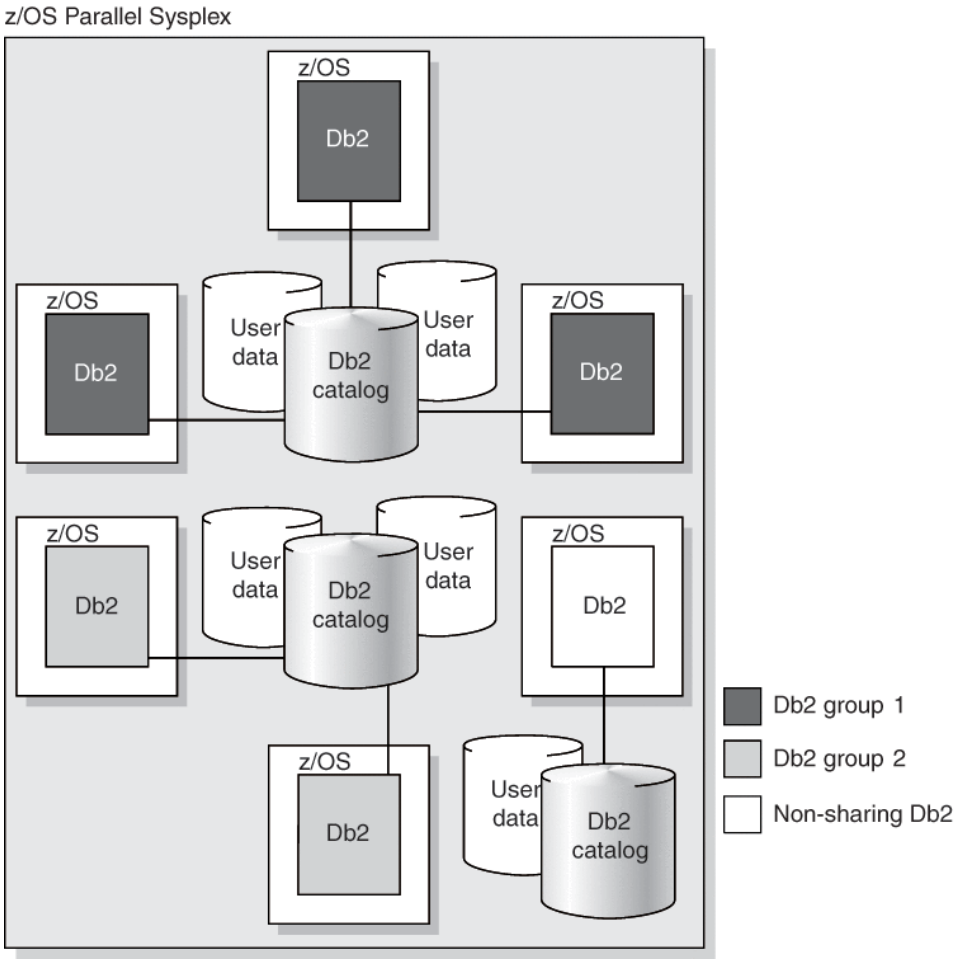


Figure 2. A possible configuration of data sharing groups

Flexible operational systems

The following figure shows how, with data sharing, you can have query user groups and online transaction user groups on separate z/OS images. This configuration lets you tailor each system specifically for that user set, control storage contention, and provide predictable levels of service for that set of users. Previously, you might have had to manage separate copies of data to meet the needs of different user groups.

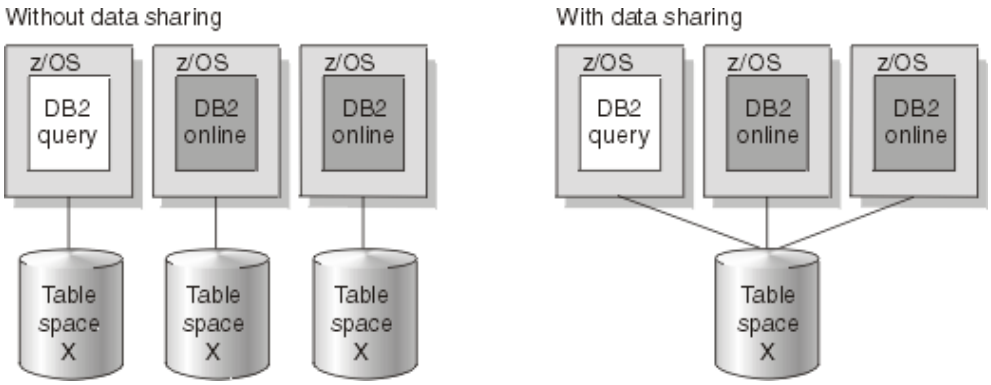


Figure 3. Flexible configurations with data sharing

Flexible decision support systems

The following figure shows two different decision support configurations. A *typical* configuration separates the operational data from the decision support data. Use this configuration when the operational system has environmental requirements that are different from those of the decision support system. For example, the decision support system might be in a different geographical area, or security requirements might be different for the two systems.

Db2 offers another option—a *combination* configuration. A combination configuration groups your operational and decision support systems into a single data sharing group and offers the following advantages:

- You can occasionally join decision support data and operational data using SQL.
- You can reconfigure the system dynamically to handle fluctuating workloads. (You can dedicate CPCs to decision support processing or operational processing at different times of the day or year.)
- You can reduce the cost of computing:
  - The infrastructure used for data management is already in place.
  - You can create a prototype of a decision support system in your existing system, and then add processing capacity as the system grows.

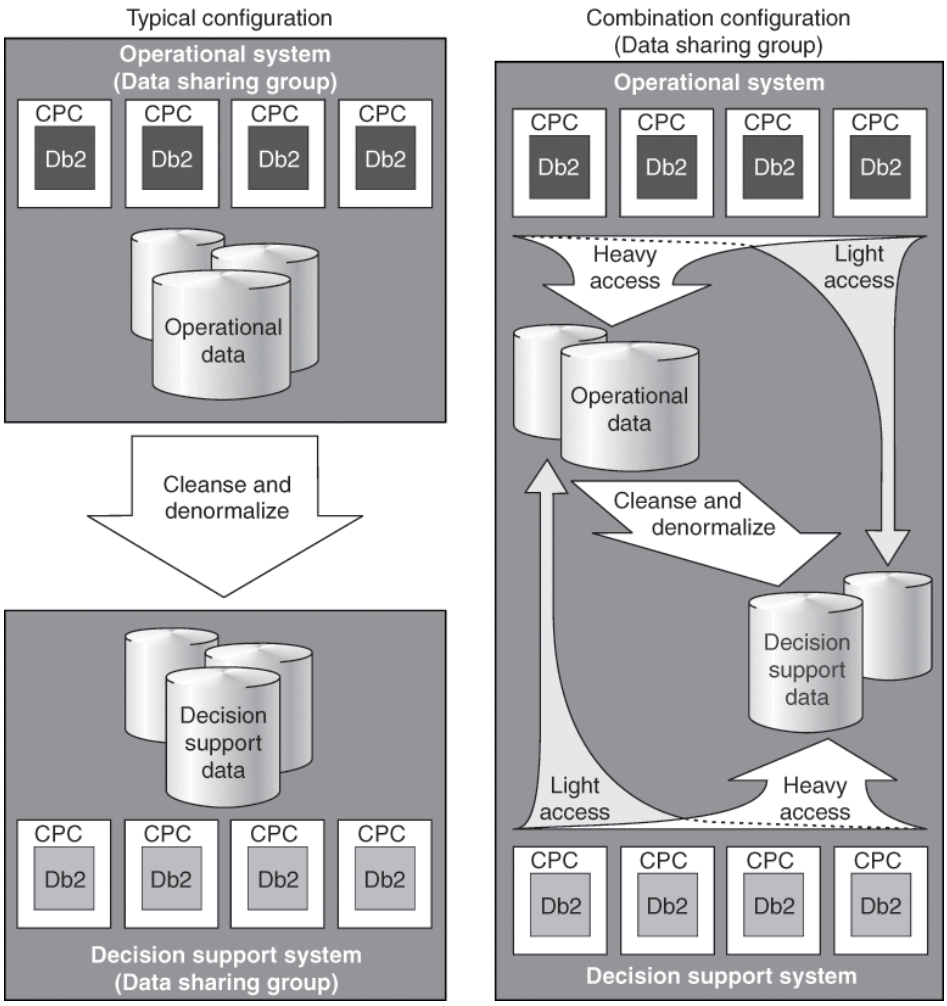


Figure 4. Flexible configurations for decision support

If you want a combination system configuration, you must separate decision support data from operational data as much as possible. Buffer pools, disks, and control units that you use to decide on a support system should be separate from those that you use in your operational system. This separation greatly minimizes any negative performance impact on the operational system.

If you are unable to maintain the needed level of separation, or if you have separated your operational data for other reasons, such as security, using a separate decision support system is your best option.

Flexible shared data management

Db2 data sharing can simplify the management of applications that must share data, such as a common customer table. Perhaps these applications were split in the past for capacity or availability reasons. But with the split architecture, the shared data must be synchronized across multiple systems (that is, by replicating data).

Db2 data sharing gives you the flexibility to configure these applications to access a single data sharing group. It also allows you to maintain a single copy of the shared data that can be read and updated by multiple systems with good performance. This is an especially powerful option when data centers are consolidated.

Higher transaction rates

Data sharing gives you opportunities to put more work through the system to produce higher transaction rates.

As the following figure illustrates, you can run the same application on more than one member to achieve transaction rates that are higher than possible on a single Db2 subsystem.

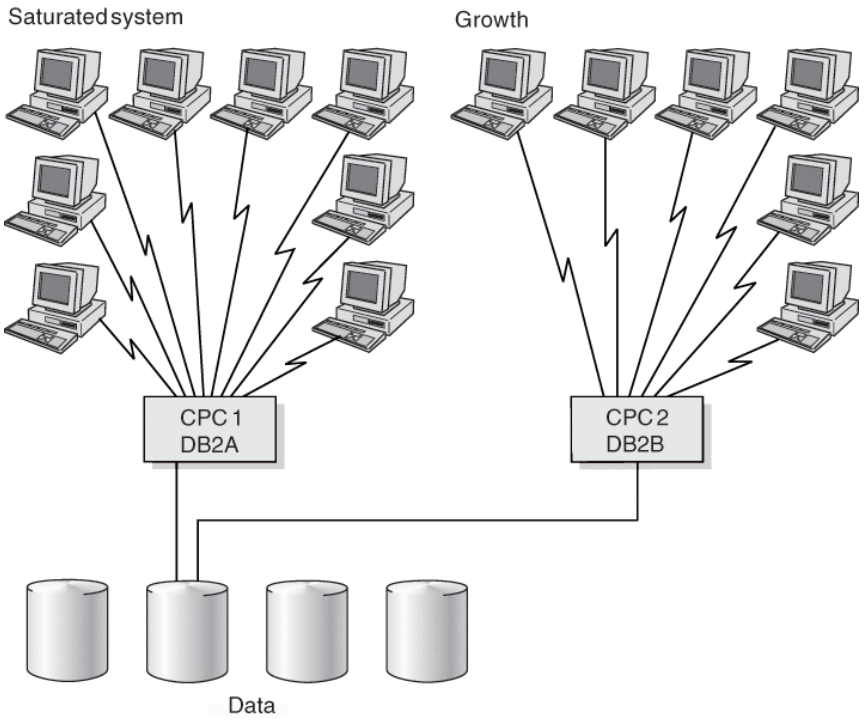


Figure 5. Data sharing enables growth

How Db2 protects data consistency

Applications can access data from any member of a data sharing group, and many members can potentially read and write the same data. Db2 for z/OS uses special data sharing locking and caching mechanisms to ensure data consistency across the applications.

When multiple members of a data sharing group open the same table space, index space, or partition, and at least one of them opens it for writing, the data is said to be of inter-Db2 read/write interest to the members. (Sometimes called "inter-Db2 interest".) To control access to data that is of inter-Db2 interest, whenever the data is changed, Db2 caches it in a storage area that is called a *group buffer pool*.

When there is inter-Db2 read/write interest in a particular table space, index, or partition, it is dependent on the group buffer pool, or *GBP-dependent* (group buffer pool-dependent).

You define group buffer pools by using coupling facility resource management (CFRM) policies.

As shown in the following figure, a mapping exists between a group buffer pool and the buffer pools of the group members. For example, each member has a buffer pool named BP0. For data sharing, you must define a group buffer pool (GBP0) in the coupling facility that maps to each member's buffer pool BP0. GBP0 is used for caching the Db2 catalog and directory table spaces and indexes, and any other table spaces, indexes, or partitions that use buffer pool 0.

Although a single group buffer pool cannot reside in more than one coupling facility (unless it is duplexed), you can put group buffer pools in more than one coupling facility.

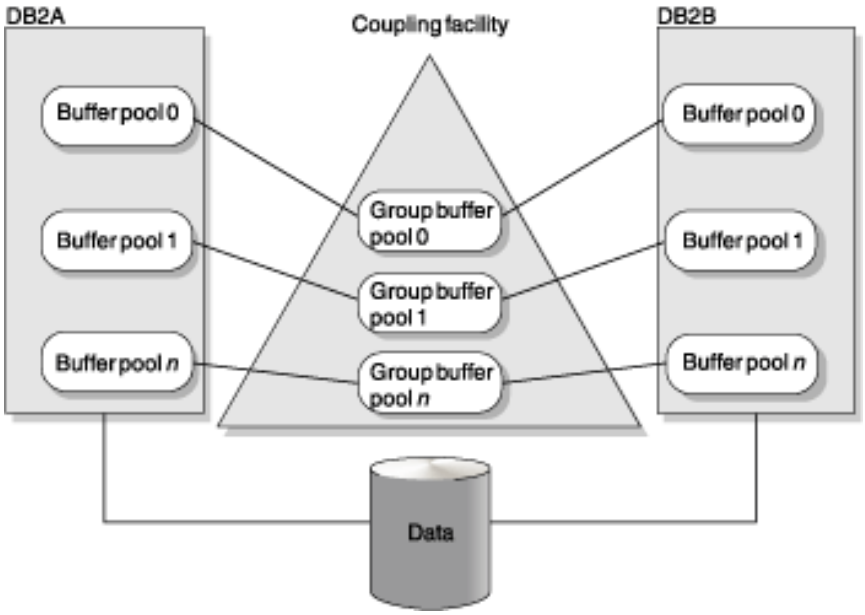


Figure 6. Relationship of member buffer pools to the group buffer pool

When you change a particular page of data, Db2 caches that page in the group buffer pool. The coupling facility invalidates any image of the page in the buffer pools associated with each member. Then, when a request for that same data is subsequently made by another member, that member looks for the data in the group buffer pool.

Performance options to fit your application's needs: By default, Db2 caches updated data, but you also have the options of caching all or none of your data. There is even an option especially for large object (LOB) data.

Related reference

Updating a CFRM Policy (z/OS MVS Setting Up a Sysplex)

How an update happens

When you are deciding whether to use data sharing, it can help to understand how data is updated in a data sharing environment.

In the following illustrations, the most recent version of the data page is shaded. This scenario assumes that the group buffer pool is used for caching changed data that is duplexed for high availability. *Duplexing* is the ability to write data to two instances of a structure: in this case, a primary group buffer pool and a secondary group buffer pool.

Suppose, as shown in the following figure, that an application issues an UPDATE statement from DB2A. The data that is being updated does not already reside in either the member's own buffer pool or the group buffer pool; therefore, DB2A retrieves the data from disk and updates the data in its own buffer pool. Simultaneously, DB2A gets the appropriate locks to prevent another member from updating the same data at the same time. After the application commits the UPDATE, DB2A releases the corresponding locks. The changed data page remains in DB2A's buffer pool.

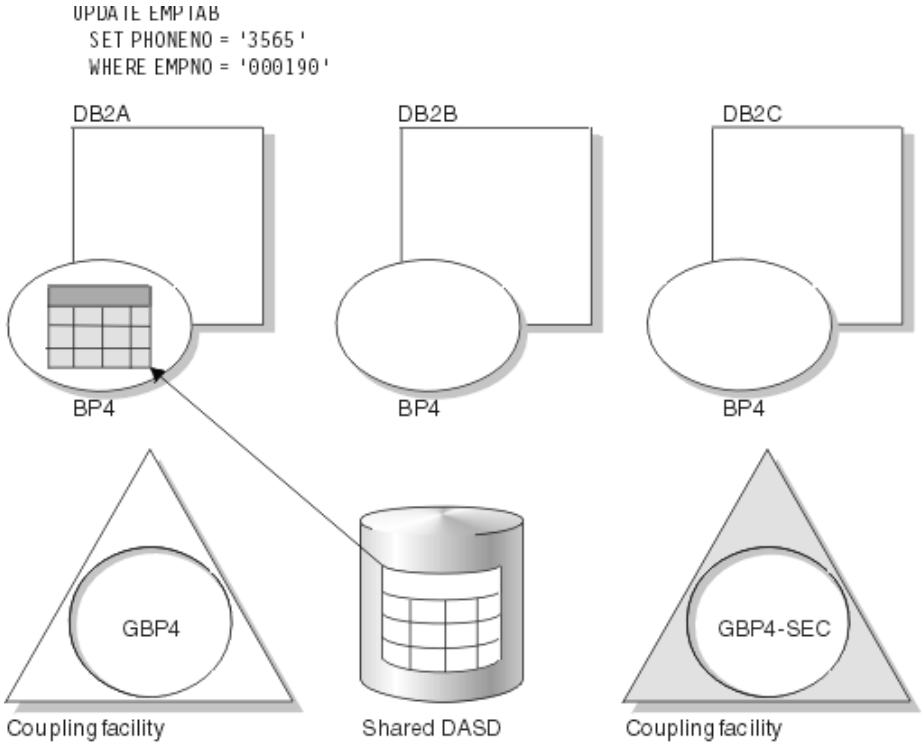


Figure 7. An application running on DB2A reads data from disk and updates it

Next, suppose another application, that runs on DB2B, needs to update the same data page as shown in the following figure. Db2 dynamically detects inter-Db2 interest in the page set, so DB2A writes the changed data page to the group buffer pools (both primary and secondary). DB2B then retrieves the data page from the primary group buffer pool.

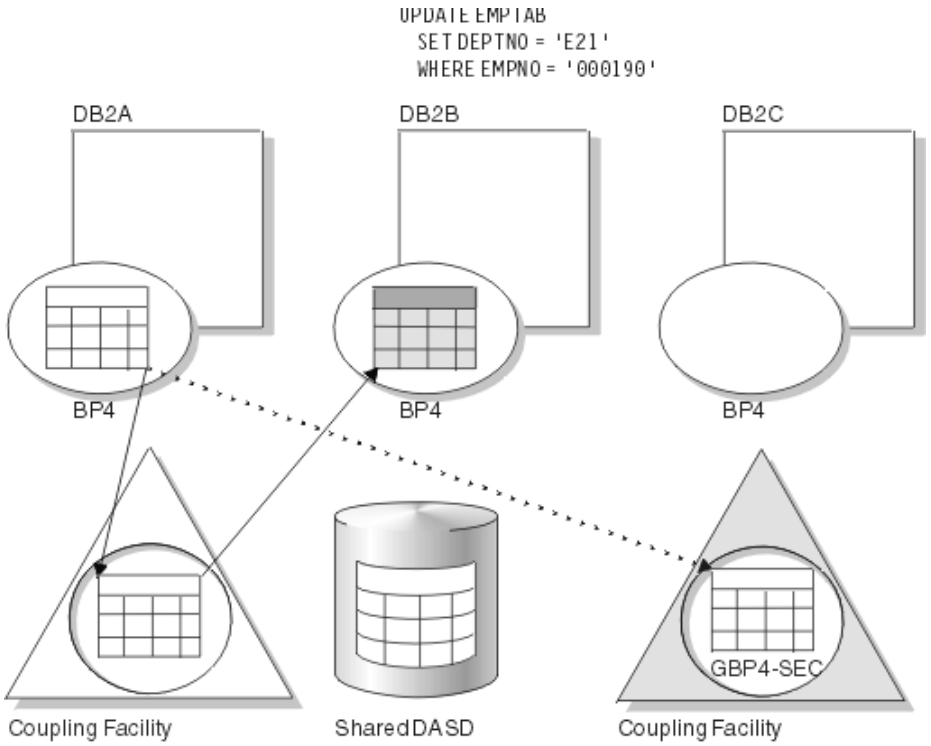


Figure 8. DB2B updates the same data page

After the application that runs on DB2B commits the UPDATE, DB2B moves a copy of the data page into the group buffer pools. This invalidates the data page in DB2A's buffer pool as shown in the following figure. Cross-invalidation occurs from the primary group buffer pool.

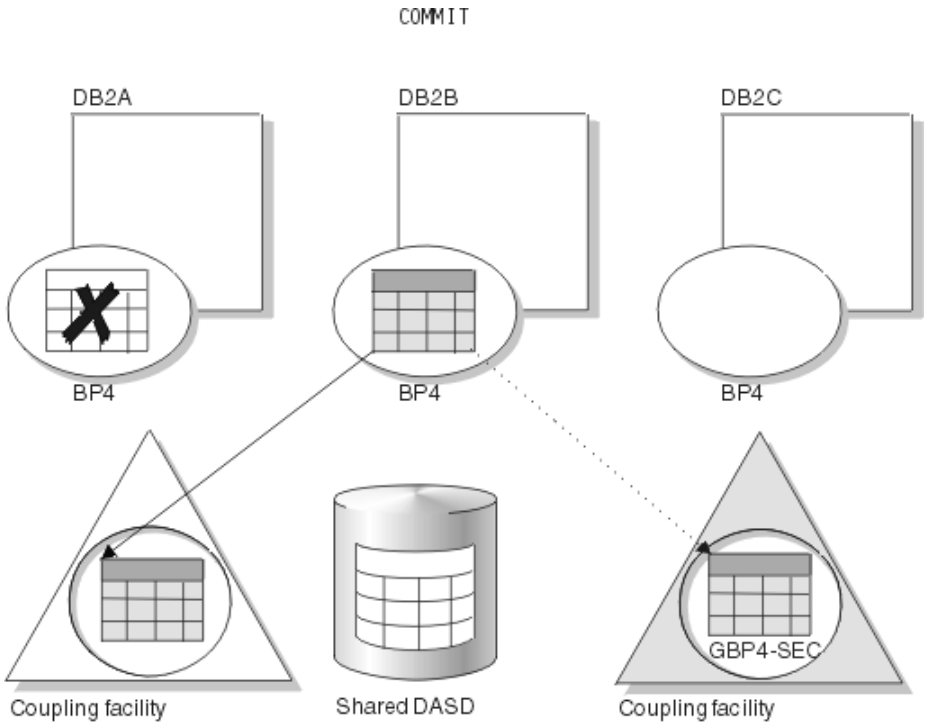


Figure 9. The updated data page is written to the group buffer pools and the data page in DB2A's buffer pool is invalidated.

Now, as shown in the following figure, when DB2A attempts to reread the data, it detects that the data page in its own buffer pool is invalid. Therefore, it reads the latest copy of the data from the primary group buffer pool.

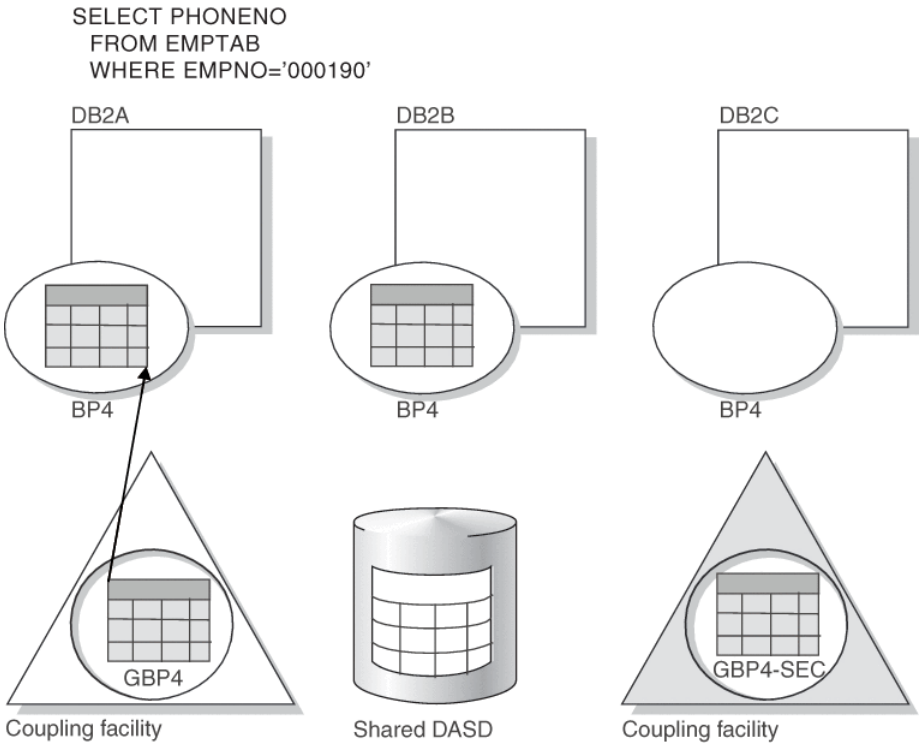


Figure 10. DB2A reads the data from the primary group buffer pool

How Db2 writes changed data to disk

Periodically, Db2 must write changed pages from the primary group buffer pool to disk. This process is called *castout*.

The member that is responsible for casting out the changed data uses its own address space because no direct connection exists from a coupling facility to disk, as shown in the following figure. The data passes through a private buffer, not through the Db2 buffer pools.

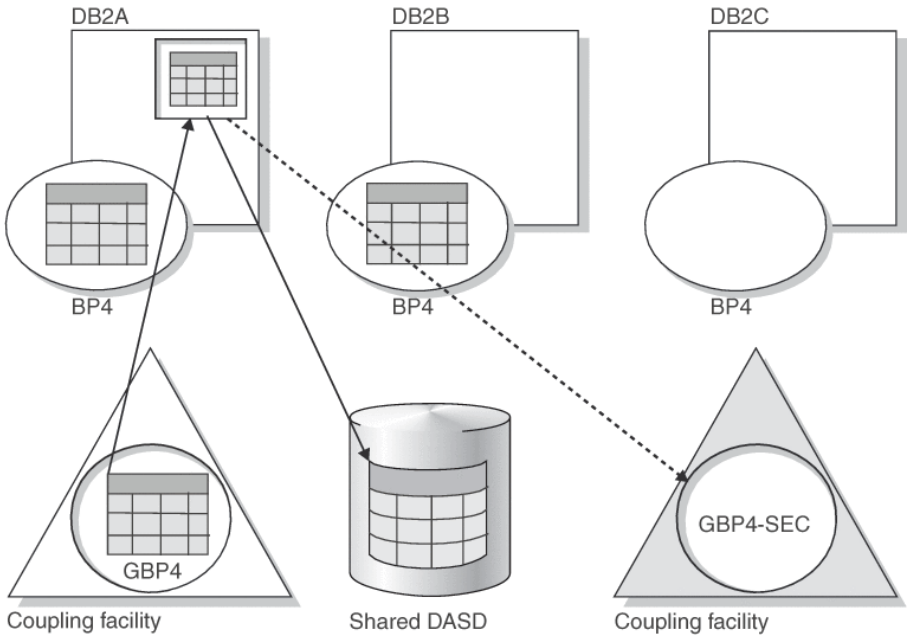


Figure 11. Writing data to disk

When a group buffer pool is duplexed, data is not cast out from the secondary group buffer pool to disk. After a set of pages is written to disk from the primary group buffer pool, Db2 deletes those pages from the secondary group buffer pool.

Implications of enabling Db2 data sharing

You must plan a naming convention before enabling data sharing on the first member (the originating member) of the group.

Because names in the Parallel Sysplex and names in the data sharing group must be unique, you must have a naming convention before you create the group. Not only must shared data objects have unique names, but you must create a unique name for every group resource.

The Db2 catalog of the originating member becomes the catalog for all the members of the data sharing group. You add additional members to the group as new installations, and those members use the originating member's Db2 catalog.

If you have data from existing Db2 subsystems that you want the group to share, you must merge the catalog definitions for that data into the group catalog. You must also ensure that all members of the group can access the data. Db2 does not provide a way to merge members' catalogs automatically.

Important: If the RBA for a non data sharing Db2 subsystem reaches or exceeds the hard or soft limit, you must either convert all table spaces and indexes to the 10-byte format or use the RBA reset procedure. Enabling data sharing is not sufficient to resolve the problem.

Related tasks

Naming in Db2 data sharing (Db2 Installation and Migration)

Communication options for data sharing groups

Applications can communicate with a data sharing group by using either Transmission Control Protocol/Internet Protocol (TCP/IP) or Systems Network Architecture (SNA) protocol.

Applications connect to a data sharing group by specifying a Db2 location name. The group provides a single-system image to requesting applications.

Related concepts

Communicating with data sharing groups
A data sharing group can be a powerful server in your client/server environment. The group can be part of a TCP/IP network, part of an SNA network, or part of a network that uses both protocols.

Database administration for data sharing

Using data sharing has implications for planning exit routines, authorizing users, and loading and reorganizing data.

Because the Db2 catalog is shared by all members of a data sharing group, data definition, authorization, and control are the same as for non–data sharing environments. Be sure that every object has a unique name, and be sure that the shared data resides on shared disks.

Planning for exit routines

If you use exit routines, such as a field or validation procedure or the access control authorization routine, ensure that all members of the group use the same routines.

Recommendation: Place all exit routines in a program library that is shared by all members of the group.

Authorizing users

Use the same authorization mechanisms that are in place for non–data sharing Db2 subsystems to control access to shared Db2 data and to members. Because all members in the group share the same Db2 catalog, an authorization ID has the same granted privileges and authorities for every member of the group.

As suggested for non–data sharing Db2 subsystems, use a security system outside of Db2 (such as RACF® or its equivalent) to control which user IDs can access which members. RACF, for example, does not recognize a data sharing group as a single resource. Therefore, you must separately define Db2 resources to RACF for each member of the group, and connect all user IDs to a RACF group that permits access to all those resources. Or you can permit separate groups of user IDs to access different sets of resources. (In the latter case, however, you cannot move work freely among all members of the data sharing group.)

Each member of a data sharing group uses the same names for the connection and sign-on exit routines. As a good practice, all members of a group should share the same exit routines. Sharing avoids authorization anomalies such as:

- Primary authorization IDs that are treated differently by different members of the group
- Primary authorization IDs that are associated with different sets of secondary IDs by different members of the group

Loading and reorganizing data

You can load or reorganize data from any member of a data sharing group.

Related reference

LOAD (Db2 Utilities)
REORG INDEX (Db2 Utilities)
REORG TABLESPACE (Db2 Utilities)

Options that affect data sharing performance

The GBPCACHE, MEMBER CLUSTER, and TRACKMOD options all affect the performance of data sharing.

GBPCACHE option

Use the GBPCACHE option when you create or alter table spaces or indexes to specify what data, if any, should be cached in the group buffer pool. Valid values for this option are NONE, SYSTEM, CHANGED, and ALL. The default is CHANGED.

MEMBER CLUSTER option

Use the MEMBER CLUSTER option when you create table spaces to specify that Db2 locate data in the table space based on available space rather than clustering the data by the implicit or explicit clustering index.

This option can benefit applications when there are many inserts to the same table from multiple members.

TRACKMOD option

Use the TRACKMOD option when you create or alter table spaces to specify whether you want Db2 to track modified pages in the space map pages of the table space or partition.

TRACKMOD NO can benefit applications when there are frequent updates from multiple members. Be aware that this option can degrade incremental image-copy performance; therefore, specify NO only if you never use incremental copies, or if you use DFSMS™ concurrent copies and LOGONLY recovery. In these cases, choosing TRACKMOD NO can improve transaction performance.

Related concepts

Options for reducing space map page contention
When you define table spaces and indexes, you have several options to help reduce space map hot spots. These hot spots can occur when a large amount of update, insert, or delete activity occurs on a page set from multiple members of a group.

Member affinity clustering
For applications that do heavy sequential insert processing from multiple members, the contention on the space map or for the data pages at the end of the table can be considerable.

Incremental image copies (Db2 Utilities)

Related reference

CREATE TABLESPACE (Db2 SQL)

Commands for data sharing

Parallel Sysplex technology lets you manage a data sharing group from a console that is attached to a single z/OS system or from consoles that are attached to multiple z/OS systems.

The following figure shows how commands are issued from a single z/OS system.

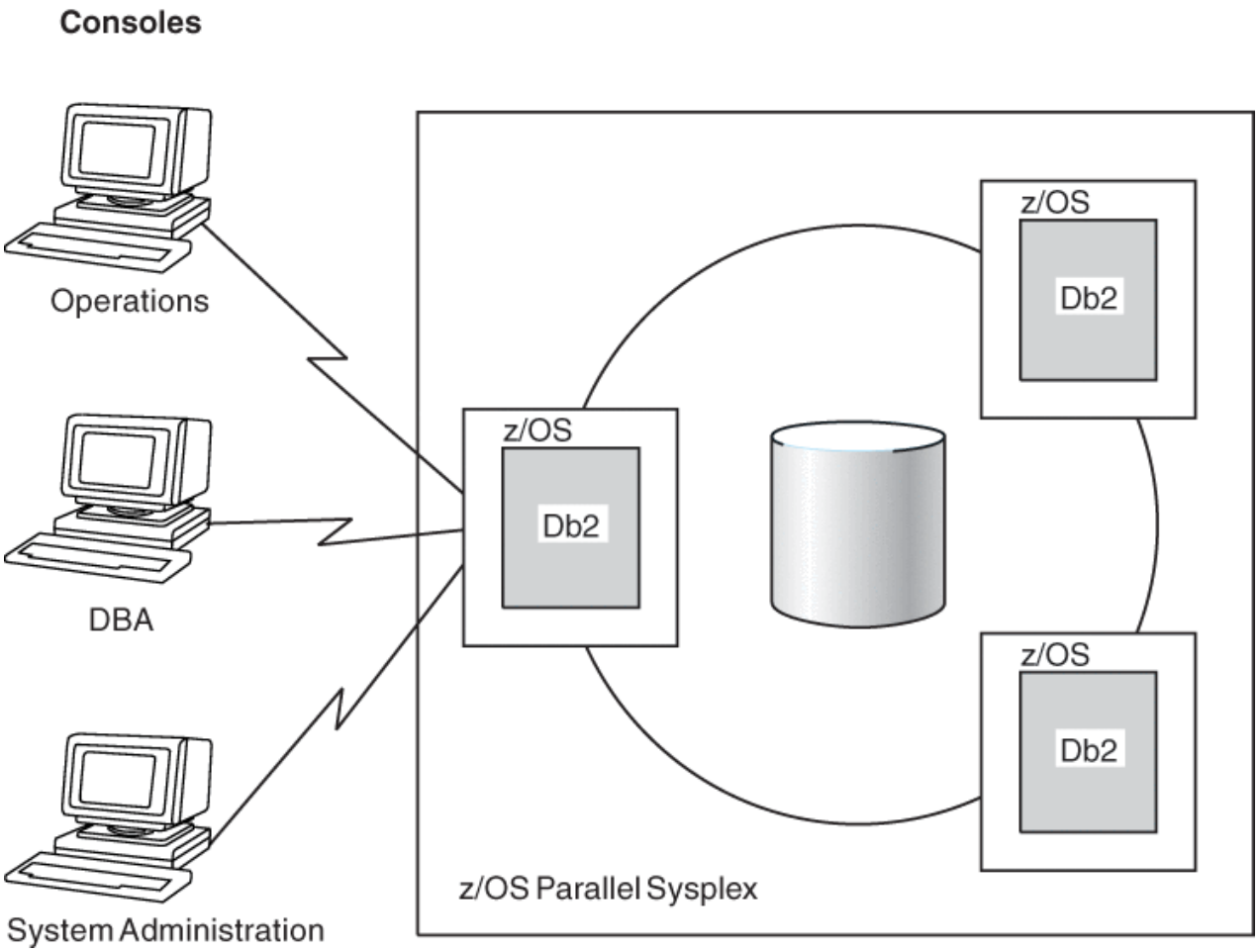


Figure 12. Issuing commands

Some commands manage group resources; others manage member resources.

Related information

About Db2 and related commands (Db2 Commands)

Data recovery in data sharing environments

Db2 recovers data from information that is contained in both the logs and the bootstrap data sets (BSDSs) of members. However, because updates can be logged on several different members, Db2 coordinates recovery by using the SCA in a coupling facility.

The *shared communications area* (SCA) contains:

- Member names
- BSDS names
- Database exception status conditions about objects and members in the group
- Recovery information, such as log data set names and the list of indoubt XA transactions

The SCA is also used to coordinate startup.

You can stop and start an individual member of a data sharing group while the other members continue to run. The startup process for each member is similar to that of non-data sharing Db2 subsystems.

Db2 uses a process called *group restart* in the rare event that critical resources in a coupling facility are lost and cannot be rebuilt. When this happens, all members of the group terminate abnormally. Group restart rebuilds the lost information from individual member logs. However, unlike data recovery, this

information can be applied in any order. Because there is no need to merge log records, Db2 can perform many of the restart phases for individual members in parallel.

Recommendation: Use an automated procedure to restart failed members of the group.

The RECOVER utility

You can run the RECOVER utility from any member of a data sharing group. The process for data recovery is basically the same for a data sharing group as it is for non-data sharing Db2 subsystems. However, updates to a single table space can be the work of several different members. Therefore, to recover an object, Db2 must merge log records from the appropriate members, using a *log record sequence number* (LRSN). The LRSN is a value derived from the store clock timestamp and synchronized across the members of a data sharing group by the Sysplex Timer.

Recommendation: Use more than one coupling facility to allow for structure duplexing and for automatic recovery in the event that a coupling facility fails.

System-level point-in-time recovery

You can perform system-level point-in-time recovery by using the BACKUP SYSTEM and RESTORE SYSTEM online utilities.

BACKUP SYSTEM online utility

This utility provides fast, volume-level copies of Db2 databases and logs. It automatically keeps track of which volumes need to be copied. Using BACKUP SYSTEM is less disruptive than using SET LOG SUSPEND in copy procedures because the log write latch is not taken. An advantage for data sharing is that BACKUP SYSTEM has group-scope, whereas SET LOG SUSPEND has only member scope.

RESTORE SYSTEM online utility

This utility provides a way to recover a data sharing group to a specific point in time. RESTORE SYSTEM automatically handles any CREATE, DROP, and LOG NO events that might have occurred between the backup and the recovery point in time.

Related concepts

Recovering data
The procedures for data recovery are fundamentally the same for data sharing as for non-data-sharing environments. Data sharing involves one catalog, but many logs and BSDSs. In addition to disk and cache controllers, the coupling facility adds a possible point of failure and requires appropriate procedures for data recovery.

Tuning group buffer pools
With Db2 data sharing, group buffer pools are a key component to ensuring that Db2 does not read down-level data in its member pools. Physical locks (P-locks) are also used in that process. Your understanding of these processes is helpful when tuning the data sharing group for best performance.

Related tasks

Planning for coupling facility availability for data sharing

Maintenance of data sharing groups

To apply maintenance, you can make most changes on one member at a time.

If you must take Db2, IRLM, or z/OS offline for a change to take place and the outage is unacceptable to users, you can move those users onto another member. Some sites find it useful to define an extra member that they bring up and down as needed to take on work while maintenance is being applied to another member.

The recommended way of testing maintenance is to apply that maintenance to a test data sharing group before moving it onto the production data sharing group.

Table 1. Actions required for planned maintenance changes

Type of change	Action required
Early code	Issue the -REFRESH Db2,EARLY command.
Db2 code	Bring down and restart each member independently.
IRLM code	Bring down and restart each IRLM member independently.
Attachment code	Apply the change and restart the transaction manager or application.
Subsystem parameters	For those that cannot be changed dynamically, update using the Db2 update process. Stop and restart the member to activate the updated parameter.

Recommendation: Consider specifying CASTOUT(NO) when you stop an individual member of a data sharing group for maintenance. This option speeds up shutdown because Db2 bypasses castout and associated cleanup processing in the group buffer pools.

Do not specify CASTOUT(NO) when you stop multiple members of a data sharing group and you need to maintain consistent data on disk. For example, if you stop all members to get a consistent copy of the databases on disk that you can send offsite, do not specify CASTOUT(NO) because some of the changed data could still reside in the group buffer pools after the members shut down. Your disk copy might not have all the most recent data.

Tip: Consider specifying a value of NODISCON for the IRLM procedure's SCOPE option to allow IRLM to continue doing work while you apply maintenance to Db2. (If you edit the IRLM procedure using the Db2 installation process, this is analogous to specifying NO for parameter DISCONNECT IRLM on installation panel DSNTIPJ.)

Applying maintenance to IRLM: Each member of a data sharing group has its own IRLM. As with Db2, you must stop and restart each IRLM in the group to roll a change throughout the group.

IRLM has a *function level* to control changes that affect the entire group. The function level for a particular IRLM member indicates the latest function that IRLM is capable of running. The *group function level* is the lowest function level that exists among the IRLMs in the group. The group function level is the function level at which all IRLMs in the group must run, even if some IRLM members are capable of running at higher function levels.

Recommendation: Keep all members of the IRLM group at the same function level. This ensures that all IRLMs are running with the same maintenance that affects the entire group. (Maintenance that does not affect the group does not increment the function level.)

To see the IRLM function levels, use the MODIFY *irlmproc*,STATUS,ALLI command of z/OS.

Related tasks

Determining the function level of an IRLM group in coexistence (Db2 Installation and Migration)

Related reference

DISCONNECT IRLM field (Db2 Installation and Migration)

Chapter 2. Consolidating data sharing members

Consider consolidating your data sharing groups so that they have fewer members. Especially consider consolidation if you originally added data sharing members to provide virtual storage constraint relief.

About this task

For DB2 10 and later releases, nearly all thread-related storage is above the bar in the DBM1 address space, which can provide significant virtual storage constraint relief.

Because of this location of the thread storage, you can potentially run more concurrent active threads for each Db2 subsystem or member. For data sharing configurations, this change might provide the opportunity to consolidate to fewer members.

For example, suppose that you have eight data sharing members, each of which is running 500 active threads, as shown in the following figure.

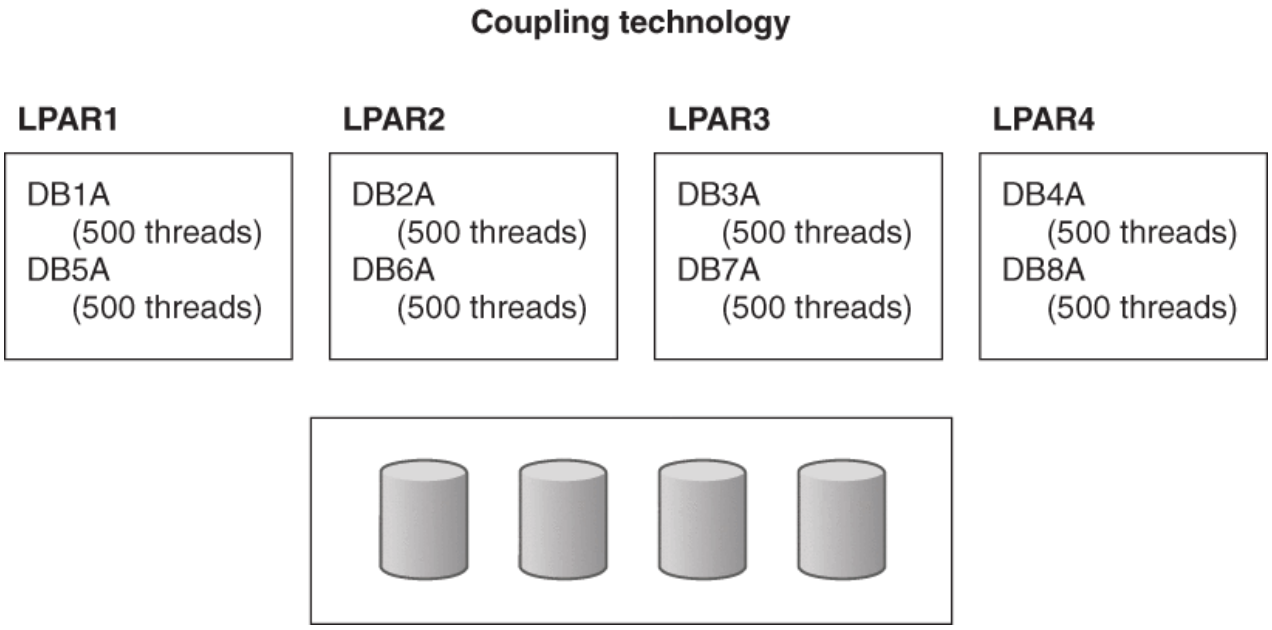


Figure 13. Data sharing configuration prior to consolidation

You might be able to consolidate to 4 members, each of which is running 2500 active threads, as shown in the following figure.

© Copyright IBM Corp. 1994, 2020

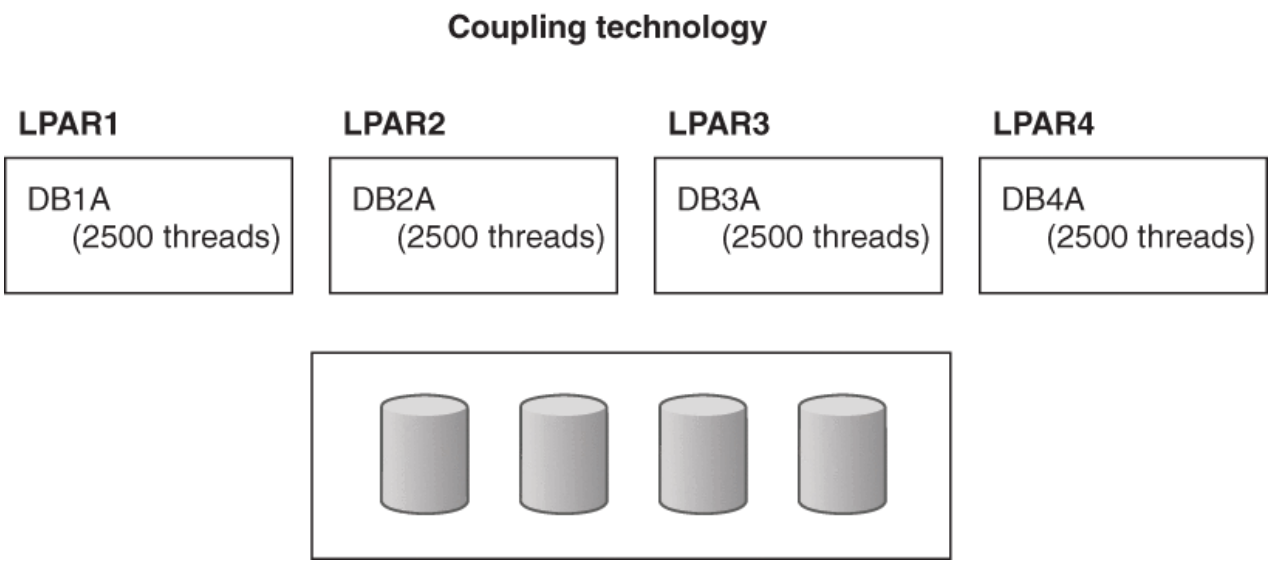


Figure 14. Data sharing configuration after consolidation

Recommendations:

- In configuring for high availability, ensure that enough spare capacity (such as CPU and memory) is available to absorb the work in case of a member outage (planned or unplanned). To accomplish this goal, one common method is to configure a 4-way data sharing group with 2 central processor complexes (CPCs) and 2 LPARs per CPC.
- Keep the members spread across more than one LPAR.
- To help with workload balancing, consider removing the same number of members from each LPAR. For example, suppose that you have four LPARs, each with two members, and you want to consolidate to fewer members. Ideally in this situation, you would remove one member from each LPAR (instead of changing LPAR 2 and LPAR 4 to have one member and leaving the other LPARs with two members). Such a symmetrical reduction can help you with real storage and CPU planning. Otherwise, more time planning is probably required to achieve the correct workload balance.

Procedure

To consolidate data sharing members:

1. Consider potential configuration changes that you might need to make when you consolidate data sharing members, and plan for these changes.
2. Redistribute the group workload:
   a) Move the work from one of the members that you plan to delete to one or more of the members that you plan to keep.
   b) After you move the work, monitor all members.

      If no issues arise, proceed to the next step. Otherwise, route any work back to the member that originally had that work, and handle any issues that arose during monitoring.
   c) Repeat steps a and b until all work is moved from the members that plan to delete to the members that you plan to keep.

   Example: Suppose that you have 8 members (member_1 through member_8) and you are consolidating to 4 members. In this scenario, assume that you plan to keep member_1 through member_4 and remove member_5 through member_8. First, move the work from member_5 to member_1 through member_4. Monitor all members. If any issues arise, route the work back to member_5 and address the issues. Otherwise, if no issues arise, move the work from member_6 to member_1 through member_4 and monitor all members. Repeat these same steps for member_7 and member_8.

3. Shut down all members of the group and run DSNJU003 to deactivate the members that you plan to delete.

   You can skip this step of deactivating members, if you are certain that those members are no longer needed.

   When you shut down all members of the data sharing group, make sure that all members shut down cleanly. Make sure that none of the members have incomplete units of recovery or outstanding utility jobs, and that none of the members have QC status.

   After you deactivate members, if you later determine that you need to bring back one of these members, you can reactivate those members.
4. When you are certain that the members that you deactivated are no longer needed, shut down all members in the group and run DSNJU003 to destroy the members that you want to delete from your data sharing group.

   When you shut down all members of the data sharing group, make sure that all members shut down cleanly. Make sure that none of the members have incomplete units of recovery or outstanding utility jobs, and that none of the members have QC status.

   After a member is destroyed, you can no longer reactivate it. That member no longer exists. If you later decide that you need an additional member, you need to add a new member.
5. Implement any configuration changes that you planned for in step 1.

Related tasks

Deactivating data sharing members
The first step in deleting a member from a data sharing group is to deactivate the member. Deactivating a member ensures that no new log data is created for that member.

Destroying data sharing members
Destroying a data sharing member permanently deletes it from the data sharing group. A destroyed member cannot be restored or reactivated.

Related reference

DSNJU003 (change log inventory) (Db2 Utilities)

Potential configuration changes when you consolidate data sharing members

Although the process of consolidating members requires only a couple of steps, you should spend time planning for this change. You might want to change various settings and configurations. If you are also consolidating the number of z/OS LPARs, you need to do additional planning.

When you consolidate data sharing members, consider making changes in the following areas:

- **Subsystem parameters:** If you plan to run more concurrent active threads for each member after consolidation, you might need to adjust certain subsystem parameter settings to accommodate the larger number of threads. For example, you might need to increase certain in-memory cache sizes, certain limit settings, or both.

  Consider adjusting the following subsystem parameters:

  - DSMAX
  - EDMPOOL
  - MAXRBLK
  - CTHREAD and MAXDBAT
  - PARAMDEG
  - MAXKEEPD
  - MAXTEMPS
  - SRTPOOL

  - OUTBUFF
  - LOBVALS
  - CHKFREQ or CHKLOGR and CHKMINS
  - XMLVALS
- **Active log data sets:** To accommodate the new consolidated member configuration, consider increasing the size and quantity of active log data sets.

  When more threads execute concurrently on each Db2 member, more Db2 logging is likely to occur on each member. Before consolidation, this logging was spread out over the active logs of multiple members.

  Making sure that you have enough active log space helps performance. Reads from the active log are faster than those from the archive log, especially if archives are on tape.

  You can dynamically add the active log data sets that you need while Db2 is operational. For instructions on how to add active log data sets, see Adding an active log data set to the active log inventory with the SET LOG command (Db2 Administration Guide).
- **Buffer pools:** If you plan to run more concurrent active threads for each member after consolidation, consider increasing the buffer pool sizes that are defined for each member. Also, consider adding new buffer pools.

  For example, suppose that member_1 had previously been running with a maximum of 400 concurrent active threads and a total buffer pool space allocation of 5 GB. Assume that after you consolidate your data sharing members, you increase the maximum number of threads for member_1 to 1200. In this case, the 5 GB of buffer pool space might no longer be sufficient, and threads might start to suffer I/O delays. In this case, you probably need to increase the 5 GB of buffer pool space to a larger size. The increase depends on whether the new threads predominantly access the same tables and indexes as the pre-existing threads. If the new threads predominantly access the same database objects, you probably do not need to increase the size of the buffer pools that much. If the new threads predominantly access different database objects, you probably need to increase the size of the buffer pools more.

  After you consolidate to fewer members, the total amount of buffer pool space that is allocated for all members is likely to be less than the amount of space that was allocated before consolidation.

  For more information about determining the appropriate buffer pool sizes, see Tuning database buffer pools (Db2 Performance).
- **Work file data sets:** If the number of concurrent active threads increases on a given subsystem after consolidation, consider allocating more temporary database storage. To create this extra space, create additional table spaces within the work file database.

  To find temporary space usage statistics, look at data section 4 of statistics record IFCID 2. You can use these statistics to determine existing consumption and to help project future usage.

  Also, consider the value of the MAXTEMPS_RID subsystem parameter. This parameter determines the amount of temporary storage in the work file database that can be used for RID overflow. If this parameter is set to a value other than NONE, more work file space might be used for each thread and therefore, you might need to define more work file space.

  You can use the MAXTEMPS subsystem parameter to regulate temporary space for each thread. You might need to increase the value of MAXTEMPS if MAXTEMPS_RID is set to a value other than NONE.

  Recommendation: If you need additional space, consider adding 32 KB work file data sets. Since DB2 9 became available, Db2 exploits 32 KB work file data sets more aggressively than in earlier releases.
- **SQA and CSA:** If you plan to run more concurrent active threads for each member after consolidation, consider increasing the sizes of the SQA (system queue area) and CSA (common service area). After you consolidate members, carefully monitor the CSA subpools and SQA subpools to determine the amount of extra space, if any, that you need.

  For example, a typical configuration might have a single Db2 subsystem on an LPAR that has 300 to 500 threads that are attached to Db2. Because of the advances in virtual storage constraint relief in DB2 10,

you can run 10 or more times the number of threads in a single LPAR. To handle this extra demand, depending on how many threads are running, you might need to increase the size of the SQA and CSA on the LPAR. This increase could be quite significant depending on several factors. For example, in the typical configuration described here, the size might need to increase to 50 MB. You need to consider your own configuration and monitor it carefully to determine an appropriate SQA and CSA size for your situation.

- **Real storage:** When Db2 members are consolidated, the demands on real storage can increase. For example, increases in buffer pools and the number of threads can increase the demands on real storage. Make sure that your consolidated system has enough real storage to prevent paging.

  To estimate how much real storage you need, look at the IFCID 225 records for the existing members prior to consolidation. Subtract the buffer pool and EDM pool numbers from the real storage number in IFCID 225. (This calculation assumes no auxiliary storage.) The resulting value approximates the amount of real storage that the threads are using. You can then use this amount to calculate the new value for the LPAR that is to contain the consolidated members.
- **Number of LPARs and central processors (CPs):** After you consolidate data sharing members, consider consolidating the number of LPARs and releasing the CPs from the LPARs that are being deleted. Reconfigure the surviving LPARs to use the additional processors. When deciding whether and how to consolidate LPARs, consider availability and failover.
- **CICS attachment facility:** You can configure the CICS attachment facility to attach to a group or to attach to a specific member. If you have this facility configured to attach to a specific member, you might need to change many different CICS regions to point to the one of the members of the new consolidated data sharing group.

  Group attachment has the advantage of easy movement of the CICS region around the sysplex. However, this method has the disadvantage of possibly allowing a Db2 member to be overloaded in DB2 Version 8 and DB2 9 if a CICS region attaches to a member that is does not have enough available storage. Attaching to specific member can guarantee improved availability.
- **DDF aliases:** When you consolidate data sharing members, you need to remove deleted members from any defined aliases.

  A data sharing group can have multiple location names, or aliases. Each alias can identify a subset of the members for remote connection to the data sharing group.

  Use the DSNJU004 utility to verify the DDF alias configuration. Then use the DSNJU003 utility to modify the DDF alias configuration to contain the remaining consolidated Db2 members that are needed for each alias.
- **TCP/IP addresses:** When members are removed from a data sharing group, review the TCP/IP configuration information for the remaining members to determine if you need to make any of the following changes:
  - Remove the IP address of any members that you are removing from the group.
  - Remove the resync port for any members that you are removing from the group.

  You need to consider both ports that are reserved in the TCPPARMS data set and ports in the DVIPA configuration.

  For instructions on how to customize the TCPPARMS data set, see Customizing the TCP/IP data sets or files (Db2 Installation and Migration). For instructions on how to customize the DVIPA configuration, see “Supported methods for specifying DVIPAs” on page 30.
- **Client CDB:** When you consolidate data sharing members, you might need to modify the configurations of remote clients that reference that data sharing group. If the client uses a DVIPA to access the data sharing group and its members, you do not need make changes. However, if the client uses IP addresses to access members of the data sharing group, you need to make changes. In this case, you need to remove from the client configuration the IP addresses of the members that are being deleted.
- **RBA rollover:** When you consolidate data sharing members, consider that you might need to reset the RBA on the consolidated members sooner than expected.

When you run more work on a single member, that member writes log records at a higher rate. Therefore, its end-of-log RBA value increases at a higher rate. For Db2 members that log at very high rates, their log RBA can eventually hit the maximum value. The maximum value is X'FFFFFFFFFFFF' for basic format with 6-byte RBA or LRSN values or X'FFFFFFFFFFFFFFFFFFFF' for extended format with 10-byte RBA or LRSN values. When this limit is approached, you need to shut down that member and reset its RBA value.

Chapter 3. Removing members from the data sharing group

One of the features of Db2 data sharing is incremental growth, which entails being able to add members to an existing group. However, a situation might occur in which you want to remove members from the group, either temporarily or permanently.

For example, assume that your group does the job it needs to do for 11 months of the year. However, a surge of additional work every December requires you to expand your capacity. You can quiesce some members of the group for those 11 months. Those members are "dormant" until you restart them. A quiesced member can also remain dormant indefinitely.

A quiesced member (whether you intend for it to be quiesced indefinitely or only temporarily) still appears in displays and reports. It appears in DISPLAY GROUP output with a status of QUIESCED.

To permanently remove a member, you can delete it from the data sharing group by following the procedure in “Deleting data sharing members” on page 24.

What data sets to keep when you quiesce a data sharing member

When you quiesce a member, you must keep the log data sets until they are no longer needed for recovery. Other members might need updates that are recorded on the quiesced member's log.

You must keep the BSDS, too, because it contains information that points to those log data sets. The BSDS is also needed for group restart. However, if you are confident that logs for the quiesced member are no longer necessary, because that member has been quiesced for a long time or is permanently quiesced, it is possible to delete the BSDS data sets. If you delete the BSDS data sets, expect the following message during group restart:

```
DSNR020I -DB2A csect-name START MEMBER DB1A, OR REPLY 'NO' or QUIESCED'
```

When you respond with QUIESCED, Db2 issues the following message:

```
DSNR030I -DB2A csect-name WILL CONTINUE WITHOUT THE DB1A MEMBER'S
LOG,
                          REPLY 'YES' OR 'NO'
```

In summary, you must do one of the following things to ensure that you have group restart capability:

- Keep the quiesced member's BSDS data sets (thus avoiding the preceding WTOR messages).
- Update your group restart procedure to ensure that operators know how to respond to the DSNR020I and DSNR030I messages.

Quiescing a data sharing member

Quiescing a member of a data sharing group temporarily or indefinitely removes the member from the group. A quiesced member can be restarted at any time.

Procedure

GUPI To quiesce a member of a data sharing group:

1. Stop the member you are going to quiesce.
   This example assumes that you want to quiesce member DB3A:

   ```
   -DB3A STOP DB2 MODE(QUIESCE)
   ```

2. From another member, enter the following commands:

© Copyright IBM Corp. 1994, 2020

```
DISPLAY GROUP
DISPLAY UTILITY (*) MEMBER(member-name)
DISPLAY DATABASE(*) RESTRICT
```

If there is no unresolved work, you can stop now. However, if you want to create an archive log, continue to the next step.

3. If there is unresolved work, or if you want to create a disaster recovery archive log (as in step 4), start the quiesced member with ACCESS(MAINT).

```
-DB3A START DB2 ACCESS(MAINT)
```

If there is unresolved work, resolve any remaining activity for the member, such as resolving indoubt threads, finishing or stopping utility work, and so on.

4. Optional: To create an archive log that can be sent to a disaster recovery site, archive the log for the member by entering the following command:

```
-DB3A ARCHIVE LOG
```

5. Stop the member again with MODE(QUIESCE).

```
-DB3A STOP DB2 MODE(QUIESCE)
```

GUPI

Related tasks

Disabling data sharing (Db2 Installation and Migration)

Related reference

-STOP DB2 (Db2) (Db2 Commands)
-DISPLAY GROUP (Db2) (Db2 Commands)
-DISPLAY UTILITY (Db2) (Db2 Commands)
-DISPLAY DATABASE (Db2) (Db2 Commands)
-START DB2 (Db2) (Db2 Commands)

Deleting data sharing members

You can permanently remove a member from a data sharing group by deleting the member. Delete a member only if you are sure that you will not need the member or its BSDS and log data sets again because a deleted member cannot be restarted.

About this task

Deleting data sharing members is a two-step process that requires deactivating the member and then destroying the member.

Deactivating data sharing members

The first step in deleting a member from a data sharing group is to deactivate the member. Deactivating a member ensures that no new log data is created for that member.

About this task

The member that is to be deleted must be quiesced before the surviving members are stopped. This requirement ensures that the BSDSs of the surviving members record the quiesced state of the member that is to be deleted.

A deactivated member can be reactivated at any time with the DSNJU003 RSTMBR control statement.

Procedure

To deactivate a data sharing member:

1. Ensure that the member that is to be deactivated does not have outstanding units of work or active utilities.
   a) Start the member with the ACCESS(MAINT) option.
   b) Check messages DSNR004I through DSNR008I to identify any active units of recovery (URs).
   c) Resolve any active URs.
   d) Issue the DISPLAY UTILITY(*) command to verify that the member has no active utilities.
   e) Issue the TERM UTILITY command to terminate any active utilities. Alternatively, active utilities can be restarted and allowed to complete normally.
   f) Issue the STOP DB2 MODE(QUIESCE) command to stop the subsystem.
2. Issue the DISPLAY GROUP command to verify that the member that is to be deactivated in the quiesced state.
3. Ensure that the BSDSs of all surviving members of the group indicate that the member that is to be deactivated is in the quiesced state.
4. Stop all members of the group.
5. Make a backup copy of all the BSDSs in the group.
6. Use the DSNJU003 DELMBR DEACTIV,MEMBERID=*x*control statement against all BSDSs of all members of the group, where *x* is the member ID of the member that is to be deactivated.

   The DSNJU003 change log inventory utility deactivates the member.
7. Restart the surviving members. All surviving members should be restarted so that they can record the updated status of the deactivated member.

Related reference

-START DB2 (Db2) (Db2 Commands)
-DISPLAY UTILITY (Db2) (Db2 Commands)
-TERM UTILITY (Db2) (Db2 Commands)
-STOP DB2 (Db2) (Db2 Commands)
-DISPLAY GROUP (Db2) (Db2 Commands)
DSNJU003 (change log inventory) (Db2 Utilities)

Destroying data sharing members

Destroying a data sharing member permanently deletes it from the data sharing group. A destroyed member cannot be restored or reactivated.

Before you begin

Before you destroy a data sharing member, you must deactivate it. If you want to destroy a member that was deactivated and then later reactivated and restarted, you must deactivate it again.

Procedure

To destroy a data sharing member:

1. Stop all members of the group.
2. Make a backup copy of all the BSDSs in the group.
3. Use the DSNJU003 DELMBR DESTROY,MEMBERID=x control statement against all BSDSs of all members of the group, where x is the member ID of the member that is to be destroyed.
4. Restart the surviving members. All surviving members should be restarted so that they can record the updated status of the destroyed member.

Results

The member is deleted, and its BSDS and logs are no longer needed.

If you want to reuse the member name of a destroyed member, you must add a new member to the data sharing group. The new member can have the same name and member ID as the member that you destroyed.

Related reference

-STOP DB2 (Db2) (Db2 Commands)

DSNJU003 (change log inventory) (Db2 Utilities)

-START DB2 (Db2) (Db2 Commands)

Restoring deactivated data sharing members

A member that has been deactivated (not destroyed) can be reactivated and restarted at any time.

About this task

No other members of the data sharing group must be stopped before restoring a deactivated member.

Procedure

To restore a deactivated member:

1. Use the DSNJU003 RSTMBR MEMBERID=*x* control statement against the BSDS of the member that is to be reactivated, where *x* is the member ID of the member.
2. Restart the member.
3. Restart all surviving members so that they can record the updated status of the reactivated member.

Related tasks

Starting inactive data sharing group members in a new Db2 release (Db2 Installation and Migration)

Related reference

DSNJU003 (change log inventory) (Db2 Utilities)

-START DB2 (Db2) (Db2 Commands)

Chapter 4. Communicating with data sharing groups

A data sharing group can be a powerful server in your client/server environment. The group can be part of a TCP/IP network, part of an SNA network, or part of a network that uses both protocols.

The group has a single-system image to requesting applications, whether requests arrive through TCP/IP or SNA. Queries can originate from any system or application that issues Structured Query Language (SQL) statements as a requester in the formats that are described by Distributed Relational Database Architecture™ (DRDA).

The distributed data facility (DDF) of Db2 uses TCP/IP and SNA to communicate with other Db2 subsystems. The DDF enables a Db2 subsystem to access data that is held by other database management systems. The DDF also enables the Db2 subsystem to make its own data accessible to other Db2 subsystems.

A data sharing group can support many more connections than a single member of the group can support. The DDF connections limit for a group is "*n* × 150,000", where *n* is the number of members in the group. Thus, a group with 16 members can support 2,400,000 DDF connections.

Related tasks

Connecting distributed database systems (Db2 Installation and Migration)

Ways to access data sharing groups

A *requester* is a client that manages connections and data access for client applications. Requesters can access data sharing groups in TCP/IP and SNA networks. Any product that supports the application requester protocols that are defined by DRDA can be a requester.

In this topic, the focus is on the following requesters:

- IBM Data Server Driver for JDBC and SQLJ.
- IBM Data Server Driver for ODBC and CLI.
- Db2 for z/OS requester support with built-in workload balancing support.
- Db2 Connect Server with connection concentrator and workload balancing support.

In a connection request, a client application specifies the Db2 location name of the data sharing group to which it wants to connect. The requester then maps this Db2 location name to an actual network element that identifies the member to which to connect. In the case of TCP/IP, this network element is either a domain name or an IP address. In the case of SNA, this network element is an LU name.

Recommendation: For TCP/IP access, use the group dynamic virtual IP address (DVIPA) and the common port to access the Db2 group from remote clients. For more information, see “Supported methods for specifying DVIPAs” on page 30

Mixed TCP/IP and SNA networks

The members of a data sharing group support both TCP/IP and SNA network protocols. However, when acting as requesters, members must be configured to communicate with other data sharing groups by using either TCP/IP or SNA. The DDF uses the group's communication database (CDB) to map each remote Db2 location name to either an IP address (for TCP/IP) or an LU name (for SNA). You can configure a Db2 location name for either TCP/IP or SNA, but not both. If you configure the same remote Db2 location name for both TCP/IP and SNA, only TCP/IP is used to communicate with the remote location.

© Copyright IBM Corp. 1994, 2020

TCP/IP access methods

Together, TCP and IP use the client/server model of communication to enable communication between computers and computer networks of the same or different types.

TCP/IP is a set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks. On the sending end, Transmission Control Protocol (TCP) manages the assembly of a message or file into smaller packets that are transmitted over a network. On the receiving end, TCP manages the re-assembly of those packets into the original message or file. Internet Protocol (IP) handles the routing of each packet, ensuring that packets reach the correct destination.

It is strongly recommended that when setting up the IP addresses to be used by a specific member of a data sharing group, that the IP address be a dynamic Virtual IP Address (DVIPA). By using such a capability, the specific Db2 member of the data sharing group is "assigned" its own IP address no matter where in the Sysplex the Db2 subsystem is started. This process will then permit successful automatic handling of indoubt units-of-work during subsystem restart or recovery especially when that member of the data sharing group is being accessed by requesters operating under the control of XA transaction managers and two-phase commit transaction coordinators.

When considering the use of a data sharing group IP address, you can only use a DVIPA which is being managed by the Sysplex Distributor. If you attempt to use a DVIPA that is not managed by the Sysplex Distributor or any other address as the group IP address, only one member of the data sharing group can be reached via that address. This process defeats the purpose of a common IP address that can be used to access any member of the data sharing group.

A TCP/IP requester can use one of several access methods to connect to a data sharing group:

Group access
A requester uses the group's dynamic virtual IP address (DVIPA) to make an initial connection to the Db2 location. Accessing the Db2 group using the group IP address is always successful if at least one member is started. If the Sysplex distributor is configured for the group IP address, the initial connection is based on the routing algorithm used by the Sysplex Distributor.

Alternatively, a requester can use a domain name, which is set up to return a list of member IP addresses. The requester will attempt to open a connection using every IP address in the list until the connection is successful.

If the requester is enabled for Sysplex workload balancing, the initial connection returns a list of members that are currently active and the capacity of each member to perform work. The requester uses this information to balance subsequent connections.

If the requester is enabled for connection concentrator support, connections return an updated list members that are currently active and the capacity of each member to perform work. The requester uses this information to route subsequent transactions to the member with the highest capacity. Transactions are processed across the group to provide the best utilization of work across the group.

Member-specific access
A requester uses location alias names, which specifies one or more members of the group, to make an initial connection to one of the members that is represented by the alias names. The member that receives the request returns a list of members that are currently active and able to perform work. The requester uses this information to send subsequent connection and query requests to available members that are represented by the location alias names. A Db2 for z/OS requester can also use the SYSIBM.IPLIST table to isolate requests to a subset of members using IPLIST or location alias names support to access remote data sharing groups.

Single-member access
A requester uses a member's DVIPA, actual IP address, or domain name to connect and send queries to a single member of the group. All connection and query requests from that requester are directed to the same member.

Group access

The preferred method of connecting to a data sharing group is to use dynamic virtual IP address (DVIPA) network addressing.

About dynamic virtual IP addressing (DVIPA)

DVIPA gives you the ability to assign a specific virtual IP address to a data sharing group and to each member of the group. This address is independent of any specific TCP/IP stack within the Parallel Sysplex. Even if a member is moved to another z/OS system, as in the case of a failure or maintenance, the member remains accessible and retains the same virtual IP address.

The TCP/IP Sysplex Distributor can play a role in DVIPA network addressing. If the Db2 group DVIPA is configured to distribute connections across all members that are listening to the Db2 DRDA PORT, the Sysplex Distributor can factor real-time information, such as member status and Quality of Service (QoS) data. This type of group DVIPA is called a *distributing DVIPA* or *DDVIPA*. This data, along with information that is obtained from the Workload Manager (WLM), ensures that the best member is chosen to serve each client connection request.

If the group DVIPA is configured to distribute connections by using the Sysplex Distributor, when the application specifies the Db2 location that maps to a group DVIPA, the Sysplex Distributor dispatches the requester's initial connection request to the active member with the most capacity. The requester ensures that workload balancing is invoked as follows:

- Requests are always routed to an active member, if at least one member is active.
- Requests are dynamically directed to those members that are using WLM system weights if you are using the default distribution method.

Sysplex workload balancing functionality on Db2 for z/OS servers provides high availability for client applications that connect directly to a data sharing group. Sysplex workload balancing functionality provides workload balancing and automatic client reroute capability. This support is available for applications that use Java™ clients (JDBC, SQLJ, or pureQuery®), or non-Java clients (ODBC, CLI, .NET, OLE DB, PHP, Ruby, or embedded SQL). For more information, see Java client direct connect support for high availability for connections to Db2 for z/OS servers (Db2 Application Programming for Java) and Non-Java client support for high availability for connections to Db2 for z/OS servers.

DVIPA and failure recovery

Using DVIPA is the most flexible way to be prepared for Db2 or system failure. If at least one member of a data sharing group is active and can perform work, connection attempts of requesters do not fail.

If a member suffers a failure or the underlying z/OS system suffers an outage, automation software such as z/OS Automatic Restart Manager (ARM) can restart the member on a different z/OS system. When this type of restart happens, the member-specific DVIPA is also moved to the new system and automatic recovery of any indoubt threads is performed, thereby enabling requesters to access the member on the new z/OS system.

Requirements and considerations

- If you plan to use DVIPA network addressing, check with your network administrator to determine whether DVIPA can be configured for the Parallel Sysplex. DVIPA network addressing is required if you are using clients that support Sysplex workload balancing. You also must use DVIPA network addressing to support remote XA connections.
- Ensure that the following conditions exist before making a data sharing group available for group access that uses DVIPA network addressing:
  - All members of the group are configured with DVIPAs before starting the DDF on any member of the group.
  - All members of the group are configured with DVIPAs before remote connections are made to any member of the group.
  - All members are running OS/390® Version 2, Release 10 or later.

- Before moving to DVIPA network addressing from another access method, have all members check for the existence of indoubt threads. If any indoubt threads exist, resolve them before moving. To check for indoubt threads, use the DISPLAY THREAD command.
- DVIPA network addressing requires you to define multiple DVIPAs, one for the data sharing group and one for each member of the group.
  - The group DVIPA is common to all members of the group, and it is used to make the initial connection to a member of the group. The group DVIPA should be owned and managed by the Sysplex Distributor, which provides workload balancing among members during new connection request processing.
  - Every member of the group needs a unique DVIPA, which is used to directly connect to a particular member after the initial connection. Configure DVIPAs for each member of the data sharing group that can be started.
- Db2 supports two methods for specifying DVIPAs: the BSDS (INADDR_ANY) method and the BINDSPECIFIC method. Each method has different characteristics that might make one method more appropriate for your environment than the other method. See "Supported methods for specifying DVIPAs" on page 30 to determine which method is best for your environment.

Supported methods for specifying DVIPAs

Db2 for z/OS supports two methods for specifying DVIPAs: the BSDS (INADDR_ANY) method and the BINDSPECIFIC method; however, only one method might be viable for your particular environment.

Important: These methods are mutually exclusive. Use only the BSDS (INADDR_ANY) method or the BINDSPECIFIC method to specify DVIPAs. Use the information in this topic to determine which method is appropriate for your environment.

Which method should you choose?

The preferred method for specifying DVIPAs is the BSDS (INADDR_ANY) method. This method provides greater flexibility with handling network addressing changes without impacting your applications. However, there are certain situations and restrictions that require to use one method instead of the other.

- You must use the BINDSPECIFIC method if within your TCP/IP configuration a DDVIPA is configured with no port values specified in its definition so that it can be shared by many applications, and all of those applications bind their ports to this DDVIPA. In this type of configuration, in which the DDVIPA dynamically learns the ports it is to manage, the DDVIPA cannot be used exclusively as the Db2 group IP address in its BSDS as the GRPIPV4 address or GRPIPV6 address.
- You must use the BSDS (INADDR_ANY) method if you're using IBM Db2 Analytics Accelerator.

Overview of the BSDS (INADDR_ANY) configuration

With the BSDS (INADDR_ANY) method, you specify DVIPAs in the bootstrap data set that use by using the DSNJU003 utility.

The following figure shows an example BSDS (INADDR_ANY) configuration for setting up a three-member data sharing group. In this example:

- Vx represents the group DVIPA, and V1, V2, and V3 represent the members' DVIPAs.
- The group DVIPA and the member-specific DVIPAs are defined on each TCP/IP stack.
- The SHAREPORT option is required only when multiple members are started on the same LPAR or TCP/IP stack. Without this option, TCP/IP does not allow multiple listeners on the DRDA port.
- The VIPADYNAMIC statements are structured as follows:
  - The group DVIPA must be defined with the VIPADEFINE and VIPADISTRIBUTE statements on one of the TCP/IP stacks within the sysplex, and the system where the definition is activated will be considered the primary system of the Sysplex Distributor.
  - The group DVIPA can also be defined with the VIPABACKUP statement on the TCP/IP stacks for DVIPA takeover. Note that the VIPABACKUP statements are coded with the MOVEABLE IMMEDIATE keywords, and that the VIPADISTRIBUTE statements are also specified on the backup TCP/IP stacks.

This allows for the group DVIPA to be activated on one of the backup stacks if it is not active anywhere else in the Sysplex. For example, if z/OS®-1 has not been started when z/OS-2 or z/OS-3 starts, then group DVIPA is activated on one of the backup stacks.

– To allow for failover, the member-specific DVIPAs are defined with the VIPARANGE statement on all TCP/IP stacks.

z/OS - 1

```
PORT
  446 TCP DB2ADIST SHAREPORT
  446 TCP DB2BDIST SHAREPORT
  446 TCP DB2CDIST SHAREPORT
5001 TCP DB2ADIST
5002 TCP DB2BDIST
5003 TCP DB2CDIST
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPADEFINE MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
     PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS - 2

```
PORT
  446 TCP DB2ADIST SHAREPORT
  446 TCP DB2BDIST SHAREPORT
  446 TCP DB2CDIST SHAREPORT
5001 TCP DB2ADIST
5002 TCP DB2BDIST
5003 TCP DB2CDIST
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPABACKUP 1 MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
     PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS - 3

```
PORT
  446 TCP DB2ADIST SHAREPORT
  446 TCP DB2BDIST SHAREPORT
  446 TCP DB2CDIST SHAREPORT
5001 TCP DB2ADIST
5002 TCP DB2BDIST
5003 TCP DB2CDIST
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPABACKUP 2 MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
     PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

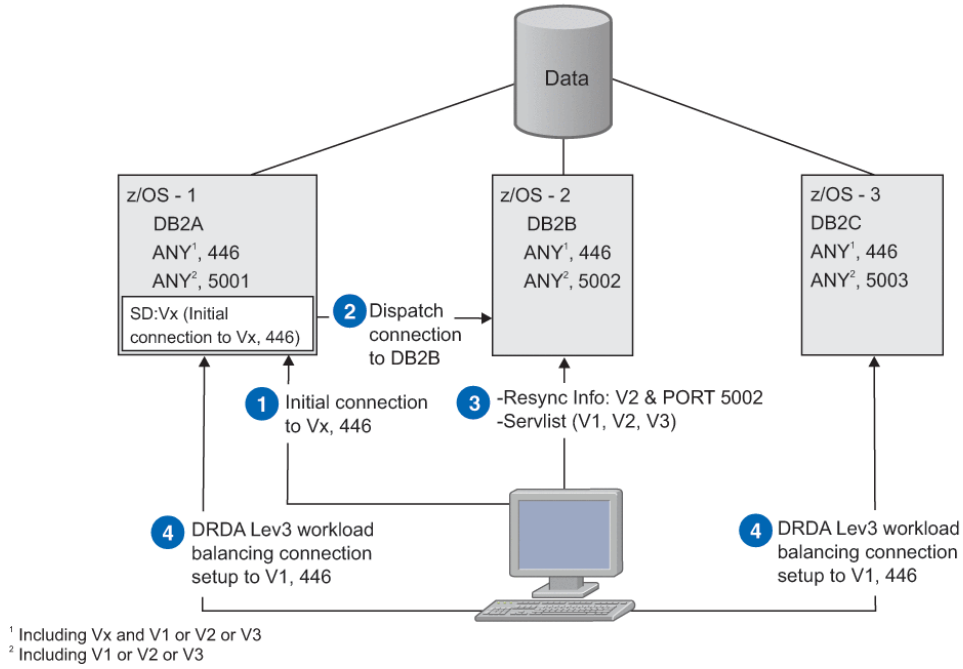


Figure 15. Example TCP/IP network configuration (INADDR_ANY IPv6 dynamic virtual IP addressing)

1. The initial connection uses the group dynamic Virtual IP Addresses (Vx). Port 446 is identified as the DRDA port in each member's PORT statements.
2. The Sysplex Distributor dispatches the initial connection request to the member with the lightest workload (DB2B).
3. Resynchronization information and a list of members in the data sharing group are returned to the requester.
4. Database-level workload balancing connections are established.

See for instructions.

Overview of the BINDSPECIFIC configuration

In a BINDSPECIFIC configuration, you specify DVIPAs on the PORT statement of the TCP/IP profile by using the BIND keyword. A Db2 data sharing group member accepts connection requests only if the IP

address that is used to target the member is either the member DVIPA (IPv4 or IPv6, but not both), or the group DVIPA (IPv4 or IPv6).

The following figure shows an example BINDSPECIFIC configuration for setting up a three-member data sharing group to route DVIPA requests and perform workload balancing across the members. In this example:

- Vx represents the group DVIPA, and V1, V2, and V3 represent the members' DVIPAs.
- The group DVIPA and the member-specific DVIPAs are defined on each TCP/IP stack.
- The SHAREPORT option is required only when multiple members are started on the same LPAR or TCP/IP stack; without this option, TCP/IP does not allow multiple listeners on the DRDA port.
- The BIND (SPECIFIC) option restricts the clients to use only the group DVIPA or member DVIPA to connect to Db2.
- The VIPADYNAMIC statements are structured as follows:
  - The group DVIPA must be defined with the VIPADEFINE and VIPADISTRIBUTE statements on one of the TCP/IP stacks within the sysplex. The system where the definition is activated is considered the primary system of the Sysplex Distributor.
  - The group DVIPA must be defined with the VIPABACKUP statement on the TCP/IP stacks for DVIPA takeover. Note that the VIPABACKUP statements are coded with the MOVEABLE IMMEDIATE keywords, and that the VIPADISTRIBUTE statements are also specified on the backup TCP/IP stacks. These statements allow for the group DVIPA to be activated on one of the backup stacks if it is not active anywhere else in the Sysplex. For example, if z/OS®-1 has not been started when z/OS-2 or z/OS-3 launch, then group DVIPA is activated on one of the backup stacks.
  - To allow for failover, the member-specific DVIPAs are defined with the VIPARANGE statement on all TCP/IP stacks.

z/OS - 1

```
PORT
  446 TCP DB2ADIST SHAREPORT BIND Vx
  446 TCP DB2BDIST SHAREPORT BIND Vx
  446 TCP DB2CDIST SHAREPORT BIND Vx
5001 TCP DB2ADIST BIND V1
5002 TCP DB2BDIST BIND V2
5003 TCP DB2CDIST BIND V3
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPADEFINE MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
    PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS - 2

```
PORT
  446 TCP DB2ADIST SHAREPORT BIND Vx
  446 TCP DB2BDIST SHAREPORT BIND Vx
  446 TCP DB2CDIST SHAREPORT BIND Vx
5001 TCP DB2ADIST BIND V1
5002 TCP DB2BDIST BIND V2
5003 TCP DB2CDIST BIND V3
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPABACKUP 1 MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
    PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS - 3

```
PORT
  446 TCP DB2ADIST SHAREPORT BIND Vx
  446 TCP DB2BDIST SHAREPORT BIND Vx
  446 TCP DB2CDIST SHAREPORT BIND Vx
5001 TCP DB2ADIST BIND V1
5002 TCP DB2BDIST BIND V2
5003 TCP DB2CDIST BIND V3
VIPADYNAMIC
  VIPARANGE DEFINE INTFV1 V1/128
  VIPARANGE DEFINE INTFV2 V2/128
  VIPARANGE DEFINE INTFV3 V3/128
  VIPABACKUP 2 MOVE IMMED INTFVx Vx
  VIPADISTRIBUTE DEFINE INTFVx
    PORT 446 DESTIP ALL
ENDVIPADYNAMIC
```

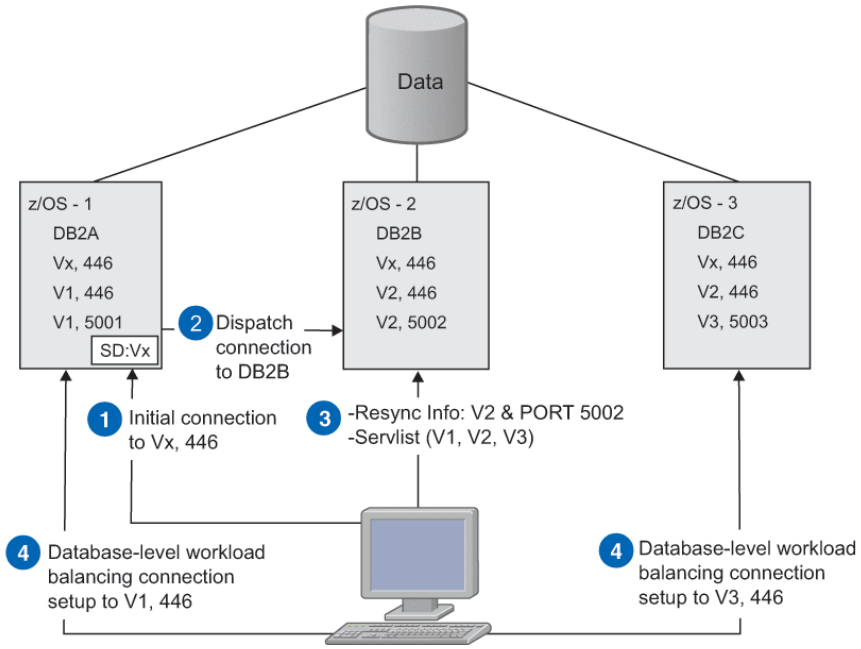


Figure 16. Example TCP/IP network configuration (BINDSPECIFIC IPv6 dynamic virtual IP addressing)

1. The initial connection uses the group dynamic virtual IP addresses (Vx). Port 446 is identified as the DRDA port in each member's PORT statements.
2. The Sysplex Distributor dispatches the initial connection request to the member with the lightest workload (DB2B).
3. Resynchronization information and a list of members in the data sharing group are returned to the requester.
4. Database-level workload balancing connections are established.

See for instructions.

Related concepts

Using dynamic VIPAs (z/OS Communications Server: IP Configuration Guide)

Java client direct connect support for high availability for connections to Db2 for z/OS servers (Db2 Application Programming for Java)

Customization of IBM Data Server Driver for JDBC and SQLJ configuration properties (Db2 Application Programming for Java)

Sysplex distributor (z/OS Communications Server: IP Configuration Guide)

Related reference

Non-Java client support for high availability for connections to Db2 for z/OS servers

Configuring TCP/IP group access by using the BSDS (INADDR_ANY) method

The bootstrap data set (BSDS) method is the preferred method for configuring TCP/IP communications for a Db2 data sharing group. Use this method to specify dynamic virtual IP addresses (DVIPAs) in the bootstrap data set by using the DSNJU003 utility.

Before you begin

You should be familiar with general instructions for configuring TCP/IP communications for Db2. See Connecting systems with TCP/IP (Db2 Installation and Migration).

The steps in this topic require you to specify several types of VIPADYNAMIC statements. For more information, see TCP/IP profile (PROFILE.TCPIP) and configuration statements.

About this task

This task describes how to define the DVIPAs that are needed by a Db2 data sharing group by specifying those DVIPAs in the BSDS. This task uses IPV4 addressing to illustrate the configuration process, but IPV6 addresses are also supported.

A member of a data sharing group can be configured to have a maximum of two IP addresses: a group IP address and a member IP address. Each pair can be of type IPV4 or IPV6, or you can specify a pair of both IPV4 and IPV6 addresses.

A group IP address

A group IP address must be the same for all members of a data sharing group. You define a group IP address by specifying a GRPIPV4 address, a GRPIPV6 address, or both in the BSDS. The instructions in this topic show how to define a group IP address as a VIPADISTRIBUTE must be the same for all members of a data sharing group. You define a group IP address by specifying a GRPIPV4 address, a GRPIPV6 address, or both in the BSDS. The instructions in this topic show how to define a group IP address as a VIPADISTRIBUTE VIPA address, which is also referred to as a *sysplex distributor address* or a *distributing DVIPA* (DDVIPA). Using a sysplex distributor address is the preferred way to define a group IP address.

Note: You can define a separate DDVIPA independent of the DDVIPA defined to Db2 as the group's IP address to also distribute connections to the member ports of a data sharing group, possibly with a different distributing method; however, Db2 will have no knowledge of this IP address.

Restriction: In some TCP/IP configurations, a DDVIPA is configured with no port values specified in its definition so that it can be shared by many applications, and all of those applications bind their ports to this DDVIPA. In these types of configurations in which the DDVIPA dynamically learns the ports it is to manage, the DDVIPA cannot be used exclusively as the Db2 group IP address in its BSDS as the GRPIPV4 address or GRPIPV6 address. For TCP/IP configurations in which a DDVIPA is used by many applications, you must use the BINDSPECIFIC method to configure TCP/IP group access.

A member IP address

A member IP address must be unique to each member of a data sharing group. You define a member IP address by specifying an IPV4 address, an IPV6 address, or both in the BSDS. The instructions in this topic show how to define a member IP address as a VIPARANGE IP address, which is the preferred way to define member IP addresses. A TCPIP VIPARANGE address will not physically exist until Db2is started with its IPV4 or IPV6 set to a VIPARANGE address value.

Important:

- The BSDS and BINDSPECIFIC methods are mutually exclusive. If any BIND specifications exist on any Db2 server port entries of the PORT reservation statement of the currently active PROFILE.TCPIP, connection requests will fail.
- This task is typically performed by or in consultation with a network communications administrator.

Procedure

1. Reserve the SQL port and resynchronization ports for each member of the data sharing group. If your environment is set up to receive SSL connection requests, also reserve the secure SQL port.

Edit the PROFILE.TCPIP file for each LPAR in the sysplex where a member of the data sharing group might be started and add entries for these ports to the PORT reservation statement as shown in the following example:

```
PORT
        446 TCP DB2A SHAREPORT  /* DRDA port for SQL requests */
        448 TCP DB2A SHAREPORT  /* Secure SQL port for SSL requests */
       5001 TCP DB2A            /* Resync port; unique for each member */
        446 TCP DB2B SHAREPORT
        448 TCP DB2B SHAREPORT
       5002 TCP DB2B
        446 TCP DB2C SHAREPORT
        448 TCP DB2C SHAREPORT
       5003 TCP DB2C
```

In this example, ports are reserved for a data sharing group that includes members DB2A, DB2B, and DB2C. Three ports are reserved for each of these members:

- Port 446 is the DRDA port number for initially receiving SQL requests (TCPPORT) and also the port where RESTful service requests are processed.
- Port 448 is the secure DRDA port number for receiving SSL connection requests (SECPORT). You need to reserve this port only if you configured Db2 to accept connection requests that use the SSL protocol. This port can also be used to accept RESTful service requests over connections that use the SSL protocol.
- Ports 5001, 5002, and 5003 are the resynchronization ports (RESPORT). The resynchronization ports must be unique for each member of the data sharing group.

If more than one member is started on the same LPAR, you must specify the SHAREPORT parameter on the DRDA port statement and on the secure SQL port statement if it is used. The SHAREPORT parameter enables multiple listeners on the same port.

2. Define a range of member IP addresses for use by the data sharing group.

   In the PROFILE.TCPIP file for each LPAR in the sysplex where a member of the data sharing group can be started, use the VIPARANGE statement to specify the range of IP addresses that will be available to the data sharing group members. For example:

   ```
   VIPARANGE DEFINE 255.255.255.255 10.1.1.2
   VIPARANGE DEFINE 255.255.255.255 10.1.1.3
   VIPARANGE DEFINE 255.255.255.255 10.1.1.4
   ```

   Because this data sharing group includes three members, three VIPARANGE statements are required.

3. Designate an LPAR of the sysplex to be the primary sysplex distributor for the group IP address. This LPAR's activated PROFILE.TCPIP VIPADYNAMIC statements must include the VIPADISTRIBUTE statement and a related VIPADEFINE statement for the group IP address. For example,:

   ```
   VIPADEFINE 255.255.255.255 10.1.1.1
   VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
   ```

   In this example:

   - The VIPADEFINE statement designates the LPAR where this statement is activated to be the primary sysplex distributor of the sysplex.
   - The VIPADISTRIBUTE statement specifies that 10.1.1.1 is to be a distributing IP address and includes in its definition the DRDA SQL/REST port numbers and secure port numbers that you reserved in a previous step.
   - DESTIP ALL specifies that all TCP/IP stacks that are active in the sysplex and that contain servers listening on any of the specified PORTs can receive the connection requests that target the IP address.

4. Create and run a change log utility job to update the members' DDF record to specify the IP addresses.

For example, the following DDF statement adds the ports and IP addresses for the DB2B member, which is assigned to port 5002:

```
//DB2B        EXEC PGM=DSNJU003
//SYSIN  DD *

DDF LOCATION=GROUP1,PORT=446,SECPORT=448,RESPORT=5002,IPV4=10.1.1.2,GRPIPV4=10.1.1.1
```

Example

The following example TCPIP.PROFILE PORT statements show how you might define the ports and IP addresses to configure TCP/IP for group access by using the BSDS method. The values that are used in these examples match the values that are used in the procedure section of this topic.

z/OS LPAR 1: Primary sysplex distributor

```
PORT
   446 TCP DB2A SHAREPORT
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A
   446 TCP DB2B SHAREPORT
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B
   446 TCP DB2C
   448 TCP DB2C
 5003 TCP DB2C
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPADEFINE 255.255.255.255 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS LPAR2: Backup sysplex distributor 1

```
PORT
   446 TCP DB2A SHAREPORT
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A
   446 TCP DB2B SHAREPORT
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B
   446 TCP DB2C
   448 TCP DB2C
 5003 TCP DB2C
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPABACKUP 2 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS LPAR3: Backup sysplex distributor 2

```
PORT
   446 TCP DB2A SHAREPORT
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A
   446 TCP DB2B SHAREPORT
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B
   446 TCP DB2C
   448 TCP DB2C
 5003 TCP DB2C
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPABACKUP 1 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

Related concepts

Using dynamic VIPAs

Sysplex distributor

Related reference

TCP/IP profile (PROFILE.TCPIP) and configuration statements

Configuring TCP/IP group access by using the BINDSPECIFIC method

The BINDSPECIFIC method for configuring TCP/IP group access involves assigning dynamic virtual IP addresses (DVIPAs) to listener ports by using the BIND parameter in the PORT statement of the TCP/IP profile.

Before you begin

You should be familiar with general instructions for configuring TCP/IP communications for Db2. See Connecting systems with TCP/IP.

The steps in this topic require you to specify several types of VIPADYNAMIC statements. For more information, see TCP/IP profile (PROFILE.TCPIP) and configuration statements.

About this task

This task describes how to define the DVIPAs that are needed by a Db2 data sharing group by specifying those DVIPAs on the PORT statement of the TCP/IP profile, and associating those DVIPAs with ports by using the BIND parameter. For example purposes, the data sharing group in this task consists of three members. One of the DVIPAs will function as a *sysplex distributor*, which is also referred to as a *distributing DVIPA* (DDVIPA). This task uses ipv4 addressing to illustrate the configuration process, but ipv6 addresses are also supported.

- Using the BSDS (INADDR_ANY) method is the preferred way to configure TCP/IP group access. However, if you need to restrict clients to using specific IP addresses to target Db2, use the BINDSPECIFIC method. The BINDSPECIFIC method ensures that the only IP addresses that can be used to target any member of the data sharing group are member-specific IP addresses (DVIPAs) or the group IP address (DDVIPA). No other IP address, including those that can reach an LPAR where a Db2 subsystem resides, can be used. The BINDSPECIFIC method provides this level of control by binding its TCP/IP listener sockets to the group and member IP addresses rather than to the INADDR_ANY address.
- The BINDSPECIFIC and BSDS methods are mutually exclusive; use only one of these methods to define a DVIPA.

  Important: The BINDSPECIFIC method requires that no IP addresses are specified in the BSDS. If any IP addresses are specified in the BSDS, the DDF TCPIP listener fails to start, and Db2 issues message DSNL512I.
- This task is typically performed by or in consultation with a network communications administrator.

Procedure

1. Reserve the SQL port and resynchronization ports for each member of the data sharing group, and bind IP addresses to each of these port statements. If your environment is set up to receive SSL connection requests, also reserve the secure SQL port.

   Edit the PROFILE.TCPIP file for each LPAR in the sysplex where a member of the data sharing group could be started, and add entries for these ports and IP addresses to the PORT reservation statement as shown in the following example:

```
PORT
    446 TCP DB2A SHAREPORT BIND 10.1.1.1   /* DRDA port for SQL requests */
    448 TCP DB2A SHAREPORT                 /* Secure SQL port for SSL requests */
   5001 TCP DB2A BIND 10.1.1.2             /* Resync port; unique for each member */
    446 TCP DB2B SHAREPORT BIND 10.1.1.1
    448 TCP DB2B SHAREPORT
   5002 TCP DB2B BIND 10.1.1.3
    446 TCP DB2C SHAREPORT BIND 10.1.1.1
```

```
  448 TCP DB2C SHAREPORT
 5003 TCP DB2C BIND 10.1.1.4
```

In this example, ports and IP addresses are reserved for a data sharing group that includes members DB2A, DB2B, and DB2C. The following ports and IP addresses are reserved for each of these members:

- Port 446 is the DRDA port number for initially receiving SQL requests (TCPPORT) and also the port where RESTful service requests are processed. 10.1.1.1 is the listener address that is assigned to port 446 and that will function as the sysplex distributor (DDVIPA) and is then also known as the group IP address.
- Ports 5001, 5002, and 5003 are the resynchronization ports (RESPORT). 10.1.1.2, 10.1.1.3, and 10.1.1.4 are the IP addresses associated with each of these ports, which function as the member IP addresses. The RESPORTs and IP addresses must be unique for each member of the data sharing group.
- Port 448 is the secure DRDA port number for receiving SSL connection requests (SECPORT). You need to reserve this port only if you configured Db2 to accept connection requests that use the SSL protocol. This port can also be used to accept RESTful service requests over connections that use the SSL protocol.

Note: Only the main PORT (TCPPORT) and the member RESPORTs require the BIND specification.

If more than one member is started on the same LPAR, you must specify the SHAREPORT parameter on the DRDA port statement and on the secure SQL port statement if it is used. The SHAREPORT parameter permits multiple listeners on the same port within the same LPAR TCP/IP configuration.

2. Designate an LPAR of the sysplex to be the primary sysplex distributor for the group IP address. This LPAR's activated PROFILE.TCPIP VIPADYNAMIC statements must include the VIPADISTRIBUTE statement and a related VIPADEFINE statement for the group IP address. For example:

```
VIPADEFINE 255.255.255.255 10.1.1.1
VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
```

In this example:

- The VIPADEFINE statement designates the LPAR where this statement is activated to be the primary sysplex distributor of the sysplex.
- The VIPADISTRIBUTE statement specifies that 10.1.1.1 is to be a distributing IP address and includes in its definition the DRDA SQL/REST port numbers and secure port numbers that you reserved in a previous step.
- DESTIP ALL specifies that all TCP/IP stacks that are active in the sysplex and that contain servers that are listening on any of the specified PORTs can receive the connection requests that target the IP address.

3. Define a range of member IP addresses for use by the data sharing group.

In the PROFILE.TCPIP file for each LPAR in the sysplex where a member of the data sharing group can be started, use the VIPARANGE statement to specify the range of IP addresses that will be available to the data sharing group. The IP addresses that you specify here must match the IP addresses in the BIND specifications of the RESPORTs in step 1. For example:

```
VIPARANGE DEFINE 255.255.255.255 10.1.1.2
          VIPARANGE DEFINE 255.255.255.255 10.1.1.3
          VIPARANGE DEFINE 255.255.255.255 10.1.1.4
```

Because this data sharing group includes three members, three VIPARANGE statements are required.

4. Designate one or more LPARs of the sysplex to provide backup sysplex distributor capability for the group IP address. The LPARs' activated PROFILE.TCPIP VIPADYNAMIC statements must include a VIPADISTRIBUTE and a related VIPABACKUP statement of the group IP address. For example:

```
VIPABACKUP 1 10.1.1.1
VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
```

In this example:

- The VIPABACKUP statement designates this LPAR as a backup sysplex distributor. The number 1 indicates the priority of this member.
- The VIPADISTRIBUTE statement matches the statement for the primary member.

Example

The following example TCPIP.PROFILE PORT statements show how you might define the ports and IP addresses to configure TCP/IP for group access by using the BINDSPECIFIC method. The values that are used in these examples match the values that are used in the procedure section of this topic.

z/OS LPAR1: Primary sysplex distributor

```
PORT
   446 TCP DB2A SHAREPORT BIND 10.1.1.1
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A BIND 10.1.1.2
   446 TCP DB2B SHAREPORT BIND 10.1.1.1
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B BIND 10.1.1.3
   446 TCP DB2C SHAREPORT BIND 10.1.1.1
   448 TCP DB2C SHAREPORT
 5003 TCP DB2C BIND 10.1.1.4
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPADEFINE 255.255.255.255 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS LPAR2: Backup sysplex distributor 1

```
PORT
   446 TCP DB2A SHAREPORT BIND 10.1.1.1
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A BIND 10.1.1.2
   446 TCP DB2B SHAREPORT BIND 10.1.1.1
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B BIND 10.1.1.3
   446 TCP DB2C SHAREPORT BIND 10.1.1.1
   448 TCP DB2C SHAREPORT
 5003 TCP DB2C BIND 10.1.1.4
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPABACKUP 2 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

z/OS LPAR3: Backup sysplex distributor 2

```
PORT
   446 TCP DB2A SHAREPORT BIND 10.1.1.1
   448 TCP DB2A SHAREPORT
 5001 TCP DB2A BIND 10.1.1.2
   446 TCP DB2B SHAREPORT BIND 10.1.1.1
   448 TCP DB2B SHAREPORT
 5002 TCP DB2B BIND 10.1.1.3
   446 TCP DB2C SHAREPORT BIND 10.1.1.1
   448 TCP DB2C SHAREPORT
 5003 TCP DB2C BIND 10.1.1.4
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.255 10.1.1.2
  VIPARANGE DEFINE 255.255.255.255 10.1.1.3
  VIPARANGE DEFINE 255.255.255.255 10.1.1.4
  VIPABACKUP 1 10.1.1.1
  VIPADISTRIBUTE DEFINE 10.1.1.1 PORT 446 448 DESTIP ALL
ENDVIPADYNAMIC
```

Related concepts

Using dynamic VIPAs

Sysplex distributor

Related reference

TCP/IP profile (PROFILE.TCPIP) and configuration statements

DNS network addressing

A domain name server (DNS) provides standard name resolution services for IP applications in a Parallel Sysplex cluster, converting domain names into IP addresses.

If, in its connection request, an application specifies a Db2 location name that maps to a group domain name, the requester queries the DNS with a call to *gethostbyname*. Either the DNS resolves the name to a member's IP address, or it resolves the name to a list of member IP addresses. The DNS returns the IP address or list of IP addresses to the requester, which then uses it to make an initial connection to a member of the data sharing group.

If your DNS supports it, and the Sysplex Distributor is not used to manage the group's DVIPA, configure the DNS to return a list of member IP addresses (member DVIPAs) instead of returning a single IP address. The list includes member IP addresses and ports, and a weight for each active member that indicates the member's current capacity. The requester uses the returned information to connect to the member with the most capacity. If you use the Sysplex Distributor to manage the DVIPA of the group, configure the DNS to return the group DVIPA.

Recommendation: If your DNS supports it, configure the DNS to return a list of member IP addresses instead of returning a single IP address. Because a DNS does not know whether a member is active when it returns the member's IP address to the requester, the DNS might return the IP address of an inactive member. Using this IP address on a subsequent connection request causes the request to fail. In contrast, if you configure the DNS to return a list of member IP addresses, the connection request can be retried using another IP address on the list. When the DNS retries the connection request, it will run through the list once, searching for the active member.

Example of group access configuration

Applications that use group access use the group's Db2 location name (DB2A) to direct connection requests to the group.

The configuration in the figure below is an example of group access that uses DNS network addressing

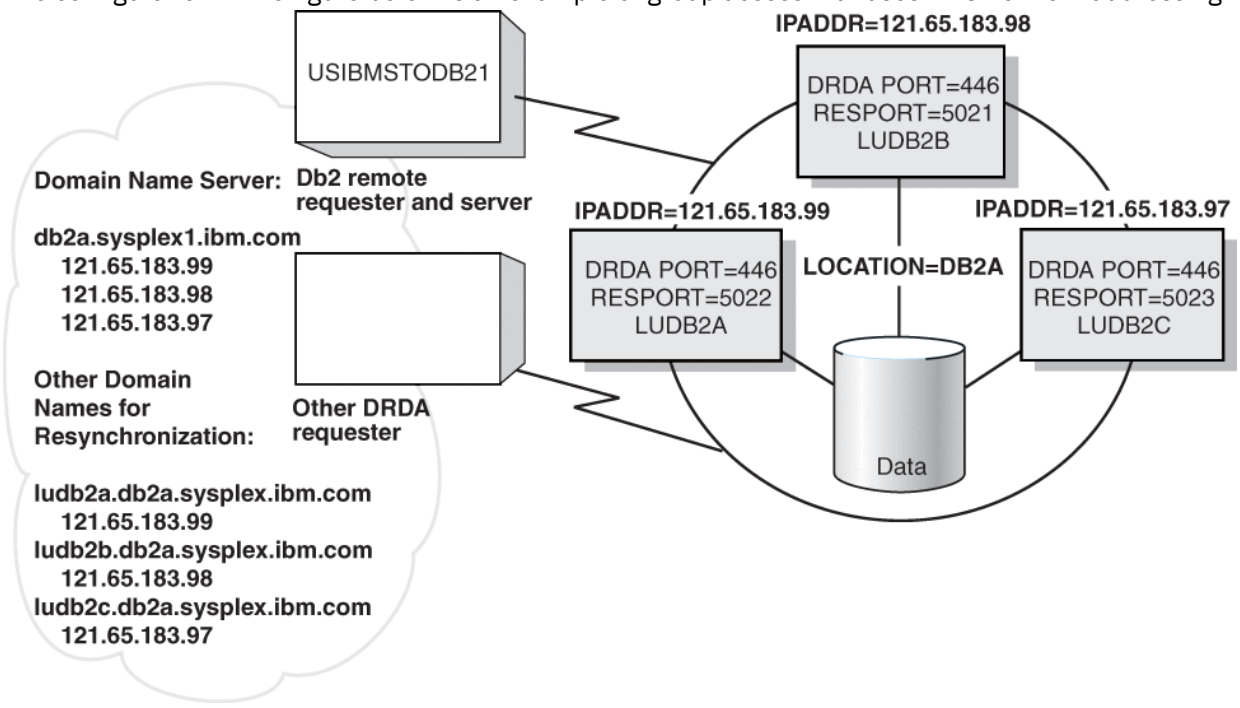


Figure 17. Example TCP/IP network configuration (DNS network addressing)

Note: When a Db2 for z/OS system is a member of a data sharing group, Db2 cannot be used as a DRDA XA server. As a result, the XA Transaction Manager cannot directly access the Db2 for z/OS DRDA server. Instead, you must configure the DRDA requester system in such a way that it insulates the Db2 for z/OS DRDA server from receiving XA calls from the XA Transaction Manager.

Member-specific access

A Db2 location name represents all members of a data sharing group. In contrast, member-specific access uses location aliases that map to the domain names or IP addresses of only a subset of the members of the data sharing group.

A requestor can use location aliases to bypass default TCP/IP workload balancing and to establish connections with one or more members. Workload is balanced among members at the discretion of the requestor.

With member-specific access, a requester uses a location alias to make an initial connection to one of the members that is represented by the alias. The member that receives the connection request works in conjunction with Workload Manager (WLM) to return a list of members that are currently active and able to perform work. The list includes member IP addresses and ports, and a weight for each active member that indicates the current capacity of the member. The IP addresses are the member-specific alias IP addresses, if they exist, or the member-specific IP addresses. The requester uses this information to connect to the member or members with the most capacity that are also associated with the location alias.

To enable RACF PassTickets you must define either a generic LU name for the data sharing group, an IPNAME for the data sharing group, or both. If all members of the data sharing group are to be configured to enable TCP/IP communications only, then all members of the data sharing group must share the same IPNAME value. If you use RACF PassTickets, then all LOCATION aliases must eventually refer to a single common IPNAMES row whose LINKNAME value must match the remote group's generic LU name or IPNAME value. The IPADDR value in the IPNAMES row must eventually refer to the group distributing DVIPA of the data sharing group.

If some of the members of the data sharing group are to be configured to enable TCP/IP communications only and the others are to enable both, then the IPNAME value of the TCP/IP only members must match the generic LU name of the members that enable both TCP/IP and SNA/APPC communications. The generic LU name or IPNAME value is used as the RACF application name when validating a RACF PassTicket at any member of the data sharing group.

Related tasks

Enabling Db2 to send RACF PassTickets (Managing Security)

Related reference

DSNJU003 (change log inventory) (Db2 Utilities)

Member-specific location aliases

Member-specific location aliases represent one, a subset, or all members of a data sharing group.

Location aliases are useful in several ways.

- Member-specific access requires that you identify to the server (remote data sharing group) each location alias that is defined by a requester. Doing so enables a server to recognize itself as the intended recipient of connection requests that specify location aliases instead of the server's location name.
- After adding a new member to a data sharing group, you can create an alias for the member's old (subsystem) Db2 location name that maps to its new (group) Db2 location name. Doing so enables applications that are coded to connect to the member's old Db2 location name to continue to work.
- Location aliases enable you to define subsets of data sharing group members. Subsetting gives you the ability to limit the members to which DRDA requesters can connect.

Clients can connect to a location alias using any port that Db2 is listening on. Therefore, location aliases can be defined without an *alias-port* or *alias-secport* value. Clients that use a location alias without an *alias-port* or *alias-secport* value receive information only about the member that processes the

connection request in the list of servers. To distribute work requests across the group or a subset of a group, clients must be configured to use the Db2 group location name or location alias name that is defined for a subset of members.

Dynamic location aliases

You can use the MODIFY DDF command with the ALIAS option to define and manage as many as 40 location aliases dynamically. You can start, stop, cancel, change, and delete dynamic location aliases without stopping either DDF or Db2. You can use the DISPLAY DDF command to find information about existing location aliases.

Static location aliases

You can use DSNJU003 (change log inventory) utility to define and modify as many as 8 static location aliases per member. Changes to these aliases require you to stop both DDF and Db2. The MODIFY DDF command cannot be used to manage this type of alias. Any alias of this type that has the same name as an alias that was previously defined by the MODIFY DDF command is not used. In that case, the dynamically defined alias takes precedence, and is used instead.

Related concepts

Update the BSDS with the DSNJU003 utility
DSNJU003 can be used to update the information stored in the bootstrap data sets of members of data sharing groups.

Related tasks

"Defining dynamic location aliases" on page 42
You can define dynamic location aliases that enable you to manage subsets of members in a data sharing group, without stopping and restarting DDF or Db2.

"Managing dynamic location aliases" on page 43
You can stop, cancel, modify, and start dynamic location aliases without stopping and restarting DDF or Db2.

Related reference

-DISPLAY LOCATION (Db2) (Db2 Commands)
-MODIFY DDF (Db2) (Db2 Commands)

Defining dynamic location aliases

You can define dynamic location aliases that enable you to manage subsets of members in a data sharing group, without stopping and restarting DDF or Db2.

Before you begin

Before you can define dynamic location aliases, Db2 must be started.

About this task

You can define as many as 40 dynamic location aliases. These aliases cannot be defined or managed by the DSNJU003 utility, and the DSNJU004 utility does not print any information about these aliases.

When a static and dynamic location alias are defined with the same name, Db2 uses only the dynamic location alias. The statically defined alias with the same name is ignored.

Procedure

To define dynamic location aliases:

1. Issue the MODIFY DDF command and specify the ALIAS and ADD options.
   For example, you might issue the following command, where *alias-name* is the name of the alias:

```
-MODIFY DDF ALIAS(alias-name) ADD
```

The location alias is defined as a stopped location alias.

2. Issue the MODIFY DDF command again to configure the alias. You can specify only one additional option each time that you issue the MODIFY DDF command with the alias option.
   For example, to configure the ALIAS1 location alias with a port of 9000 and an IPv4 address of 1.1.1.1, you would issue the following sequence of commands:
   a. `-MODIFY DDF ALIAS(ALIAS1) PORT(9000)`
   b. `-MODIFY DDF ALIAS(ALIAS1) IPV4(1.1.1.1)`
3. Issue the MODIFY DDF command and specify the ALIAS and START options.
   For example, you might issue the following command, where *alias-name* is the name of the alias:

   ```
   -MODIFY DDF ALIAS(alias-name) START
   ```

   DDF accepts requests to the newly defined location alias whenever DDF is started.

What to do next

You can manage location aliases that are created by the MODIFY DDF command dynamically by issuing the MODIFY DDF command to stop or cancel the alias, modify its configuration, and restart it, all without stopping DDF or Db2.

Related concepts

Member-specific location aliases

Member-specific location aliases represent one, a subset, or all members of a data sharing group.

Related tasks

Managing dynamic location aliases

You can stop, cancel, modify, and start dynamic location aliases without stopping and restarting DDF or Db2.

Managing dynamic location aliases

You can stop, cancel, modify, and start dynamic location aliases without stopping and restarting DDF or Db2.

Before you begin

Before you can modify existing dynamic location aliases, Db2 must be started.

About this task

Unlike statically defined location aliases, you cannot define or modify dynamic aliases by using the DSNJU003 utility, and the DSNJU004 utility does not print any information for dynamic location aliases.

When a static and dynamic location aliases are defined with the same name, Db2 accepts requests only to the dynamic location alias.

Procedure

To modify an existing dynamic location alias:

1. Issue the MODIFY DDF command to stop or cancel the alias.

Option	Description
To stop the alias	Issue the following command: `-MODIFY DDF ALIAS(alias-name) STOP` Db2 stops accepting new connection requests to the specified alias and existing database access threads continue to process connections to the specified alias. Inactive connections to the alias are automatically closed.

Option	Description
To cancel the alias	Issue the following command: `-MODIFY DDF ALIAS(alias-name) CANCEL` Db2 stops accepting new connection requests to the specified alias and existing database access threads that process connections to the specified alias are terminated. Inactive connections to the alias are automatically closed.

Stopped and canceled dynamic location aliases do not start automatically when you start DDF. In a data sharing group, Db2 deregisters the alias with WLM, which means that Db2 is no longer included in sysplex workload balancing information, related to the alias, that is returned to remote client systems. Those steps are not needed in non-data sharing environments, because Db2 never registers the alias with WLM.

2. Issue the MODIFY DDF command one or more times to modify the configuration of the alias. You can specify only one additional option with the MODIFY DDF command when you specify the ALIAS option. Therefore, you might need to issue the MODIFY DDF command more than once to make multiple changes to the configuration of the alias.
3. Issue the MODIFY DDF command to start the alias.
   For example, you might issue the following command, where *alias-name* is the name of the alias:

```
-MODIFY DDF ALIAS(alias-name) START
```

Example

For example, assume that a location alias ALIAS1 was defined by the MODIFY DDF command, and has the following configuration:

- A port: 9000
- An IPv4 address: 1.1.1.1

However, you want to modify the location alias to use following configuration:

- The default port
- No IPv4 address
- An IPv6 address: 3::3

You might issue the following series of commands:

1. `-MODIFY DDF ALIAS(ALIAS1) CANCEL`

   Db2 stops accepting new connection requests to the specified alias and existing database access threads that process connections to the specified alias are cancelled. You can now modify the configuration of the alias.
2. `-MODIFY DDF ALIAS(ALIAS1) NPORT`

   The port value is removed from the alias. The group SQL port is used instead.
3. `-MODIFY DDF ALIAS(ALIAS1) NIPV4`

   The IPv4 address is removed from the alias. The member-specific IP address is returned in the server list instead.
4. `-MODIFY DDF ALIAS(ALIAS1) IPV6(3::3)`

   The IPv6 address is added to the alias.
5. `-MODIFY DDF ALIAS(ALIAS1) START`

   DDF starts accepting new connections to the alias using its new configuration.

Related concepts

Member-specific location aliases

Member-specific location aliases represent one, a subset, or all members of a data sharing group.

Related tasks

Defining dynamic location aliases

You can define dynamic location aliases that enable you to manage subsets of members in a data sharing group, without stopping and restarting DDF or Db2.

Single-member access

Single-member access using TCP/IP is the same method that is used to access Db2 in non-data sharing environments.

With single-member access, a requester uses the real or virtual IP address in its TCP/IP configuration to connect to a specific member of the group. (Alternatively, a requester's TCP/IP configuration can contain a domain name, which resolves to an IP address.) The requester sends all connection and query requests to this same member.

Db2 Connect:

Db2 Connect Server is a connectivity server that concentrates and manages connections from multiple desktop clients and web applications to Db2 database servers. Configure Db2 Connect to use single-member access by disabling its Sysplex or concentrator support. When you disable Sysplex or concentrator support, no failover capability will be provided and if the single member fails, the application will not have any access. If you do not disable Sysplex support,Db2 Connect uses DRDA workload balancing by default to allocate requests among the members of the group.

Recommendation: Do not use single-member access for the following reasons:

- It is dependent on the specified member being operational.
- It provides no workload balancing or failover capability across the members of a data sharing group

Related tasks

Connecting systems with TCP/IP (Db2 Installation and Migration)

Related reference

Db2 Connect overview

Setting up Db2 for z/OS as a requester

Much of the processing in a distributed database environment requires the exchange of messages with other locations in the network.

About this task

When Db2 for z/OS acts as a requester, it can connect applications that run on the z/OS system to remote database servers, including Db2 for z/OS, Db2 for Linux®, UNIX, and Windows, DB2 for i, and DB2 Server for VSE & VM.

In its role as a requester, Db2 for z/OS accepts Db2 location names and translates them into TCP/IP addresses. It uses the communications database (CDB) to register location names and their corresponding network parameters. The data that is stored in the CDB enables Db2 to pass the required information when making distributed database requests over TCP/IP connections.

Procedure

To set up Db2 for z/OS as a requester:

1. Define the Db2 for z/OS requester to the local TCP/IP system.
2. Identify the remote data sharing groups to which applications can connect.

Related concepts

Db2 configuration with TCP/IP (Db2 Installation and Migration)

Remote data sharing group requirements

When an application requests data from a remote data sharing group, the Db2 for z/OS requester searches the CDB for information about the remote group. Db2 uses the CDB to store information about how to communicate with remote groups, and the DDF uses the CDB to map Db2 location names and location aliases to IP addresses.

Group access requires:

- A Db2 location name entry in the SYSIBM.LOCATIONS table
- An entry in SYSIBM.IPNAMES table with the group DVIPA and any outbound user authentication requirements

Member-routing access requires:

- Location alias entries in the SYSIBM.LOCATIONS table
- An entry in SYSIBM.IPNAMES table with the group DVIPA and any outbound user authentication requirements
- Optionally, location alias member entries in the SYSIBM.IPLIST table if you use subset location aliases

Single-member access requires:

- A location alias entry in the SYSIBM.LOCATIONS table.
- An entry in SYSIBM.IPNAMES table with the group DVIPA and any outbound user authentication requirements
- A location alias member entry in the SYSIBM.IPLIST table

SYSIBM.LOCATIONS

SYSIBM.LOCATIONS maps the Db2 location names in connection requests to the port numbers (or service names) of remote systems.

SYSIBM.LOCATIONS must contain at least one row for each remote group, depending on the access method that is used.

- For group access, the LOCATION column of the row contains the group's Db2 location name and PORT name. The LOCATION, PORT and the corresponding IPNAMES row identifies the members used to access Db2.
- For member-routing access, the LOCATION column of each row contains the group PORT number and location alias that identifies one, several, or all members of the group. The LOCATION, PORT and corresponding IPNAMES row controls which member is used to access Db2.
- For single-member access, the LOCATION column of the row contains the PORT number and location alias that identifies one member of the group. The LOCATION, PORT, and corresponding IPNAMES row identifies controls which member is used to access Db2.

Tip: You can specify a case-sensitive service name, instead of a port number, in the PORT column of the SYSIBM.LOCATIONS table.

Related reference

LOCATIONS catalog table (Db2 SQL)

SYSIBM.IPNAMES

SYSIBM.IPNAMES maps Db2 location names to the IP addresses or domain names of remote systems.

It also maps Db2 location names to the network security information required by the remote system. SYSIBM.IPNAMES must contain at least one row for each remote group, depending on the access method that is used.

- For group access, the IPADDR column of the row contains the group's domain name.

- For member-specific access, the LINKNAME column of each row contains a link name, and the IPADDR column of each row is blank.
- For single-member access, the LINKNAME column of the row contains a link name, and the IPADDR column of the row is blank.

SYSIBM.IPLIST

SYSIBM.IPLIST supports member-specific access to a remote data sharing group by enabling you to associate location aliases with one or more members of the group.

SYSIBM.IPLIST must contain a row for every member that is associated with a location alias. For members that are associated with multiple location aliases, insert multiple rows.

- For member-specific access, the LINKNAME column of each row contains a location alias and the IPADDR column of that row contains the DVIPA or domain name of a member that is associated with that location alias.
- For single-member access, the LINKNAME column of the row contains a location alias and the IPADDR column of that row contains the DVIPA or domain name of the member that is associated with that location alias.

Recommendation: If you are trying to limit the members of a data sharing group that can be accessed from this Db2 for z/OS requester, then the requester should be setup to access a data sharing group subset setup at the server (data sharing group). If the requester is going to access a group's subset, then the SYSIBM.IPLIST table should not be used.

Example: Assume that a remote data sharing group has six members. With member-specific access, you might have two location aliases, one of which is associated with three of the members and the other of which is associated with two of the members. In this example, you insert five rows into the SYSIBM.IPLIST table, three rows for the members that are associated with the first location alias and two rows for the members that are associated with the second location alias.

How Db2 sends requests

When sending a request, Db2 uses the value in the LINKNAME column of the SYSIBM.LOCATIONS table to determine which network protocol to use.

- If Db2 finds the same value in the LINKNAME column of the SYSIBM.IPNAMES table, it uses TCP/IP.
- If Db2 finds the same value in the LUNAME column of the SYSIBM.LUNAMES table, it uses SNA.
- If Db2 finds the same value in both SYSIBM.IPNAMES and SYSIBM.LUNAMES, it uses TCP/IP.

Updates to the communications database for TCP/IP connections

You can update the CDB while DDF is active.

Changes to the SYSIBM.LOCATIONS, SYSIBM.IPNAMES, SYSIBM.IPLIST tables take effect in the following manner for TCP/IP connections:

- Updates take effect when DDF opens a new connection to the remote group.
- Updates do not affect communication already in progress; existing communication continues as if the updates had not occurred.

Configuring data sharing groups as TCP/IP servers

Configuring a Db2 for z/OS data sharing group as a TCP/IP server includes several steps such as specifying port numbers and configuring member-specific access.

Procedure

To configure a data sharing group as a TCP/IP server:

1. Specify the DRDA port number on which all members listen for incoming SQL requests.
2. Specify a unique resynchronization port number for each member.
3. Designate subsets of members if you want to limit the members to which DRDA requesters can connect.

4. Specify a generic LU name and IPNAME value for the data sharing group if you use RACF PassTickets.

Specifying the DRDA port number
Db2 requires that all members of a data sharing group use the same, well-known port number to receive incoming SQL requests.

446 is the recommended DRDA port number. DRDA port numbers are stored in the bootstrap data sets (BSDSs) of members.

Use installation panel DSNTIP5 or the DSNJU003 (change log inventory) utility to specify the DRDA port number (PORT) for each member. Specify the same DRDA port number for all members of the same data sharing group.

The network administrator must also register the DRDA port number with TCP/IP on each member's z/OS system.

Related reference
DSNJU003 (change log inventory) (Db2 Utilities)

Reserving the DRDA port
If a member of a data sharing group is restarted on another LPAR, the TCP/IP on that system must be configured to allow the member to use the DRDA port.

To ensure this, reserve the DRDA port on each system for the Db2 DDF address space by assigning the port to every member that could conceivably start on that system. By explicitly assigning this port, you prevent other programs from using the DRDA port number.

On each system, replicate the TCP/IP PORT configuration profile statement shown here:

```
PORT
⋮
  446 TCP DB1ADIST SHAREPORT
  446 TCP DB2ADIST SHAREPORT
  446 TCP DB3ADIST SHAREPORT
  446 TCP DB4ADIST SHAREPORT
```

Recommendation: Specify the SHAREPORT option, as shown in the TCP/IP PORT configuration profile statement. Specifying this option configures TCP/IP to allow multiple listeners on the DRDA port (port 446). As client connection requests arrive for this port, TCP/IP distributes them across the members. TCP/IP selects the member with the fewest number of connections (both active and queued) at the time that the request is received.

The member chosen by TCP/IP receives all of the DRDA server's workload for that TCP/IP instance, leaving the other members with no TCP/IP server threads for DRDA. This is transparent to the DRDA clients if the member that is processing the TCP/IP requests does not reach the MAX REMOTE CONNECTED thread limit. If this limit is reached, the client's connection request is rejected.

Tip: After you resolve a failure situation, move the member back to its original CPC.

Specifying the resynchronization port numbers
Db2 requires that each member of a data sharing group have a resynchronization port number that is unique within the Parallel Sysplex.

In the event of a failure, this unique port number allows a requester to reconnect to the correct member so that units of work that require two-phase commit can be resolved.

Use the DSNJU003 (change log inventory) utility to specify a unique resynchronization port number (RESYNC) for each member of the group.

The network administrator must also register the resynchronization port number with TCP/IP on each member's z/OS system.

Related reference
DSNJU003 (change log inventory) (Db2 Utilities)

Configuring subsets for member-specific access

Db2 for z/OS allows you to define subsets of data sharing group members in TCP/IP networks for member-specific access.

About this task

By designating subsets of members, you can:

- Limit the members to which DRDA requesters can connect.

  System and database administrators might find this useful for any number of purposes.
- Ensure that initial connections are established only with members that belong to the specified subset.

  Without subsets, requesters can make initial and subsequent connections to any member of the data sharing group.
- Provide requesters with information about only those members in the subset.

  With subsets, a member that receives an initial connection request can return to the requester a list of members that are currently active, able to perform work, and represented by the location alias.

Procedure

To configure a subset for member-specific access:

- Complete one of the following actions:
  - Use the MODIFY DDF command to modify member specific location aliases without stopping Db2. For example, to modify the port number used by a member for an alias ALIAS1, you would issue the following sequence of commands.
    1. Issue a MODIFY DDF command to stop the alias:

       ```
       MODIFY DDF ALIAS (ALIAS1) STOP
       ```
    2. Issue a MODIFY DDF command to modify the member-specific alias port:

       ```
       MODIFY DDF ALIAS (ALIAS1) PORT (447)
       ```

       You can also add new aliases and define, modify, and remove SECPORT, IPV4 and IPV6 values for an existing alias.
    3. Issue a MODIFY DDF command to start the alias:

       ```
       MODIFY DDF ALIAS (ALIAS1) START
       ```

       - If any ports are specified for the alias, DDF registers any subset location aliases with z/OS Workload Manager (WLM). The list of members in the subset is managed automatically by z/OS.
       - DDF adds the TCP/IP port numbers of the subset location aliases to a TCP/IP SELECT socket call for the SQL request listener. Doing so enables the Sysplex Distributor to send requests that are intended for subset members to only those members that belong to the subset. It also enables members of the subset to respond to those requests.
  - Stop Db2 and invoke the DSNJU003 utility and specify the *alias-port* parameter of the ALIAS option of the DSNJU003 (change log inventory) utility. When you start DDF, it performs the following tasks:
    - If any ports are specified for an alias, DDF registers any subset location aliases with z/OS Workload Manager (WLM). The list of members in the subset is managed automatically by z/OS.
    - DDF listens for SQL requests on the alias ports specified for each subset that it is part of. This enables members of a subset identified by an alias name to respond to SQL requests to that particular alias, either coming in directly from a remote client or through the Sysplex Distributor.

Specifying a generic LU name and IPNAME value for the data sharing group

If you use RACF PassTickets for security, you must define either a generic LU name for the data sharing group, an IPNAME for the data sharing group, or both.

If all members of the data sharing group are to be configured to enable TCP/IP communications only, then all members of the data sharing group must share the same IPNAME value. Use the DSNJU003 (change log inventory) utility to define the IPNAME value. If all of the members of the data sharing group are to be configured to enable both SNA/APPC and TCP/IP communications, then all members of the data sharing group must share the same generic LU name. Use the DB2 GENERIC LUNAME parameter on installation panel DSNTIPR or use the DSNJU003 (change log inventory) utility to define the generic LU name.

If some of the members of the data sharing group are to be configured to enable TCP/IP communications only and the others are to enable both, then the IPNAME value of the TCP/IP only members must match the generic LU name of the members that enable both TCP/IP and SNA/APPC communications. The generic LU name or IPNAME value is used as the RACF application name when validating a RACF PassTicket at any member of the data sharing group.

Related tasks

Enabling Db2 to send RACF PassTickets (Managing Security)

Related reference

DSNJU003 (change log inventory) (Db2 Utilities)

Example of configuring Db2 Connect to access a subset of a data sharing group

When you need to configure Db2 Connect to access a subset of a data sharing group, looking at an example configuration can be helpful.

For this example, three aliases are defined, each of which specifies a subset of the data sharing group.

The aliases are associated with ports 5031, 5032, and 5033.

- Group Location: DB2GROUP1, port: 446
- Location ALIAS 1: ALIAS1, port 5031
- Location ALIAS 2: ALIAS2, port 5032
- Location ALIAS 3: ALIAS3, port 5033

One aspect of how the data sharing group is configured is whether dynamic virtual IP addressing (DVIPA) (with or without Sysplex Distributor) is being used. The only way to ensure that an initial connection attempt is made against an available member of the data sharing group that participates in a particular location alias of the group, is to configure:

- Each member of the data sharing group with its own member-specific DVIPA
- The group itself with a distributing DVIPA

The Sysplex Distributor capability of the z/OS Communications Server in a Sysplex environment can instantiate a group distributing DVIPA. Also, the group-distributing DVIPA is configured within the Sysplex to distribute requests to all servers that are listening on ports 446, 5031, 5032, and 5033. For this example, assume that the group distributing DVIPA has a value of Vgrp, which is a character string that represent an IPv4 or IPv6 address. When using the group distributing DVIPA, you do not need to know the member-specific DVIPAs.

The following configuration statements configure Db2 Connect to connect to any member of the group:

```
db2 catalog tcpip node grp1node remote Vgrp server 446
db2 catalog dcs db db2grp1 as db2group1 parms ',,,,,sysplex'
db2 catalog db db2grp1 as db2grp1 at node grp1node authentication server
```

In the preceding configuration statements, a node profile is defined with the group distributing DVIPA, and that node profile is also defined with the group TCP/IP port. You should create additional Db2 Connect node profiles that are configured with the group distributing DVIPA and with each of the locations aliases port values.

The following Db2 Connect configuration statements are required to connect to members of the location aliases:

```
db2 catalog tcpip node alias1 remote Vgrp server 5031
db2 catalog dcs db alias1 as alias1 parms ',,,,,sysplex'
db2 catalog db alias1 as alias1 at node alias1 authentication server

db2 catalog tcpip node alias2 remote Vgrp server 5032
db2 catalog dcs db alias2 as alias2 parms ',,,,,sysplex'
db2 catalog db alias2 as alias2 at node alias2 authentication server

db2 catalog tcpip node alias3 remote Vgrp server 5033
db2 catalog dcs db alias3 as alias3 parms ',,,,,sysplex'
db2 catalog db alias3 as alias3 at node alias3 authentication server
```

When the initial connection is successful to a member of the location or the location alias, a list of available member-specific IP addresses is returned to the Db2 Connect server. When the initial connection is made to a location alias, the IP addresses returned are the member-specific alias IP addresses, if they exist. Because the Sysplex parameter is specified on each of the dcs db profiles, subsequent connection attempts are made to the servers by using the member-specific DVIPAs that are returned.

If you choose not to use the Sysplex Distributor but have configured each member of the data sharing group with a member-specific DVIPA, a group-distributing DVIPA is not available to connect to the various members of the data sharing group. If this is how the group was originally configured, you need to create separate domain names in a DNS that represent the entire group (meaning that DVIPAs of the members must be defined in this domain name). Also, each of the aliases that specifies a subset of data sharing member (DVIPAs for only the members that are defined to part of a subsetting location alias can be defined in a particular domain name). Using this configuration, you can substitute Vgrp in the tcpip node profile named grp1node with the domain name that represents the entire group. For each tcpip node profile that is named ALIAS1, ALIAS2, and ALIAS3, you can substitute Vgrp with the domain name that represents each location alias respectively. If you do not use a group distributing DVIPA, Db2 Connect might attempt to connect to an unavailable member of the group or location alias, which will result in a TCP/IP connection failure. A connection failure is not returned to the requesting client until connection attempts to each IP address that is defined in the specified domain name have failed.

You might also choose not to use DVIPAs. This choice causes the domain name entries in the DNS to be populated with an IP address of each system that has a member running on it. If a member moves to another system, you need to manually update the DNS entries.

Related concepts

Java client direct connect support for high availability for connections to Db2 for z/OS servers (Db2 Application Programming for Java)

Related reference

Non-Java client support for high availability for connections to Db2 for z/OS servers

Db2 Connect overview

Connecting distributed partners in a TCP/IP network

Db2 Connect requesters can be configured to use group access to access remote data sharing groups in TCP/IP networks. Db2 for z/OS requesters can be configured to use group access or member-specific access to access remote data sharing groups in TCP/IP networks.

Configuring a Db2 Connect requester to use group access

You can configure a Db2 Connect requester to use group access to access remote data sharing groups in TCP/IP networks.

About this task

You should use the Sysplex and connection concentrator functions when configuring a Db2 Connect requester. The connection concentrator reduces the resources required on Db2 for z/OS database servers to support large numbers of workstation and web users. The concentrator also allows applications to stay

connected without any resources being consumed on the Db2 host server. The database directory should use the group IP address.

The IBM Data Server Driver for JDBC and SQLJ is an architecture-neutral JDBC driver for distributed and local Db2 access. The IBM Data Server Driver for JDBC and SQLJ architecture is independent of any particular JDBC driver-type connectivity or target platform, which allows for both Type 4 and Type 2 connectivity in a single driver instance to Db2 platforms.

The z/OS Sysplex distributor can be used to provide an automated IP layer of connectivity within the Sysplex, but should not be viewed as a replacement of the Connect Server SYSPLEX and concentrator functions. It is a combination of the high availability features of DVIPA, Sysplex distributor, and Connect Server SYSPLEX and concentrator functions that provides the highest availability, best quality of service, and the workload optimization capabilities of WLM. The WLM Routing services and the Db2 weights provided to Connect Server, which are used as routing recommendations, are specific to the Db2 servers. They take several indicators into account, like system load of the member that the Db2 member is running on, the Performance Index (PI) of the given member or the enclave queue-time waiting for a free database access thread related to the member and others.

In addition, Db2 for z/OS can have health conditions, which are conditions that reduce the ability of the server to handle requests. Usually these conditions cannot be directly realized or measured by WLM, but are only known to the server itself or the application that owns or monitors that server. The health indicator enhancement gives the application the opportunity to inform WLM about health conditions, so that the routing service is able to take them into account when evaluating routing recommendations. Db2 for z/OS storage utilization is used as the indicator of health of the member.

This topic outlines the most important steps for configuring a Db2 Connect requester to use group access.

Procedure

To configure a Db2 Connect requester to use group access:

1. Use the Configuration Assistant to update the database directories that Db2 Connect uses to manage database connection information.

   Important: You must enable Sysplex support for the data sharing group by specifying the SYSPLEX parameter in the parameter string of the target database name in the DCS directory.
2. If you want applications to be able to update data in multiple remote database servers with guaranteed integrity, use the Db2 Control Center to enable and test multisite updates.
3. Bind the Db2 Connect utilities, and any applications that are developed using embedded SQL, to the databases with which they operate.

Related concepts

Java client direct connect support for high availability for connections to Db2 for z/OS servers (Db2 Application Programming for Java)

Related reference

Db2 Connect overview

Configuring a Db2 requester to use group access

You can update the CDB of a Db2 for z/OS requester to use group access.

About this task

Recall that DVIPA network addressing uses dynamic virtual IP addresses (DVIPAs).

Procedure

To enable group access to a remote data sharing group:

1. Identify the Db2 location name of the remote group. Insert the group's Db2 location name into the LOCATION column of the requester's SYSIBM.LOCATIONS table.

2. Identify the DVIPA or domain name of the remote group, and specify the security definitions for conversations with group members. Insert the group's DVIPA or domain name into the IPADDR column of the requester's SYSIBM.IPNAMES table.
3. Map the Db2 location name of the remote group to its DVIPA or domain name.

   If you use RACF PassTickets, specify the group's generic LU name or IPNAME value as the value of the LINKNAME columns.

   Insert the same link name into the LINKNAME columns of the requester's SYSIBM.LOCATIONS and SYSIBM.IPNAMES tables. Be sure to update only those rows that are associated with the remote group.

Example

The following example provides sample SQL statements for enabling group access that uses DVIPA network addressing. It also shows the results of those statements in the form of table excerpts. This example assumes that a remote data sharing group exists with a group location name of DB2A and a group DVIPA of Vx.

The following SQL statements update the CDB of a Db2 for z/OS requester to use group access (DVIPA network addressing):

1. This statement identifies the Db2 location name of the remote group:

   GUPI

   ```
   INSERT INTO SYSIBM.LOCATIONS (LOCATION, PORT)
     VALUES ('DB2A', '446');
   ```

   GUPI

2. This statement identifies the DVIPA of the remote group and specifies security definitions for conversations with the group's members:

   GUPI

   ```
   INSERT INTO SYSIBM.IPNAMES (SECURITY_OUT, USERNAMES, IPADDR)
     VALUES ('A', ' ', 'Vx');
   ```

   GUPI

3. These statements map the Db2 location name of the remote group to the DVIPA of the remote group:

   GUPI

   ```
   UPDATE SYSIBM.LOCATIONS
              SET LINKNAME='DB2ALINK'
              WHERE LOCATION='DB2A';
   ```

   ```
   UPDATE SYSIBM.IPNAMES
              SET LINKNAME='DB2ALINK'
              WHERE IPADDR='Vx';
   ```

   GUPI

The following table shows an example excerpt of the SYSIBM.LOCATIONS table after the update.

Table 2. Db2 location name of a remote data sharing group in a Db2 requester's SYSIBM.LOCATIONS table. Not all columns are shown.

LOCATION	LINKNAME	PORT
DB2A	DB2ALINK	446

The following table shows an example excerpt of the SYSIBM.IPNAMES table after the update.

Table 3. DVIPA of a remote data sharing group in a Db2 requester's SYSIBM.IPNAMES table. Not all columns are shown.

LINKNAME	SECURITY_OUT	USERNAMES	IPADDR
DB2ALINK	A		Vx

The following SQL statement connects to the remote group:

```
CONNECT TO DB2A;
```

Configuring a Db2 requester to use member-specific access

You can update the CDB of a Db2 for z/OS requester to use member-specific access.

Before you begin

You should disable Sysplex support when configuring a Db2 requester for member-specific access. If you do not disable Sysplex support, Db2 Connect uses DRDA workload balancing by default to allocate requests among the members of the group. The database directory should use the member IP address.

Procedure

To enable member-specific access to a remote data sharing group:

1. Define one or more location aliases that identify different sets of members. Insert each location alias into the LOCATION column of the requester's SYSIBM.LOCATIONS table.
2. Map each location alias to the security definitions for conversations with each set of members.

   If you use RACF PassTickets, then all LOCATION aliases must eventually refer to a single common IPNAMES row whose LINKNAME value must match the remote group's generic LU name or IPNAME value. The IPADDR value in the IPNAMES row must eventually refer to the group distributing DVIPA of the data sharing group.

   Insert the same link names into the LINKNAME columns of the requester's SYSIBM.LOCATIONS and SYSIBM.IPNAMES tables. In the SYSIBM.IPNAMES table, insert the appropriate security definitions for your site, and leave the IPADDR column blank.
3. Map each location alias to the domain names or DVIPAs of the data sharing members in that set.

   For each member of a set, insert the domain name or DVIPA of the member into the IPADDR column of the requester's SYSIBM.IPLIST table.
4. Define location aliases on the server (remote data sharing group).

   For each location alias on the requester's side, you must define a location alias at the server by using the DSNJU003 (change log inventory) utility.

Example

The following example provides sample SQL statements for enabling member-specific access that uses DVIPA network addressing. It also shows the results of those statements in the form of table excerpts. This example assumes that a remote data sharing group exists with a Db2 location name of DB2A and three members with DVIPAs V1, V2, and V3. Location aliases that are to be defined are DB2B (with members V1 and V2), DB2C (with members V2 and V3), and DB2D (with member V1).

The following SQL statements update the CDB of a Db2 for z/OS requester to use member-specific access (DVIPA network addressing):

1. These statements define location aliases that identify different sets of members:

   GUPI

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, PORT)
  VALUES ('DB2B', '446');
```

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, PORT)
  VALUES ('DB2C', '446');
```

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, PORT)
  VALUES ('DB2D', '446');
```

GUPI

2. These statements map each location alias to the security definitions for conversations with each set of members:

GUPI

```
UPDATE SYSIBM.LOCATIONS
          SET LINKNAME='ALIASB'
          WHERE LOCATION='DB2B';
```

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME)
          VALUES ('ALIASB')
```

```
UPDATE SYSIBM.LOCATIONS
          SET LINKNAME='ALIASC'
          WHERE LOCATION='DB2C';
```

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME)
          VALUES ('ALIASC')
```

```
UPDATE SYSIBM.LOCATIONS
          SET LINKNAME='ALIASD'
          WHERE LOCATION='DB2D';
```

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME)
          VALUES ('ALIASD')
```

GUPI

3. These statements map each location alias to the DVIPAs of the data sharing members in that set:

GUPI

```
INSERT INTO SYSIBM.IPLIST (LINKNAME, IPADDR)
          VALUES ('ALIASB', 'V1')
```

```
INSERT INTO SYSIBM.IPLIST (LINKNAME, IPADDR)
          VALUES ('ALIASB', 'V2')
```

```
INSERT INTO SYSIBM.IPLIST (LINKNAME, IPADDR)
          VALUES ('ALIASC', 'V2')
```

```
INSERT INTO SYSIBM.IPLIST (LINKNAME, IPADDR)
          VALUES ('ALIASC', 'V3')
```

```
INSERT INTO SYSIBM.IPLIST (LINKNAME, IPADDR)
          VALUES ('ALIASD', 'V1')
```

GUPI

The following table shows an example excerpt of the SYSIBM.LOCATIONS table after the update.

Table 4. Location aliases for a remote data sharing group in a Db2 requester's SYSIBM.LOCATIONS table. Not all columns are shown.

LOCATION	LINKNAME	PORT
DB2B	ALIASB	446
DB2C	ALIASC	446
DB2D	ALIASD	446

Location aliases are not associated with the DVIPA of a remote data sharing group in a Db2 requester's SYSIBM.IPNAMES table. The following table shows an example excerpt of the SYSIBM.IPNAMES table after the update.

Table 5. Excerpt of the SYSIBM.IPNAMES table. Not all columns are shown.

LINKNAME	IPADDR
ALIASB	
ALIASC	
ALIASD	

The following table shows an example excerpt of the SYSIBM.IPLIST table after the update.

Table 6. DVIPAs of the members that are associated with each location alias in a Db2 requester's SYSIBM.IPLIST table. Not all columns are shown.

LINKNAME	IPADDR
ALIASB	V1
ALIASB	V2
ALIASC	V2
ALIASC	V3
ALIASD	V1

The following SQL statement connects to the remote group using a location alias:

```
CONNECT TO DB2C;
```

Related reference

Db2 Connect overview

SNA access methods

Systems Network Architecture (SNA) is a proprietary IBM architecture that describes the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

SNA contains several functional layers and includes the following components:

- An application programming interface (API) called Virtual Telecommunications Access Method (VTAM®)
- A communications protocol for the exchange of control information and data
- A data link layer called Synchronous Data Link Control (SDLC)

SNA also includes the concept of nodes, which can contain both physical units that provide certain setup functions, and logical units (LUs), each of which is associated with a particular network transaction.

A data sharing group can simultaneously serve requesters that use member-routing access, group-generic access, and single-member access. As a server, the group is flexible about the access methods that requesters use. A requester can use member-routing access in one session and use group-generic access in another session. However, as a requester, the group always uses the same access method when communicating with a particular server.

Recommendation: Configure a data sharing group to use member-routing access whenever possible, unless your requester does not support it. It provides better workload balancing and two-phase commit support than group-generic access.

In SNA networks, each member of a data sharing group has its own LU name (in addition to the net ID that the member obtains from VTAM when the DDF starts). You can also define a *generic LU name* that represents all the members in the group.

An SNA requester can use one of several access methods to connect to a data sharing group:

Member-routing access
A requester can define a server location name that is associated with many LU names. Unlike with the generic LU name, a requester can establish sessions with one or more subsystems in the group. For member routing, the requests are distributed to members of the data sharing group based on their capacity.

Group-generic access
A requester uses the group's generic LU name to make an initial connection to any member of the group. With this access method, VTAM chooses one of the group members and establishes a session with that member on behalf of the requester. Subsequent queries from the requester are also directed to that same member.

Single-member access
A requester uses a member's LU name to connect and send queries to a single member of the group. All connection and query requests from that requester are directed to the same member.

Member-routing access

To access any member of a remote data sharing group with member routing, the location name of the data sharing group is associated with a list of LUNAMEs of the group members.

A subset of the member LUNAMEs can comprise the list to limit which members of the group can be accessed by the requester.

Requesters can use an LULIST to bypass VTAM workload balancing by establishing sessions with one or more members. Workload is balanced among members at the requester's discretion.

With member-routing access, a requester uses an LULIST to make an initial connection to one of the members that is represented by the alias. The Db2 Sysplex transaction program (Db2 STP) works in conjunction with the z/OS Workload Manager (WLM) to return a list of members that are currently active and can perform work. It also returns a weight for each active member that indicates the member's current capacity. The requester uses this information to connect to the member or members with the most capacity. If the LULIST contains a subset of the members, only the members of that subset can be accessed.

Two-phase commit resynchronization

If two-phase commit resynchronization is necessary, due to previous communication or system failure, the resynchronization request is sent to the member that is involved in the failure. Db2 performs resynchronization asynchronously, so that requesters are not prevented from establishing new sessions with other members.

RACF PassTicket limitation

If you use RACF PassTickets, you can define only one location alias. The generic LU name is used to generate a valid PassTicket for the data sharing group. Generation of a valid PassTicket for individual members or sets of members in a group is not possible.

Related concepts

Two-phase commit process (Db2 Administration Guide)

Group-generic access

With group-generic access, a requester uses the generic LU name of the group to make an initial connection to any member of the group.

VTAM then chooses one of the group members and establishes a session with that member on behalf of the requester. The requester uses the LU name of the chosen member when it makes subsequent requests.

VTAM provides workload balancing at the session level by allocating each connection request to the member with the fewest number of sessions at the time of connection. VTAM determines which member has the most capacity based on either the number of each member's active DDF sessions or the result of a user-written VTAM or z/OS Workload Manager exit routine.

After a connection is established between a requester and a member, all requests from the same requester that are made during the session are directed to the same member. VTAM does not reassess a member's capacity at the time of subsequent requests. Only after all connections between the requester and member are closed, can VTAM choose a different member to process future requests from the requester.

Exception: If the connection between the requester and the member is enabled for two-phase commit processing, VTAM continues to direct all requests from this requester to the same member. The mapping between the requester LU and the member LU is preserved until the Db2 command RESET GENERICLU is issued.

If a requester is active at the time of a communication or system failure, VTAM handles the reconnection in one of two ways:

- If the pre-failure session was enabled for two-phase commit support, VTAM reconnects the requester to the same member.

  If that member is unavailable, a communication error is returned.
- If the pre-failure session was not enabled for two-phase commit support, VTAM can connect the requester to any member.

Because VTAM balances the workload based on the number of currently established sessions rather than on true capacity, you should carefully evaluate this access method before implementing it. If the number of sessions is an accurate reflection of your workload, this method can be a good choice. It has the added benefit of being relatively easy to set up.

Single-member access

Single-member access using SNA is the same method that is used to access Db2 in non-data sharing environments.

With single-member access, a requester specifies the LU name of the specific member of the group as the value of the LINKNAME column of both a SYSIBM.IPNAMES row and a SYSIBM.LOCATIONS row. The requester sends all connection and query requests to that member.

With single-member access, a requester uses the LU name in its TCP/IP configuration to connect to a specific member of the group. The requester sends all connection and query requests to this same member.

Db2 for z/OS: Configure Db2 for z/OS to use single-member access by creating and using a location alias that represents only one member of the data sharing group. Otherwise, Db2 for z/OS uses DRDA workload balancing by default to allocate requests among the members of the group.

Recommendation: Do not use single-member access for the following reasons:

- It is dependent on the specified member being operational.
- It provides no workload balancing for requesters that cannot perform DRDA workload balancing.

Restriction: Version 8 of Db2 Connect does not support single-member access.

Setting up Db2 for z/OS as a requester

Much of the processing in a distributed database environment requires the exchange of messages with other locations in the network.

About this task

When Db2 acts as a requester, it can connect applications that run on the system to remote database servers including Db2 for z/OS, DB2 for i, and DB2 for VSE & VM.

In its role as a requester, Db2 for z/OS accepts Db2 location names and translates them into SNA NETID.LUNAMEs. It uses the communications database (CDB) to register Db2 location names and their corresponding network parameters. The data that is stored in the CDB enables Db2 to pass the required information when making distributed database requests over SNA connections.

Procedure

To set up Db2 for z/OS as a requester:

1. Define the Db2 for z/OS requester to the local VTAM system.
2. Identify the remote data sharing groups to which applications can connect.

Related tasks

Connecting systems with VTAM (Db2 Installation and Migration)

Remote data sharing group requirements

This section describes the requirements of member-specific, group-generic, and single-member access.

When an application requests data from a remote subsystem, the Db2 for z/OS requester searches the CDB for information about the remote group. Recall that Db2 uses the CDB to store information about how to communicate with remote groups, and that the DDF uses the CDB to map Db2 location names to LU names.

Member-specific access requires:

- Location entries in the SYSIBM.LOCATIONS table
- Conversational security requirements entries in the SYSIBM.LUNAMES table
- LUNAMEs of the members to be accessed in the SYSIBM.LULIST table

Group-generic access requires:

- A Db2 location name entry in the SYSIBM.LOCATIONS table
- A conversational security requirements entry in the SYSIBM.LUNAMES table

Single-member access requires:

- A location entry in the SYSIBM.LOCATIONS table. If you plan to allow group access concurrently with single-member or member-specific access, then the requester should use a server-defined location alias name in the SYSIBM.LOCATIONS table.
- A conversational security requirements entry in the SYSIBM.LUNAMES table
- The LUNAME of a single member in the SYSIBM.LULIST table

SYSIBM.LOCATIONS

SYSIBM.LOCATIONS maps the Db2 location names in connection requests to the VTAM LU names of remote systems and, if necessary, transaction program names (TPNs).

SYSIBM.LOCATIONS must contain at least one row for each remote group, depending on the access method that is used.

- For member-routing access, the LOCATION column of the row contains the group's location name, and the LINKNAME column of the row should contain the group's generic LU name.

- For group-generic access, the LOCATION column of the row contains the group's Db2 location name and the LINKNAME column of the row contains that group's generic LU name.
- For single-member access, the LOCATION column of the row contains a location alias that identifies one member of the group.

SYSIBM.LUNAMES

SYSIBM.LUNAMES maps Db2 location names to the LU names of remote systems. It also maps Db2 location names to the security and mode requirements for conversations with those systems.

SYSIBM.LUNAMES must contain at least one row for each remote group, depending on the access method that is used.

- For member-routing access, the LINKNAME column of the row should contains the group's generic LU name.
- For group-generic access, the LUNAME column of the row contains the group's generic LU name.
- For single-member access, the LINKNAME column of the row contains a link name.

The GENERIC column of SYSIBM.LUNAMES is for the specific use of data sharing group members that act as requesters. Its value indicates whether a requester uses the group's generic LU name or its own real LU name when identifying itself to a remote subsystem. If you want a requester to use the group's generic LU name, specify a value of Y; if you want a requester to use its own LU name, specify a value of N.

Important: The value of the GENERIC column is ignored if the Db2 GENERIC LUNAME parameter on installation panel DSNTIPR is blank. In this case, Db2 requesters are identified to remote systems by their own LU names.

Be aware that only one member of a data sharing group can access a remote system by using the group's generic LU name. If one member of a group is already connected to a remote system and is using the group's generic LU name, subsequent connections to that remote system by other members of the same group use the requesters' own LU names.

SYSIBM.LULIST

SYSIBM.LULIST supports member-routing access to a remote data sharing group by enabling you to associate a list of LU names with one or more members of the group.

SYSIBM.LULIST must contain a row for every member that the requester is allowed to access.

- For member-routing access, the LUNAME column of each row contains a member's LU name.
- For single-member access, the LUNAME column of the row contains the member's LU name.

Example: Assume that a remote data sharing group has six members. You might want the requester to be able to access only three members of the group. In this example, you insert three rows into the SYSIBM.LULIST table for the members to be accessed.

How Db2 sends requests

When sending a request, Db2 uses the value in the LINKNAME column of the SYSIBM.LOCATIONS table to determine which network protocol to use.

- If Db2 finds that same value in the LINKNAME column of the SYSIBM.IPNAMES table, it uses TCP/IP.
- If Db2 find that same value in the LUNAME column of the SYSIBM.LUNAMES table, it uses SNA.
- If Db2 finds that same value in both SYSIBM.IPNAMES and SYSIBM.LUNAMES, it uses TCP/IP.

Updates to the communications database for VTAM (SNA) connections to a data sharing group

You can update the CDB while DDF is active.

Changes to the SYSIBM.LOCATIONS, SYSIBM.LUNAMES, and SYSIBM.LULIST table take effect in the following manner for SNA connections:

- If the DDF has not yet tried to communicate with a particular remote group, updates take effect when DDF attempts to communicate with that group.

- If the DDF has already attempted to communicate with a particular remote group, updates take effect take the next time DDF is started.
- Updates do not affect existing conversations; existing conversations continue to operate as if the updates had not occurred.

Configuring data sharing groups as servers

DRDA requesters can access any member of a data sharing group through SNA if the SYSIBM.LUNAMES table of the communications database contains a default row that enables access by any requester that is not defined by a row in the table.

Installation job DSNTIJSG normally creates a default row unless the installation process has been altered to remove that step. If the SYSIBM.LUNAMES table does not contain a default row, you must, for each DRDA SNA requester, insert a row that contains the LUNAME of that requester.

Related concepts

Configuring data sharing groups as servers

DRDA requesters can access any member of a data sharing group through SNA if the SYSIBM.LUNAMES table of the communications database contains a default row that enables access by any requester that is not defined by a row in the table.

Configuring data sharing groups for group-generic access

You can configure a Db2 server for group-generic access to make requests and handle requests.

Procedure

To enable group-generic access to a data sharing group:

1. Include information in the coupling facility for support of VTAM generic resources (the ISTGENERIC structure).
2. Define a generic LU name for the group.
3. Configure the members to use the group's generic LU name when identifying themselves to remote subsystems.
4. Identify the generic LU names of requesting data sharing groups.
5. Identify the LU names of individual requesters, such as Db2 Connect and members of requesting data sharing groups.

Related reference

Generic resources (SNA Network Implementation Guide)

zSeries Processor Resource/Systems Manager Planning Guide

Defining a generic LU name for the group

Define a generic LU name for the data sharing group by specifying the name for each member in the DB2 GENERIC LUNAME parameter on installation panel DSNTIPR.

A generic LU name can be one to eight characters in length, and it is stored in each member's bootstrap data set (BSDS).

Using the originating member's LU name

Using the LU name of the originating member of the group as the generic LU name, and changing the originating member's LU name can be useful. This is particularly true when the originating member is already acting as a server for applications that will migrate to the data sharing group.

By configuring the group to use the originating member's LU name, you avoid needing to make extensive changes to the CDBs of Db2 for z/OS requesters.

To change the LU name of the originating member of the group, you must change both the LU name value in the member's VTAM APPL statement and the LUNAME value in the member's bootstrap data set (BSDS).

For example, assume that a group is formed whose originating member is already defined to the requesters that are shown in the following figure. Until you define a generic LU name when you enable

group-generic access, requesters can access DB2A's data only through the originating member (LUDB2A). If the originating member is unable to handle requests, requesters are unable to access DB2A's data.

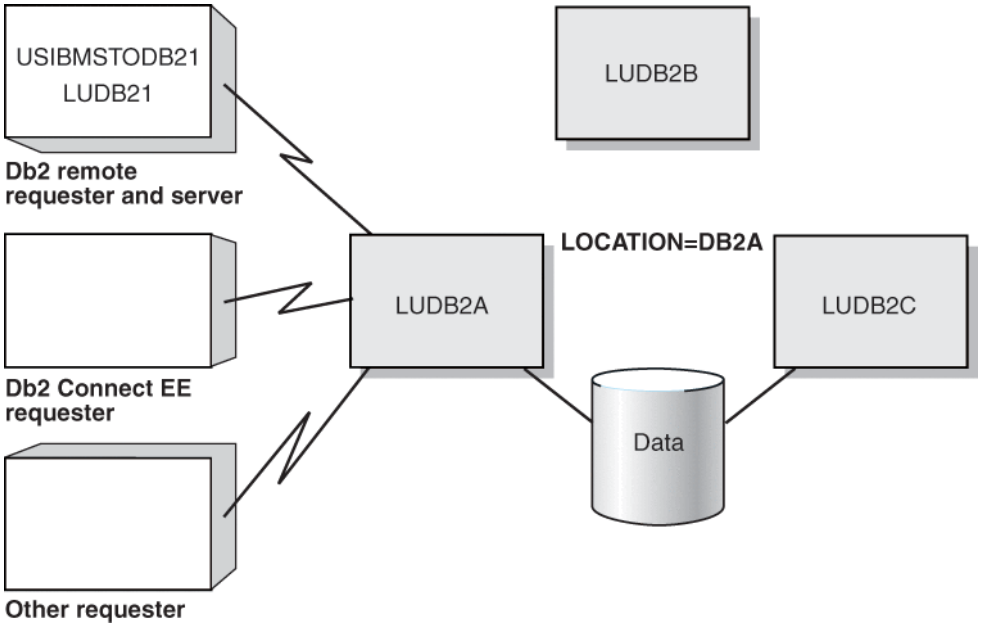


Figure 18. Example configuration before enabling group-generic access.

As shown in the following figure, by enabling group-generic access and by using the LU name of the originating member as the generic LU name, requesters can, with minimal change, access DB2A's data from any member of the group.

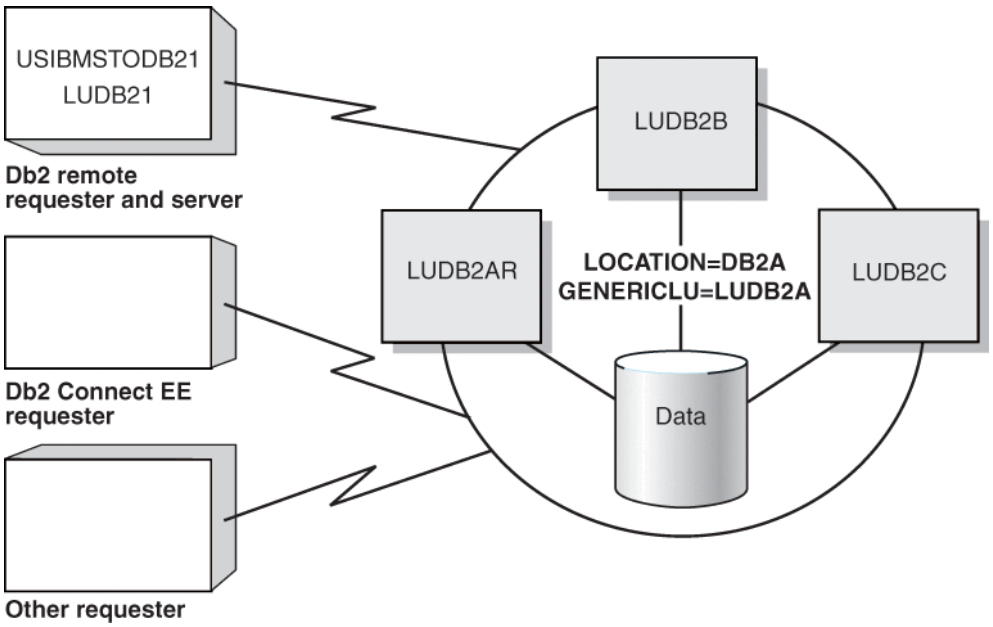


Figure 19. Example configuration after enabling group-generic access.

Configuring members to use the group's generic LU name

Specify a value of Y in the GENERIC column of each row of the SYSIBM.LUNAMES table that corresponds to a remote subsystem with which the data sharing group communicates.

The GENERIC column of the SYSIBM.LUNAMES table is for the specific use of group members that act as requesters. Its value indicates whether a member uses the group's generic LU name or its own real LU

name when identifying itself to a remote subsystem. The remote subsystem must be able to recognize the generic LU name.

Important: The value of the GENERIC column is ignored if the DB2 GENERIC LUNAME parameter on installation panel DSNTIPR is blank.

Be aware that only one member of a data sharing group can access a remote subsystem by using the group's generic LU name. If one member of a group is already connected to a remote subsystem and is using the group's generic LU name, subsequent connections to that remote subsystem by other members of the same group use the requesters' own LU names. If the remote subsystem only accepts generic LU names, all requests from the members of the requesting group must be routed through one member.

Clarification: The GENERIC column does **not** determine whether a member uses group-generic or member-specific access. The rows, or lack of rows, in SYSIBM.LULIST determine whether a member uses group-generic or member-specific access.

If the remote subsystem starts CNOS processing first, VTAM uses the name with which the remote subsystem connected, whether this is the real LU name or the generic LU name. Because this behavior is not always predictable, the subsystem that is handling requests from the data sharing group should be able to accept either the generic LU name or the real LU name when group-generic processing is used.

Example: Assume that the LU names of two group members are LUDB2NY and LUDB2LA. An excerpt of the SYSIBM.LUNAMES table would look similar to the following table.

Table 7. SYSIBM.LUNAMES table of a requesting data sharing group that uses group-generic access. Not all columns are shown.

LUNAME	GENERIC
LUDB2NY	Y
LUDB2LA	Y

Use the following SQL statements to update the SYSIBM.LUNAMES table to specify that members use their group's generic LU name when identifying themselves to remote data sharing groups:

GUPI

```
UPDATE SYSIBM.LUNAMES
   SET GENERIC='Y'
   WHERE LUNAME='LUDB2NY';
```

```
UPDATE SYSIBM.LUNAMES
   SET GENERIC='Y'
   WHERE LUNAME='LUDB2LA';
```

GUPI

Identifying the generic LU names of requesting data sharing groups

As part of configuring a data sharing group for group-generic access, you must identify the generic LU names of requesting data sharing groups.

Procedure

Use the INSERT statement to add the generic LU names of requesting data sharing groups to the LUNAME column of the server's SYSIBM.LUNAMES table.

Example

GUPI

Assume that the generic LU names of two requesting data sharing groups are LUDSG1 and LUDSG2. Use the following SQL statements to populate the SYSIBM.LUNAMES table with the generic LU names.

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUDSG1');
```

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUDSG2');
```

GUPI

An excerpt of the server's SYSIBM.LUNAMES table would look similar to the following table.

Table 8. Generic LU names of requesting data sharing groups in a Db2 server's SYSIBM.LUNAMES table. Not all columns are shown.

LUNAME	GENERIC
LUDSG1	
LUDSG2	

Specifying the LU names of requesters

As part of configuring a data sharing group for group-generic access, you must specify the LU names of requesters.

About this task

Installation job DSNTIJSG inserts a default row into the SYSIBM.LUNAMES table unless the installation process has been altered to remove this step. The default row enables access by any requester that is not defined by a row in the table. The following table shows the default row in the SYSIBM.LUNAMES table of a Db2 server.

Table 9. An example of the default row in the SYSIBM.LUNAMES table of a Db2 server. This example shows only those columns that are accessed when Db2 is a server.

LUNAME	SYSMODENAME	SECURITY_IN	GENERIC
(8 blanks)	'A'	'N'	

You can change the values in the default row before you start DDF.

Individual requesters include Db2 for z/OS and the members of requesting data sharing groups.

Procedure

To specify LU names of requesters:

1. Use the INSERT statement to add the LU names of requesters to the LUNAME column of the SYSIBM.LUNAMES table of the server.
2. To limit the number of conversations that a Db2 server can accept from a DRDA SNA LU partner, change or add the row for the partner LU in the SYSIBM.LUNAMES table to specify an SNA MODENAME. You must also add or update a row in the SYSIBM.LUMODES table that contains the SNA MODENAME.
3. To block inbound already-verified conversations from another partner LU name, change or add a row for the LU name to specify 'V' for SECURITY_IN.
4. To perform inbound userid translation from a partner LU name, update or add a row in the SYSIBM.LUNAMES table to specify 'I' or 'B' in the USERNAMES column. You must also update entries in the SYSIBM.USERNAMES table.

Example

GUPI

Assume that the LU names of the members of one requesting data sharing group are LUMEM1, LUMEM2, and LUMEM3, and that the LU names of the members of another requesting data sharing group are LUMEMA and LUMEMB. The following statements insert the LU names of requesters into the SYSIBM.LUNAMES table.

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUMEM1');
```

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUMEM2');
```

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUMEM3');
```

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUMEMA');
```

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUMEMB');
```

GUPI

An excerpt of the server's SYSIBM.LUNAMES table would look similar to the following table.

Table 10. LU names of requesters in a SYSIBM.LUNAMES table of a Db2 server. Not all columns are shown.

LUNAME	SYSMODENAME	SECURITY_IN	USERNAMES	GENERIC
LUMEM1				
LUMEM2				
LUMEM3				
LUMEMA				
LUMEMB				

Connecting distributed partners in an SNA network

You can configure Db2 requesters to use member-routing access, group-generic access, or member-specific access to access remote data sharing groups in SNA networks.

For information about how to configure other types of requesters, consult the product documentation for your specific requester.

Configuring a Db2 requester to use member-routing access

You can update the CDB of a Db2 requester to use member-routing access.

Procedure

To enable member-routing access to a remote data sharing group:

1. Insert the group's location name into the LOCATION column of the requester's SYSIBM.LOCATIONS table.

   If you use RACF PassTickets, define only one location row.
2. Map the location to the security and mode requirements for conversations with each set of members.

Insert the same link name into the LINKNAME column of the requester's SYSIBM.LOCATIONS and SYSIBM.LUNAMES tables. Insert the appropriate security and mode requirements for your site.

If you use RACF PassTickets, specify the group's generic LU name as the value of the LINKNAME columns.

3. Map the location to the LU names of the data sharing members to be accessed.

   For each member, insert the LU name of the member into the LUNAME column of the requester's SYSIBM.LULIST table.

Example

The following example provides sample SQL statements for enabling member-routing access. It also shows the results of those statements in the form of table excerpts. This example assumes that a remote data sharing group exists with a Db2 location name of DB2A and a generic LU name of LUDB2A, and three members with names LUDB2AR, LUDB2B, and LUDB2C. Only members DB2B and DB2C with LU names LUDB2B and LUDB2C, respectively, are to be accessed.

The following SQL statements update the CDB of a Db2 for z/OS requester to use member-routing access:

1. This statement defines the location and uses the group generic LU name to identify different sets of members:

   GUPI

   ```
   INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME)
     VALUES ('DB2A', 'LUDB2A');
   ```

   GUPI

2. This statement defines the location's security and mode requirements for conversations with the members:

   GUPI

   ```
   INSERT INTO SYSIBM.LUNAMES (LUNAME)
      VALUES ('LUDB2A');
   ```

   GUPI

3. These statements map the location name to the two members of the group to be accessed:

   GUPI

   ```
   INSERT INTO SYSIBM.LULIST (LINKNAME, LUNAME)
      VALUES ('LUDB2A', 'LUDB2B');
   ```

   ```
   INSERT INTO SYSIBM.LULIST (LINKNAME, LUNAME)
      VALUES ('LUDB2A', 'LUDB2C');
   ```

   GUPI

The following table shows an example excerpt of the SYSIBM.LOCATIONS table after the update.

Table 11. Location aliases for a remote data sharing group in a Db2 requester's SYSIBM.LOCATIONS table. Not all columns are shown.

LOCATION	LINKNAME	TPN
DB2A	LUDB2A	

The following table shows an example excerpt of the SYSIBM.LUNAMES table after the update.

Table 12. Location aliases in a Db2 requester's SYSIBM.LUNAMES table. Not all columns are shown.

LUNAME	GENERIC
LUDB2A	

The following table shows an example excerpt of the SYSIBM.LULIST table after the update.

Table 13. LU names of the members that are associated with each location alias in a Db2 requester's SYSIBM.LULIST table. Not all columns are shown.

LINKNAME	LUNAME
LUDB2A	LUDB2B
LUDB2A	LUDB2C

Related concepts

Update the BSDS with the DSNJU003 utility

DSNJU003 can be used to update the information stored in the bootstrap data sets of members of data sharing groups.

Configuring a Db2 requester to use group-generic access

You can update the CDB of a Db2 requester to use group-generic access.

Procedure

To enable group-generic access to a remote data sharing group:

1. Identify the generic LU name of the remote group. Insert the group's generic LU name into the LUNAME column of the requester's SYSIBM.LUNAMES table. Do not include in this table the real LU names of the members of the group.
2. Map the generic LU name to the group's Db2 location name. Insert the generic LU name and the Db2 location name of the group into the LINKNAME and LOCATION columns of the requester's SYSIBM.LOCATIONS table.

Example

The following example provides sample SQL statements for enabling group-generic access. It also shows the results of those statements in the form of table excerpts. This example assumes that a remote data sharing group exists with a Db2 location name of DB2R and a generic LU name of LUDSGA.

The following SQL statements update the CDB of a Db2 for z/OS requester to use group-generic access:

1. This statement identifies the group's generic LU name:

   GUPI

   ```
   INSERT INTO SYSIBM.LUNAMES (LUNAME, GENERIC)
      VALUES ('LUDSGA', 'Y');
   ```

   GUPI

2. This statement maps the group's generic LU name to the group's Db2 location name.

   GUPI

   ```
   INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME)
      VALUES ('DB2R', 'LUDSGA');
   ```

   GUPI

The following table shows an example excerpt of the SYSIBM.LUNAMES table after the update.

Table 14. Generic LU name of a remote data sharing group in a Db2 requester's SYSIBM.LUNAMES table. Not all columns are shown.

LUNAME	GENERIC
LUDSGA	Y

The following table shows an example excerpt of the SYSIBM.LOCATIONS table after the update.

Table 15. Mapping the generic LU name of a remote data sharing group to its Db2 location name in a Db2 requester's SYSIBM.LOCATIONS table. Not all columns are shown.

LOCATION	LINKNAME	TPN
DB2R	LUDSGA	

Switching from group-generic to member-specific access

To switch from group-generic to member-specific access, you must break the affinity between systems.

About this task

Member-specific access provides several benefits over group-generic access, including better workload balancing and two-phase commit support. However, before a requester that was using group-generic access can take advantage of these benefits by switching to member-specific access, you need to break any affinity that might exist between the requester and the server. This section describes how to break the affinity between systems.

If a requester uses group-generic access to connect to a data sharing group with two-phase commit support enabled on both subsystems, VTAM records information in the coupling facility about which member of the group is involved in the communication. This information is required to ensure that future VTAM sessions are directed to the same group member, thus providing access to the correct member log for resolution of indoubt threads. When switching the requester from using group-generic access to member-specific access, you must break this affinity between the systems.

Procedure

To break the affinity between systems:

1. Shut down the network connections between the requester and the data sharing group.

   Some ways to do this include:

   - Using the VTAM command VARY NET,INACT,ID=*luname*
   - Entering the Db2 command STOP DDF
2. From an active member of the data sharing group, issue the RESET GENERICLU command.

   Issue this command from the member with the VTAM affinity to the requester whose information is being deleted.

   - If the requester is configured to pass its own, real LU name to the data sharing group, *netid* is the net ID of the requester, and *luname* is the real LU name of the requester.
   - If the requester is configured to pass its generic LU name, *netid* is the net ID of the requesting data sharing group, and *luname* is the generic LU name of the requesting data sharing group.
3. Change the requester's CDB to ensure that the requester uses member-specific access from this point forward.

   - Populate the requester's SYSIBM.LULIST table.

     Make sure that the GENERIC column of the requester's SYSIBM.LUNAMES table contains a value of N for the row that is associated with the data sharing group.
   - Delete existing generic LU names from the SYSIBM.LUNAMES table.
4. Re-enable the network connections between the requester and the data sharing group.

Unlike group-generic access, which uses generic LU names, member-specific access uses real LU names. When switching the requester from using group-generic access to member-specific access, remember to delete existing generic LU names.

Related tasks

Configuring a Db2 requester to use member-routing access
You can update the CDB of a Db2 requester to use member-routing access.

Related reference

-RESET GENERICLU (Db2) (Db2 Commands)

Preventing a member from processing requests

Transparently to users, you can prevent one or more members of a data sharing group from handling DDF requests while still letting those members make DDF requests.

Procedure

To prevent a member from handling DDF requests:

- Use one of the following options:
  - Set the MAX REMOTE ACTIVE option of installation panel DSNTIPE to 0 for that member.
  - Dynamically change the value of the MAXDBAT subsystem parameter in macro DSN6SYSP to 0 for that member.

  The MAX REMOTE ACTIVE installation panel option and the MAXDBAT subsystem parameter specify the maximum number of database access threads (DBATs) that can be active concurrently. By setting the value of either to 0, you restrict DDF server activity on the affected member. Subsequent connection requests are directed to those members whose MAX REMOTE ACTIVE option or MAXDBAT parameter value are greater than 0. Any work that is already in progress by the affected member continues, but new requests are directed to other members of the data sharing group.

Related reference

MAX REMOTE ACTIVE field (MAXDBAT subsystem parameter) (Db2 Installation and Migration)

Update the BSDS with the DSNJU003 utility

DSNJU003 can be used to update the information stored in the bootstrap data sets of members of data sharing groups.

You can use the DSNJU003 (change log inventory) utility to update the following information related to Db2 data sharing. This information is stored in the bootstrap data sets (BSDSs) of members of data sharing groups:

- Distributed Data Facility (DDF)

  Updates the LOCATION, and other DDF related information, in the BSDS.

  If you use this statement to insert new values into the BSDS, you must include at least the SNA, TCP/IP information, or both, and any elements you want to remove from the BSDS in the DDF statement. To update an existing set of values, you need to include only those values that you want to change. The DDF record cannot be deleted from the BSDS after it has been added, it can only be modified.
- DRDA port (PORT=*port*)

  The DRDA port is the TCP/IP port number that is used by the DDF to accept incoming connection requests. 446 is the recommended DRDA port.

  If you change this value for one member of the group, you must change it for all members. Db2 requires that all members of a data sharing group use the same, well-known port number to receive incoming connection requests.
- Generic LU name (GENERIC=*gluname*)

The generic LU name represents all the members of a data sharing group, in either an SNA network or a TCP/IP network.

If you change this value for one member of the group, you must change it for all members. All members of a data sharing group must specify the same name.

- GRPIPV4

  Identifies a constant IPV4 address to be associated with the data sharing group for which this DDF is a member.

  It is used for the purposes of accepting incoming connection requests that can be serviced by any member of the data sharing group. This address must be entered in dotted decimal form. An associated IPV4 subsystem/member address (see IPV4) must also be specified in order to identify the IP address associated to this specific member of the group. If an IP address is not specified, Db2 will automatically determine the IP address from TCP/IP. It is strongly recommended that you refer to a Sysplex distributor owned distributing dynamic virtual IP address (DVIPA).

- GRPIPV6

  Identifies a constant IPV6 address to be associated with the data sharing group for which this DDF is a member.

  It is used for the purposes of accepting incoming connection requests that can be serviced by any member of the data sharing group. This address must be entered in colon hexadecimal form. An associated IPV6 subsystem/member address (see IPV6) must also be specified in order to identify the IP address associated to this specific member of the group. If an IP address is not specified, Db2 will automatically determine the IP address from TCP/IP. It is strongly recommended that you refer to a Sysplex distributor owned distributing dynamic virtual IP address (DVIPA).

- IPV4

  Identifies a constant IPV4 address to be associated to DDF for the purposes of accepting incoming connection requests to this specific subsystem only.

  This address must be entered in dotted decimal form. If an IP address is not specified, Db2 will automatically determine the IP address from TCP/IP. When Db2 is a member of a data sharing group, it is strongly recommended that you refer to a dynamic virtual IP address (DVIPA). A group IP address should also be specified.

- IPV6

  Identifies a constant IPV6 address to be associated to DDF for the purposes of accepting incoming connection requests to this specific subsystem only.

  This address must be entered in colon hexadecimal form. If an IP address is not specified, Db2 will automatically determine the IP address from TCP/IP. When Db2 is a member of a data sharing group, it is strongly recommended that you refer to a dynamic virtual IP address (DVIPA). A group IP address should also be specified.

- Location aliases (ALIAS=*alias-name:alias-port:alias-secport*)

  Location aliases can represent one, several, or all members of a data sharing group for member specific access, to enable clients to continue to connect to a member's old location name, and for defining subsets of data sharing groups.

  Location aliases can be defined without an *alias-port* or *alias-secport* value. Clients that use a location alias without an *alias-port* or *alias-secport* value receive information only about the member that processes the connection request in the list of servers. To distribute work requests across the members or a subset of a group, clients must be configured to use the Db2 group location name or location alias that is defined for a subset of members. For location aliases that are defined by the DNJU003 utility, *alias-port* and *alias-secport* values are ignored in non-data sharing environments.

  Important: You can define and manage only eight static location aliases by using DSNJU003 utility. However, you can also define additional dynamic location aliases by using the MODIFY DDF command. When a static alias is defined with the same name as an existing dynamic alias, Db2 uses only the

dynamic alias. Dynamic location aliases, unlike those defined by the DSNJU003 utility, can be added, modified, and deleted without stopping and restarting Db2 or DDF.

- Resynchronization port (RESPORT=*resport*)

  The resynchronization port is the TCP/IP port number that is used by the DDF to accept incoming DRDA two-phase commit resynchronization requests.

  Db2 requires that each member of a data sharing group in a TCP/IP network have a resynchronization port number that is unique within the Parallel Sysplex. In the event of a failure, this unique port number allows a requester to reconnect to the correct member so that units of work that require two-phase commit can be resolved.

- Secure port (SECPORT=*secport*)

  The secure port identifies the TCP/IP port number that is used by DDF to accept inbound secure DRDA connection requests.

  This value must be a decimal number between 0 and 65535, including 65535; zero indicates that DDF's secure connection support for TCP/IP is deactivated. If the secure port has not been specified, remote connections can still use the DRDA PORT, and use SSL on it, but DDF will not validate if the connection uses SSL protocol or not.

Important: Before updating any of the information related to Db2 data sharing, have all members check for the existence of indoubt threads. If any indoubt threads exist, resolve them before making any updates. To check for indoubt threads, use the DISPLAY THREAD command.

Related tasks

Initializing a TCP stack for use with both IPv4 and IPv6 addresses (Db2 Installation and Migration)

Related reference

DSNJU003 (change log inventory) (Db2 Utilities)

-DISPLAY THREAD (Db2) (Db2 Commands)

Chapter 5. Operating with data sharing

Most data sharing operations are accomplished by using commands to Db2 for z/OS.

Commands for data sharing environments

As you begin to operate your data sharing environment, you should understand how commands work in a data sharing environment.

Command routing

You can control operations on an individual member of a data sharing group from any z/OS console by entering commands prefixed with the appropriate command prefix.

For example, assuming you chose -DB1A as the command prefix for member DB1A, you can start a Db2 statistics trace on that member by entering this command at any z/OS console in the Parallel Sysplex:

GUPI

```
-DB1A START TRACE (STAT)
```

GUPI

Command routing requires that the command prefix scope is registered as S or X on the IEFSSN*xx* parmlib member.

You can also control operations on certain objects by using commands or command options that affect an entire group. These, also, can be entered from any z/OS console. For example, assuming that DB1A is active, you can start database XYZ by entering this command at any z/OS console in the Parallel Sysplex:

GUPI

```
-DB1A START DATABASE (XYZ)
```

GUPI

Related tasks

Registering the command prefixes, member group attachment name, and subgroup attachment name (Db2 Installation and Migration)

Command scope

Many commands that are used in a data sharing environment affect only the member for which they are issued.

The breadth of a command's impact is called the *scope* of that command. For example, a STOP DB2 command stops only the member identified by the command prefix. Such commands have *member scope*.

Other commands have *group scope* because they affect an object in such a way that all members of the group are affected. For example, a STOP DATABASE command, issued from any member of the group, stops that database for all members of the group. The following commands have group scope:

© Copyright IBM Corp. 1994, 2020

ALTER GROUPBUFFERPOOL (Db2)	FREE PLAN (DSN)
BIND PACKAGE (DSN)	MODIFY irlmproc,DIAG (z/OS IRLM)
BIND PLAN (DSN)	REBIND PACKAGE (DSN)
DCLGEN (DSN)	REBIND PLAN (DSN)
DISPLAY DATABASE (Db2)	REBIND TRIGGER PACKAGE (DSN)
DISPLAY GROUP (Db2)	START DATABASE (Db2)
DISPLAY GROUPBUFFERPOOL (Db2)	STOP DATABASE (Db2)
FREE PACKAGE (DSN)	

The following commands have either group or member scope, depending on the options that you specify:

ARCHIVE LOG (Db2)	START FUNCTION SPECIFIC (Db2)
DISPLAY FUNCTION SPECIFIC (Db2)	START PROCEDURE (Db2)
DISPLAY PROCEDURE (Db2)	START TRACE (Db2)
DISPLAY THREAD (Db2)	STOP FUNCTION SPECIFIC (Db2)
DISPLAY TRACE (Db2)	STOP PROCEDURE (Db2)
DISPLAY UTILITY (Db2)	STOP TRACE (Db2)
MODIFY irlmproc,SET (z/OS IRLM)	TERM UTILITY (Db2)
MODIFY irlmproc,STATUS (z/OS IRLM)	

All other commands have member scope. For information about the specific data sharing scope of each command, see the "Environment" sections in the description of each command.

Commands issued from application programs

You can enter commands from an application program that is attached to a Db2 subsystem through any of the attachment facilities: IMS, CICS, TSO, CAF, and RRSAF.

Commands that are entered in this way are executed by the Db2 subsystem to which the application program is attached. The application cannot send a command to a different Db2 subsystem.

Command authorization

Data sharing does not introduce any new techniques for establishing and checking authorization IDs.

Because all members of a data sharing group share the Db2 catalog, any ID has the same privileges and authorities on every member.It is your responsibility to use the same connection or sign-on exit routines on every member of the data sharing group to avoid authorization anomalies.

Where messages are received

You can receive messages from all members at a single console.

Hence, a message must include a member identifier as well as a message identifier. The member's command prefix appears in messages to identify the source of a message.

Effect of data sharing on sequence number caching

Db2 always assigns sequence numbers in order of request.

Members in a data sharing environment can use the CREATE SEQUENCE statement with the CACHE option to request a "chunk" of sequence numbers. These numbers become the members local cache. Since each member assigns cache from its local chunk, numbers may not be assigned in order across the data sharing group. Each value is guaranteed to be unique.

For example, the following CREATE SEQUENCE statement results in sequence SEQ1 being defined as a block of 20 cache values numbered 1 through 20:

GUPI

```
CREATE SEQUENCE SEQ1 START WITH 1 INCREMENT BY 1 CACHE 20
```

GUPI

Because each data sharing group member gets its own sequenced block of cache values, all values assigned to that member are from that block. If member DB2A requests the first cache value for SEQ1, it is assigned value 1. If member DB2B requests the next cache value, it is assigned value 21. When member DB2A requests another cache value, it is assigned value 2, and when member DB2B requests another cache value, it is assigned value 22. When a member's block of cache values is exhausted, the next available block of cache values is allocated.

In data sharing systems, if cache value sequence must be assigned in strict numeric order, then the NOCACHE option of the CREATE SEQUENCE statement must be specified.

Related reference

CREATE SEQUENCE (Db2 SQL)

Starting a data sharing member

To start members of a Db2 data sharing group, you must enter a START DB2 command for each member of the group.

If this is the first startup for the group, **you must start the originating member first**.

Impact of command prefix scope: If Db2 is installed with a command prefix scope of STARTED (the default and recommended value), you must issue the START DB2 command from the z/OS system on which you want to start Db2 or you must route the command to that z/OS system. For example, to route the START DB2 command to the z/OS system on which you want to start DB1A, issue the following command:

```
ROUTE MVS1,-DB1A START DB2
```

After Db2 is started, you can issue all other commands from any z/OS system in the Parallel Sysplex, and the commands are routed to the appropriate member.

Stopping a data sharing member

Stop individual members of a Db2 data sharing group by using the STOP DB2 command.

This option speeds up shutdown because Db2 bypasses castout and associated cleanup processing in the group buffer pools. Consider specifying CASTOUT(NO) when you stop an individual member of a data sharing group for maintenance.

Related reference

-STOP DATABASE (Db2) (Db2 Commands)

States of connections and structures after stopping Db2

When Db2 allocates its coupling facility structures, it specifies a disposition for the structures and for connections to the structures after a normal or abnormal termination.

When you display the structures, you can see different states for the connections and structures based on how the disposition is defined and whether Db2 was stopped normally or shut down abnormally.

Normal shutdown

After a normal shutdown of Db2, the coupling facility structures specify different states for the connections and structures.

The following table summarizes the information that you see after a normal termination.

Table 16. States of structures and connections after normal Db2 termination

Structure	Connections	State
SCA	None	Allocated
Lock	Failed-persistent [1]	Allocated
Group buffer pools	None with CASTOUT(YES); Failed-persistent with CASTOUT(NO) [2]	Unallocated with CASTOUT(YES); Allocated with CASTOUT(NO)

Notes:

1. If a member has no retained locks, its failed-persistent connection to the lock structure is removed when it shuts down. If this is the last member to shut down, the connection remains in a failed-persistent state.
2. If castout failure occurs during shutdown, group buffer pool connections show as failed-persistent, even though Db2 terminates normally.

Abnormal shutdown

After an abnormal shutdown of Db2, the coupling facility structures specify different states for the connections and structures.

The following table summarizes the information that you see after an abnormal termination.

Table 17. States of structures and connections after abnormal Db2 termination

Structure	Connections	State
SCA	None	Allocated
Lock	Failed-persistent [1]	Allocated
Group buffer pools	Failed-persistent	Allocated

Note:

1. If a member has no retained locks, its failed-persistent connection to the lock structure is removed when it shuts down. If this is the last member to shut down, the connection remains in a failed-persistent state.

Submitting work to be processed

The methods that you use to submit work in a non-data sharing environment do not need to change for data sharing. However, you might find it to your advantage to use group attachment names and subgroup attachment names to direct jobs.

Group attachments and subgroup attachments

Utilities and applications that use TSO, batch, DL/I batch, the RRSAF, the CAF, or DB2I (Db2 Interactive) to connect to Db2 have two methods for specifying the member to which they want to connect.

The first method is to specify the name of the subsystem. The second method is to use the group attachment name, which acts as a generic name for all the members of a data sharing group. The group attachment name can be used in place of the name of the Db2 subsystem that runs on the z/OS system from which the job was submitted. If you use a group attachment name, you can also use a subgroup attachment name to provide more organization and control over connection requests.

If you specify a group attachment name, the utility or application does not need to be sensitive to a particular member, which makes it easier to move jobs around the Parallel Sysplex as needed. The utility or application connects to the first active group member that it finds. However, when the name of a subsystem is the same as the group attachment name, the utility or application attempts to connect to

the member first. If that member is not started and group attachment name processing is not disabled, the utility or application attempts to connect to the next group member that is active. (NOGROUP is the group override and is available for CAF and RRSAF.)

If you do not explicitly specify a subsystem name or group attachment name, Db2 uses DSN as the name of the intended subsystem. As with any application program, make sure you are accessing the set of Db2 libraries with the correct DSNHDECP programming defaults.

Related concepts

Group attachment names and subgroup attachment names (Db2 Installation and Migration)

Related tasks

Preparing an application to run on Db2 for z/OS (Db2 Application programming and SQL)

CICS and IMS applications with Db2 data sharing

You can specify the group attachment name when running CICS and IMS applications for data sharing.

CICS and IMS applications must be aware of the particular member to which they are attached so they can resolve indoubt units of recovery in the event of a failure. IMS Version 7 or later allows IMS-dependent regions to use the Db2 group attachment name to direct jobs to a data sharing group. CICS Transaction Server Version 2.2 or later allows you to use the CICS RESYNCMEMBER=YES option to handle indoubt units of work.

Related tasks

Running CICS application programs (Db2 Administration Guide)

Online utility jobs in data sharing environments

If you specify a group attachment name, you can also specify a subgroup attachment name. When you submit a utility job, you must specify either the group attachment name or the name of the member to which the utility is to attach.

For example, if you specify the group attachment name, the EXEC statement might look like the following statement:

```
//stepname EXEC PGM=DSNUTILB,PARM='group-attach-name,[uid],[utproc]'
```

If you do not use the group attachment name, the utility job must run on the z/OS system where the specified Db2 subsystem is running. Ensure that the utility job runs on the appropriate z/OS system. You must use one of several z/OS installation-specific statements to make sure this happens. These include:

- For JES2 multi-access spool (MAS) systems, insert the following statement into the utility JCL:

  ```
  /*JOBPARM SYSAFF=cccc
  ```

  where *cccc* is the JES2 name. You can specify an asterisk (SYSAFF=*) to indicate that the job should run on the system from which it was submitted.
- For JES3 systems, insert the following statement into the utility JCL:

  ```
  //*MAIN SYSTEM=(main-name)
  ```

  where *main-name* is the JES3 name.

Your installation might have other mechanisms for controlling where batch jobs run, such as the use of job classes.

How to stop and restart utilities

In a data sharing environment, you can use the TERM UTILITY command to stop the execution of an active utility only from the member on which the utility is active. You can then use this same command from any active member of the data sharing group to release the resources that are associated with the stopped utility. If a member fails while a utility is executing, you must restart Db2 on either the same or another z/OS system before stopping the utility. For remote site recovery from a disaster at the local site,

you can stop utilities that were active at the local site from any restarted member of the group at the remote site.

Recommendation: Define all work data sets that are used by utilities on shared disks. Doing so allows you to restart a utility on any member because all members are able to access all required data sets. Use the same utility ID (UID) to restart the utility. UIDs are unique within a data sharing group. If a member fails while a utility is executing, you must restart Db2 on either the same or another z/OS system before restarting the utility.

How to alter a REORG job

In a data sharing environment, you can use the ALTER UTILITY command to alter the REORG utility only from the member on which the utility is active. This is true for non-data sharing environments as well.

Related reference

JES2 Control Statements (MVS JCL Reference)
JES3 Control Statements (MVS JCL Reference)

Stand-alone utility jobs

Db2 stand-alone utilities (such as DSN1COPY) run as z/OS jobs that have no direct connection to Db2 services. Therefore, a data sharing group has no indication that one of these utilities is running.

In a data sharing environment, if a table space has inter-Db2 read/write interest, its most recently updated pages might be in the coupling facility and a stand-alone utility might not be running with current data. If it is important that the data is in a consistent state, you must stop or quiesce the table space. Also, the data must not be in the recovery pending (RECP) or group buffer pool recovery pending (GRECP) state nor have any logical page list entries. Use the DISPLAY DATABASE command with the RESTRICT option to find out if there are exception statuses for a given table space or index.

Monitoring the group

Monitoring data sharing groups includes tasks such as monitoring the database, and monitoring information about the group, structures, and group buffer pools.

Obtaining information about the group

You can use the DISPLAY GROUP command to obtain information about a data sharing group.

To obtain general information about all members of a particular group, use the DISPLAY GROUP command, shown here:

GUPI

```
-DB1A DISPLAY GROUP
```

The command can be issued from any active member of the group, and it displays the output that is shown below.

```
DSN7100I -DB1A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNDB10 ) CATALOG LEVEL(111) MODE(C   )
           PROTOCOL LEVEL(2)  GROUP ATTACH NAME(DB10)
------------------------------------------------------------------------
DB2                                    DB2 SYSTEM    IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
-------- --- ----   --------  -------- --- --------  ----   --------
DB1A       1 DB1A   -DB1A     ACTIVE   111 MVSA      DJ1A   DB1AIRLM
DB1B       2 DB1B   -DB1B     ACTIVE   111 MVSB      DJ1B   DB1BIRLM
DB1C       3 DB1C   -DB1C     ACTIVE   111 MVSC      DJ1C   DB1CIRLM
DB1D       4 DB1D   -DB1D     FAILED   111 MVSD      DJ1D   DB1DIRLM
DB1E       5 DB1E   -DB1E     QUIESCED 111 MVSE      DJ1E   DB1EIRLM
DB1F       6 DB1F   -DB1F     ACTIVE   111 MVSF      DJ1F   DB1FIRLM
DB1G       7 DB1G   -DB1G     ACTIVE   111 MVSG      DJ1G   DB1GIRLM
------------------------------------------------------------------------
SCA   STRUCTURE SIZE:     1024 KB, STATUS= AC,   SCA IN USE:    11 %
LOCK1 STRUCTURE SIZE:     1536 KB
NUMBER  LOCK ENTRIES:      262144
NUMBER  LIST ENTRIES:        7353, LIST ENTRIES  IN USE:           0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

Figure 20. Output of DISPLAY GROUP command

The figure above shows the following information:

- The Db2 group name and group release level
- The group attachment name
- The member names and release levels
- The command prefix for each member
- The status of each member (ACTIVE, QUIESCED with or without additional conditions, or FAILED)
- The Db2 release level of each member
- The name of the z/OS system where each member is running (or was last running, in cases where the member is not active)
- The names of the IRLM subsystems to which members are connected. For more information about the IRLM data sharing group, use the z/OS command F *irlmproc*,STATUS,ALLI
- The procedure names of the connected IRLMs
- The SCA structure size and the percentage currently in use
- Lock structure size

  The display also shows the following information:

  – The maximum number of lock entries possible for the lock table
  – The maximum number of modify lock list entries and how many of those list entries are currently in use

GUPI

Related concepts

Avoiding false contention

The Db2 coupling facility lock structure has two parts: a lock table, which is used to determine whether inter-Db2 read/write interest exists on a particular resource, and a list of the update locks that are currently held.

Related reference

-DISPLAY GROUP (Db2) (Db2 Commands)

Related information

DSN7100I (Db2 Messages)

Obtaining information about structures

Use the z/OS command D XCF,STR to display information about coupling facility structures and policy information.

Related information

z/OS MVS System Commands

Displaying all structures

The z/OS D XCF,STR command displays summary information about all coupling facility structures in the policy.

The command has the following syntax:

```
D XCF,STR
```

The command produces the following output:

```
D XCF,STR
IXC359I  15.57.52  DISPLAY XCF
STRNAME            ALLOCATION TIME   STATUS
DSNCAT_GBP0      11/27/2005 15:57:47 DUPLEXING REBUILD NEW STRUCTURE
                                     DUPLEXING REBUILD
                                     REBUILD PHASE: DUPLEX ESTABLISHED

DSNCAT_GBP0      11/27/2005 15:29:43 DUPLEXING REBUILD OLD STRUCTURE
                                     DUPLEXING REBUILD
DSNCAT_GBP1           --       --    NOT ALLOCATED
DSNCAT_GBP11          --       --    NOT ALLOCATED
DSNCAT_GBP16K0        --       --    NOT ALLOCATED
DSNCAT_GBP16K1        --       --    NOT ALLOCATED
DSNCAT_GBP22          --       --    NOT ALLOCATED
DSNCAT_GBP32K         --       --    NOT ALLOCATED
DSNCAT_GBP33          --       --    NOT ALLOCATED
DSNCAT_GBP44          --       --    NOT ALLOCATED
DSNCAT_GBP8K0         --       --    NOT ALLOCATED
DSNCAT_GBP8K1         --       --    NOT ALLOCATED
DSNCAT_LOCK1     11/27/2005 15:26:16 ALLOCATED
DSNCAT_SCA       11/27/2005 15:26:13 ALLOCATED
ISTGENERIC       11/27/2005 15:11:09 ALLOCATED
IXCLINKS              --       --    NOT ALLOCATED
LOCK2                 --       --    NOT ALLOCATED
RRSSTRUCT1            --       --    NOT ALLOCATED
```

Figure 21. Output from D XCF,STR command

Displaying information about specific structures

The STRNAME keyword of the z/OS D XCF, STR command displays detailed information about specific structures.

The following command displays information about the duplexed group buffer pool GBP0 for group DSNCAT:

```
D XCF,STR,STRNAME=DSNCAT_GBP0
```

The command produces the following output:

```
D XCF,STR,STRNAME=DSNCAT_GBP0
IXC360I  11.13.38  DISPLAY XCF
STRNAME: DSNCAT_GBP0
 STATUS: REASON SPECIFIED WITH REBUILD START:
           OPERATOR INITIATED
           DUPLEXING REBUILD
         REBUILD PHASE: DUPLEX ESTABLISHED
  POLICY SIZE    : 32768 K
  POLICY INITSIZE: 5000 K
  REBUILD PERCENT: N/A
  DUPLEX         : ALLOWED
  PREFERENCE LIST: LF01     CACHE01
  EXCLUSION LIST IS EMPTY

 DUPLEXING REBUILD NEW STRUCTURE
 -------------------------------
  ALLOCATION TIME: 04/12/1999 11:13:31
  CFNAME         : CACHE01
  COUPLING FACILITY: SIMDEV.IBM.EN.ND0200000000
                     PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 5120 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : B2162049 D1E56F02
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 DUPLEXING REBUILD OLD STRUCTURE
 -------------------------------
  ALLOCATION TIME: 04/12/1999 11:12:51
  CFNAME         : LF01
  COUPLING FACILITY: SIMDEV.IBM.EN.ND0100000000
                     PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 5120 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : B2162023 45B3CB06
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DB2_V71A         02 00020001 UTEC277  V71ADBM1 002F ACTIVE NEW,OLD
 DB2_V71B         01 00010001 UTEC277  V71BDBM1 0033 ACTIVE NEW,OLD
```

Figure 22. Output from D XCF,STR,STRNAME command

Obtaining information about group buffer pools

Db2 provides a DISPLAY GROUPBUFFERPOOL command that is useful for displaying statistical information about group buffer pool use.

However, you can also obtain information about group buffer pools by using the z/OS command D XCF,STR.

Depending on the options you choose for the command, the display output contains the following information:

- A list of all connections to the group buffer pools. For duplexed group buffer pools, only one set of connections exists for both instances of the group buffer pool. For example, if there are three connections to duplexed structure GBP0, there are just three connections, not six connections.
- Statistical reports on group buffer pool use, either by a specific member or by the whole group. Some statistical information is also available for the secondary allocation of a duplexed group buffer pool.

Related reference

Displaying information about specific structures
The STRNAME keyword of the z/OS D XCF, STR command displays detailed information about specific structures.

Group buffer pool monitoring with the DISPLAY GROUPBUFFERPOOL command

Use the DISPLAY GROUPBUFFERPOOL command to display information about group buffer pools.

Database monitoring options

Db2 attempts to automatically recover logical page list page sets.

For page sets or partitions that have LPL or GRECP status and that are not automatically recovered, either start the page set or partition using the START DATABASE command with SPACENAM and ACCESS (RW) or (RO), or run the RECOVER utility. If any table or index space that is required to confirm START DATABASE command authority is unavailable, INSTALL SYSADM might be required to issue the command.

Related concepts

Recovery of pages on the logical page list

Db2 attempts to automatically recover pages when they are added to the logical page list. Recovered pages are deleted from the logical page list when recovery processing successfully completes.

Related reference

-START DATABASE (Db2) (Db2 Commands)

Data sharing status descriptions

The group buffer pool recovery pending (GRECP) and logical page list (LPL) statuses are specific to Db2 data sharing. These statuses can appear on the output from the DISPLAY DATABASE command.

GUPI

GRECP

"Group buffer pool recovery pending." The group buffer pool was lost, and the changes that are recorded in the log must be applied to the page set. When a page set is placed in the GRECP state, Db2 sets the starting point for the merge log scan to the log record sequence number (LRSN) of the last complete group buffer pool checkpoint.

Db2 automatically recovers GRECP page sets when the group buffer pool is defined with AUTOREC (YES) and at least one member was connected when the failure occurred.

LPL

"Logical page list." Some pages were not read from, or written to, the group buffer pool because of some failure, such as a channel failure between the group buffer pool and the processor. Or perhaps pages could not be read from, or written to, disk because of a transient disk problem.

GUPI

Pages in error and the logical page list

The *logical page list (LPL)* contains a list of pages (or a page range) that could not be read or written for some reason. Reasons could include transient disk read and write problems that can be fixed without redefining new disk tracks or volumes.

Specific to data sharing, the LPL also contains pages that could not be read or written for "must-complete" operations, such as a commit or a restart, because of some problem with the coupling facility. For example, pages can be added if there is a channel failure to the coupling facility or disk, or if locks are held by a failed subsystem, disallowing access to the page that you want.

The logical page list is kept in the SCA and is thus accessible to all members of the group.

If an application tries to read data from a page that is on the logical page list, it receives a "resource unavailable" SQLCODE. In order to be accessible, pages in the logical page list must first have their logged changes applied to the page set.

To verify the existence of logical page list entries, issue the DISPLAY DATABASE command. The LPL option of DISPLAY DATABASE can then be used to see the specific list of pages:

GUPI

```
-DB1A DIS DB(DSNDB01) SPACENAM(*) LIMIT(*) LPL ONLY
```

Output similar to the following is produced:

```
DSNT360I -DB1A
************************************************************
DSNT361I -DB1A *  DISPLAY DATABASE SUMMARY
               *  GLOBAL LPL
DSNT360I -DB1A
************************************************************
DSNT362I -DB1A    DATABASE = DSNDB01  STATUS = RW
               DBD LENGTH = 8000
DSNT397I -DB1A
NAME     TYPE PART STATUS             LPL PAGES
-------- ---- ---- ------------------ ------------------
DBD01    TS        RW,LPL,GRECP       000001,000004,00000C,000010
----                                   000039-00003C
SYSLGRNX TS        RW,LPL,GRECP       000000-FFFFFF
******* DISPLAY OF DATABASE DSNDB01  ENDED      **********************
DSN9022I -DB1A DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

GUPI

Automatic LPL recovery is attempted when pages are added to the logical page list. Automatic LPL recovery is also performed when you start the table space, index, or partition by using the START DATABASE command with ACCESS(RW) or ACCESS(RO).

Note: When a table space or partition is placed in the LPL because undo processing is needed for a NOT LOGGED table space, automatic LPL recovery is not initiated at restart time or during rollback processing. A -START DATABASE command identifying this table space will have no effect on the LPL status.

Physical read and write errors

In some previous releases, Db2 recorded physical read and write errors in an error page range. This remains true. however, if a read or write problem is of undetermined cause, the error is first recorded in the logical page list. If recovery from the logical page list is unsuccessful, the error is then recorded in the error page range.

Related concepts

Recovery of pages on the logical page list
Db2 attempts to automatically recover pages when they are added to the logical page list. Recovered pages are deleted from the logical page list when recovery processing successfully completes.

Related tasks

Displaying the logical page list (Db2 Administration Guide)

Removing pages from the logical page list (Db2 Administration Guide)

Related reference

-DISPLAY DATABASE (Db2) (Db2 Commands)

-START DATABASE (Db2) (Db2 Commands)

Related information

DSNT361I (Db2 Messages)

Identifying and resolving retained locks for Db2 failures

When a lock is used to allow an object to be changed it is called a *modify* lock. Modify locks are kept in a list in the coupling facility lock structure to allow for recovery in case a data sharing member fails. If a member fails, the modify locks become *retained locks*, which are held until the failed member is restarted.

Procedure

To identify and resolve retained locks for a Db2 member, take to following actions:

1. To identify retained locks, issue a DISPLAY DATABASE command with the LOCKS option.

```
-DB1A DISPLAY DATABASE(TESTDB) LOCKS ONLY
```

The letter R in the LOCKINFO field indicates a retained lock.

```
NAME     TYPE PART STATUS             CONNID   CORRID        LOCKINFO
-------- ---- ---- ------------------ -------- ----------    --------
⋮
TBS43     TS  01   RW                                        R-IX,PP
                   MEMBER NAME DB2A
```

GUPI

2. To resolve the retained locks, restart the failed member normally.

   Db2 resolves and removes retained locks that are held by that member with the full data integrity control. For more information, see "Restarting Db2 after termination in a data sharing environment" on page 111.
3. If you cannot restart the failed member, and it has retained locks that are severely affecting transactions on other members, consider taking the following actions:

   - Defer the restart processing of the objects that have retained locks. When you defer restart processing, the pages that locks are protecting are placed in the logical page list. Those pages are still inaccessible. However, this approach has the advantage of removing any retained page set P-locks, which have the potential of locking out access to an entire page set.
   - Cold start the failed member. This approach causes Db2 to purge the retained locks, but **data integrity is not protected.** When the locks are released after the cold start, Db2 looks at data whose status is unclear.
   - Use the command, MODIFY *irlmproc*,PURGE,*db2name*. Like a cold start, this command causes Db2 to purge the retained locks and with this method, **data integrity is not protected.**
   - Restart the failed member in light mode (restart light). Restart light is not recommended for a restart in place. It is intended for a cross-system restart in the event of a failed z/OS to quickly recover retained locks. Restart light enables Db2 to restart with a minimal storage footprint and then terminate normally after the locks are released.

Related concepts

Active and retained locks for Db2data sharing members

Active locks, retained locks, and retained utility locks can be held by a failed member. Locks that are held by a failed member can affect the availability of data to the other members of the group that are still running Db2 applications.

Restart light

The LIGHT(YES), LIGHT(NOINDOUBTS), and LIGHT(CASTOUT) clauses of the -START DB2 command restart a member in "light" mode.

Related tasks

Deferring restart processing (Db2 Administration Guide)

Related reference

-DISPLAY DATABASE (Db2) (Db2 Commands)

Determining the data sharing member on which SQL statements run

Use the special register CURRENT MEMBER to determine the member of a data sharing group on which SQL statements execute.

The data type of CURRENT MEMBER is CHAR(8). If necessary, the member name is padded on the right with blanks so that its length is 8 bytes. The value of CURRENT MEMBER is a string of blanks when the application process is connected to a Db2 subsystem that is *not* a member of a data sharing group.

Example: To set the host variable MEM to the name of the current member, use one of the following statements:

GUPI

```
EXEC SQL SET :MEM = CURRENT MEMBER;
```

```
EXEC VALUES (CURRENT MEMBER) INTO :MEM
```

GUPI

Controlling connections to remote systems in a data sharing environment

Controlling DDF connections in a data sharing environment is somewhat different from controlling DDF connections in a non-data sharing environment.

Starting and stopping DDF

You control the distributed data facility (DDF) on a member basis, not on a group-wide basis.

GUPI This gives you more granular control over DDF processing. For example, assume that you want to devote member DB1A to batch processing for some period of time without disrupting other connections. You can enter the following command to disallow any further distributed connections from coming into this member:

```
-DB1A STOP DDF MODE(QUIESCE)
```

To stop **all** DDF processing for the group, you need to issue the STOP DDF command for every member of the group. For example, you might need to do this when you change the SYSIBM.LOCATIONS table.

To allow for completion of CREATE, ALTER, DROP, GRANT, or REVOKE operations from remote requesters, issue the STOP DDF MODE(SUSPEND) command. GUPI

Monitoring connections to remote systems

The DISPLAY THREAD command can display information about all threads in the data sharing group. The DISPLAY LOCATION command shows thread information only for the member on which it is issued.

If a data sharing group is defined to have a generic LU, you must use a member's real LU name for *luwid* if you are requesting information by *luwid* (logical unit of work ID).

If the DISPLAY THREAD is being related to global transactions coordinated by *xid*, DISPLAY THREAD shows *xid* information. The thread is managed by an XA transaction manager, such as WebSphere®, which identifies the transaction with an *xid*. The *xid* is provided to allow correlation with the XA transaction manager. Db2 uses both the logical unit of work identifier, *luwid*, and the XA transaction identifier, *xid*, to coordinate and recover transactions.

When a remote Db2 subsystem issues a DISPLAY LOCATION command to obtain information about connections to a data sharing group, the output displays information about every LU at that location.

Related reference

-DISPLAY THREAD (Db2) (Db2 Commands)

-DISPLAY LOCATION (Db2) (Db2 Commands)

Resetting generic LU information

If you use a generic LU name to connect to a member of a data sharing group that is using two-phase commit, VTAM permanently records information in the coupling facility about which member of the group was involved in the communication.

This permanently recorded information is required to guarantee that future VTAM sessions are always directed to the same member, making it possible to provide access to the correct member log for resolution of indoubt threads.

At times, you might need to break that affinity between the member and the other system. You would need to do this, for example, if you want to use member-specific access, or if you want to remove a member from the data sharing group.

To break this affinity, issue the RESET GENERICLU command from an active member of the data sharing group. You must issue this command from the member with the VTAM affinity to the particular LU. Here is an example of a command that removes information about USIBMSTODB22 from DB1A:

GUPI

```
-DB1A RESET GENERICLU(LUDB22)
```

GUPI

Take great care when using this command because it can potentially cause the specified partner LUs to connect to different members on later sessions. This can cause operational problems if indoubt threads exist at a partner LU when this command is issued.

Related reference

-RESET GENERICLU (Db2) (Db2 Commands)

Logging environment for data sharing

In a data sharing environment, the member subsystems still maintain separate recovery logs. Each manages its own active and archive log data sets and records those in its own bootstrap data set (BSDS).

The shared communications area (SCA) in the coupling facility contains information about all members' BSDSs and log data sets. In addition, every member's BSDS contains information about other members' BSDS and log data sets, in case the SCA is not available.

The following figure illustrates a typical logging environment.

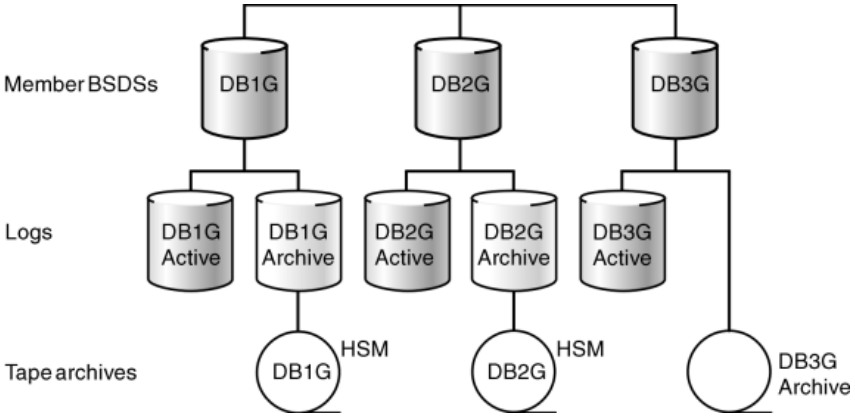


Figure 23. Member BSDSs and logs in the data sharing environment

The impact of archiving logs in a data sharing group

In data sharing, the Db2 RECOVER utility needs log records from every member that has changed the object that needs to be recovered.

If the logs are archived, the impact on RECOVER depends on how the log data sets are archived:

- Archive to disk without DFSMShsm to migrate the data sets from disk to tape.

  No major impact on performance is experienced for this type of archive. But you need enough disk space to hold archive logs, and the disk devices must be shared (accessible) by all members in a data sharing group. Because DFSMShsm or its equivalent is not used, you must manage disk space carefully to avoid running out of space.

- Archive to disk with DFSMShsm.

  DFSMShsm can do automatic space and data availability management among storage devices in a system. DFSMShsm can migrate the archive on disk to less expensive storage (such as tape), and recall the archive back to disk when needed.

Using DFSMShsm, a particular RECOVER job needs only one tape unit to recall migrated archive data sets. If the archive data sets have been migrated, recovery time might be adversely affected, because the recalls of the migrated archive data sets are done one at a time from the member running the RECOVER job. For example, a RECOVER job started on DB1A might need log data sets from DB1A, DB2A, and DB3A. DB1A sends the recall requests to DFSMShsm one at a time for the tapes needed for recovery.

- Archive to tape.

  The RECOVER job needs at least one tape unit for each member whose archived log records are to be merged. (More might be needed if you run more than one recover at the same time for different partitions of a partitioned table space.) Therefore, **do not archive logs from more than one system to the same tape**.

  Recommendation: For data sharing, avoid using tape archive logs for data recovery.

  If you must archive to tape, ensure that the value for READ TAPE UNITS on installation panel DSNTIPA for each member is high enough to handle anticipated recovery work. For example, if you have eight members, each member should specify at least eight tape drives. You need more if you run more than one recovery job at the same time on a given member, or if multiple members run recovery jobs at the same time.

  If there are not enough tape units to do the recovery, Db2 can possibly deadlock. If this happens, use the command SET ARCHIVE to increase the number of tape units that can be used.

  Also, ensure that you specify 0 for the DEALLOC PERIOD parameter on installation DSNTIPA to avoid making an archive tape inaccessible to other members of the data sharing group. (If you intend to run all RECOVER jobs from a single member, this suggestion does not apply.)

Related concepts

How recovery works in a data sharing group

In case you need to recover data, it can be helpful to understand how data recovery works in a data sharing environment.

How to avoid using the archive log

A recovery cycle for a table space is defined by how often its image copy is taken.

A RECOVER job needs the following copies and records:

- The latest image copy
- The optional incremental copies
- The log records since the last incremental or image copy
- Archived log records, if all of the log records since the last incremental image copy are **not** still in active log data sets

Keeping all of the log records since the last incremental image copy in the active log data sets is to your advantage, because reading log records from the active log is much faster than reading from archive logs, even if those archives are on disk.

Several ways exist to minimize the need to use the archive log:

- Increase the total active log space.

  The total amount of active log space is the number of active log data sets multiplied by its size. Currently, Db2 limits the maximum number of active log data sets to 93. Because each member can have up to 93 active log data sets, the total number of active log data sets is effectively increased by the number of members in a data sharing group.

  The size of an active log data set is up to 4 GB but is usually limited by the size of a tape cartridge. Most installations prefer not to have an archived data set on more than one tape volume.

  Recommendation: Do not use tape compression for the Db2 archive log, because Db2 needs to read the log backwards for backout operations. Performance for backout can be severely degraded if tape compression is used.

Tape compression is not Db2 data compression, which compresses the data portion of a Db2 log record. With Db2 data compression, the log record header is not compressed and causes no extra performance degradation for backward scans.

- Increase the frequency of incremental image copies.

  Because only the log records generated since the last incremental image copy are needed for recovery, the more often you make incremental image copies, the less chance there is that archive log records will be needed. Weigh this consideration against the time it takes to make the incremental image copies and the effects on SQL transactions.

- Make sure applications commit frequently.

  To avoid having to mount an archive log for backing out changes, ensure that applications are committing frequently. Consider using the UR CHECK FREQ parameter or the UR LOG WRITE CHECK parameter of installation panel DSNTIPL to help you track when applications are not committing as frequently as your site guidelines suggest.

Related tasks

Improving recovery performance
You can improve your recovery performance by taking more frequent image copies.

Recovering data in data sharing

The procedures for data recovery are fundamentally the same for data sharing as for non-data-sharing environments. Data sharing involves one catalog, but many logs and BSDSs. In addition to disk and cache controllers, the coupling facility adds a possible point of failure and requires appropriate procedures for data recovery.

In planning for data sharing, consider having more than one coupling facility. If an SCA or lock structure failure occurs, recovery for that structure can proceed automatically if a second coupling facility is available. For more information and detailed recommendations, see Planning for coupling facility availability for data sharing.

How recovery works in a data sharing group

In case you need to recover data, it can be helpful to understand how data recovery works in a data sharing environment.

Logs that are needed for recovery

Assume that three members of a data sharing group are making updates to table space TS1, as shown in the figure below.

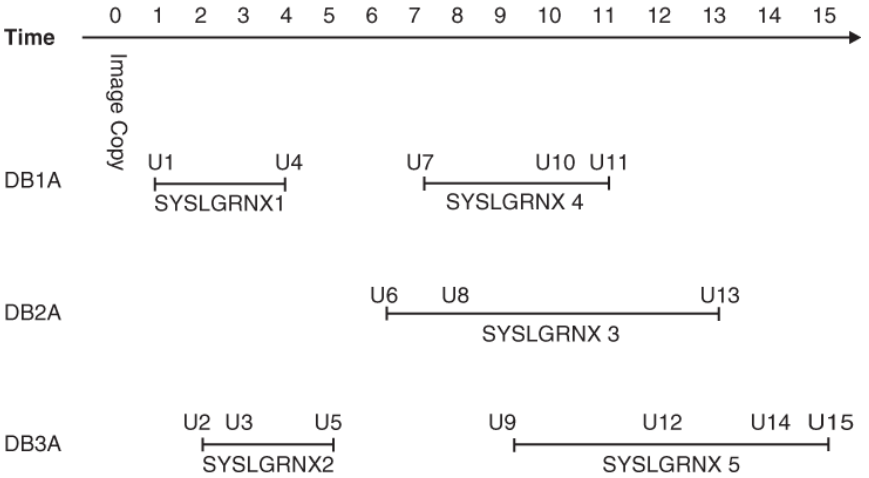


Figure 24. Three members of a data sharing group updating table space TS1

The following is the sequence of overlapping updates leading up to the time of recovery:

1. DB1A updates TS1 between Time 1 and 4 (SYSLGRNX record 1) with two updates (U1 and U4).
2. DB3A updates TS1 between Time 2 and 5 (SYSLGRNX record 2) with three updates (U2, U3, and U5).
3. DB2A updates TS1 between Time 6 and 13 (SYSLGRNX record 3) with three updates (U6, U8, and U13).
4. DB1A updates TS1 again between Time 7 and 11 (SYSLGRNX record 4) with three updates (U7, U10, and U11).
5. DB3A updates TS1 again between Time 9 and 15 (SYSLGRNX record 5) with four updates (U9, U12, U14 and U15).

Now, assume that you want to recover TS1 to time 9. The full image copy taken at T0 is used as the recovery base. All the SYSLGRNX records mentioned previously are selected to determine the log ranges of each system for the log scan. Updates U1 through U9 are applied in order.

How log records are applied

Db2 can access the logs of other Db2 subsystems in the group and merge them in sequence. The log record sequence number (LRSN) identifies the log records of a data sharing member. Multiple row insert might produce duplicate LRSNs for changes to the same page. The following figure illustrates the structure of the log record.

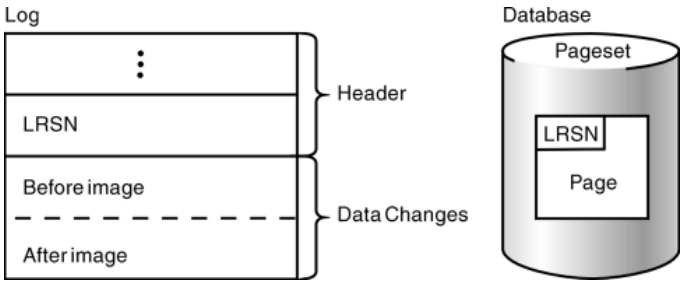


Figure 25. The log and LRSN in the data sharing environment

The log record header contains an LRSN. The LRSN is a 6-byte or 10-byte value that is greater than or equal to the timestamp value truncated to 6 or 10 bytes. This value also appears in the page header. During recovery, Db2 compares the LRSN in the log record to the LRSN in the page header before applying changes to disk. If the LRSN in the log record is larger than the LRSN on the data page, the change is applied.

Related concepts

The extended 10-byte RBA and LRSN in Db2 11 (Db2 for z/OS What's New?)

Expanded RBA and LRSN log records (Db2 for z/OS What's New?)

How RBA and LRSN values are displayed (Db2 Administration Guide)

Related tasks

What to do before RBA or LRSN limits are reached (Db2 Administration Guide)

Related information

Reading log records (Db2 Administration Guide)

Improving recovery performance

You can improve your recovery performance by taking more frequent image copies.

About this task

You might want to limit this activity by determining which table spaces most need fast recovery. The following guidelines are provided as a starting point to help you determine how often you must do incremental image copies. As with a single subsystem, doing frequent image copies can help you avoid using the archive log for recovery.

Procedure

Use the following guidelines to increase the frequency of image copies for each member of the data sharing group. Use the output of the DSNJU004 (print log map) utility for each member.

a. Find the starting timestamp of the active log data set with the lowest STARTRBA.
b. Find the ending timestamp of the active log data set with the highest ENDRBA.
c. Calculate the time interval:

```
time_interval = end_TIMESTAMP - start_TIMESTAMP
```

d. Calculate the interval at which to perform incremental image copies:

```
interval of copy = time_interval * (n-1) / n
```

n is the number of active log data sets.

e. Take the smallest interval for the group and, to account for fluctuations of log activity, decrease the interval by 30%. (30% is an arbitrary figure; you might have to adjust this interval based on your system's workload.)

This is the recommended interval for doing incremental image copies. If the interval is too small to be realistically accomplished, consider increasing the size or number of active log data sets.

Periodically run the MERGECOPY utility with incremental image copies. The RECOVER utility attempts to mount tape drives for all the incremental image copies at the same time. If it runs out of tape drives, it switches to log apply. MERGECOPY merges what it can and then mounts more incremental image copies.

Related reference

COPY (Db2 Utilities)

Recovery options for data sharing environments

Use the RECOVER utility to recover to currency or to a prior point in time.

Recovery to currency

This process is used to recover from damaged data by restoring from a backup and applying all logs to the current time. The recovery process operates similarly in the data sharing and non-data-sharing environments. Image copies are restored and subsequently updated based on changes recorded in the logs. In the data sharing group, multiple member logs are read concurrently in log record sequence.

Point-in-time recovery

This process discards potentially corrupt data by restoring a database to a prior point of consistency. Corrupt data might result from a logical error. The following point-in-time recovery options are available:

TORBA
This option is used to recover to a point on the log defined by a receive byte address (RBA). In a data sharing environment, TORBA can only be used to recover to a point prior to defining the data sharing group.

TOLOGPOINT
This option is used to recover to a point on the log defined by a log record sequence number (LRSN). The TOLOGPOINT keyword must be used when you recover to a point on the log after the data sharing group was defined. However, you can also use TOLOGPOINT in a non-data-sharing environment.

The LRSN is a 10-byte hexadecimal number derived from a store clock timestamp. LRSNs are reported by the DSN1LOGP stand-alone utility.

TOCOPY
This option is used to recover data or indexes to the values contained in an image copy without subsequent application of log changes.

Successful recovery clears recovery pending conditions and brings data to a point of consistency. In a data sharing environment, all pages associated with the recovered data entity are removed from the group buffer pool and written to disk.

Recovery using a system-level backup

The RECOVER utility can recover an object or a list of objects using a system-level backup, that is created with the BACKUP SYSTEM utility, as a recovery base.

Related reference
RECOVER (Db2 Utilities)

System-level point-in-time recovery

Use the BACKUP SYSTEM and RESTORE SYSTEM online utilities to perform system-level point-in-time recovery.

Db2 provides the following solutions for system-level recovery:

Recovery to a specific point in time.
This point in time is either between two backup times or between the last backup time and the current time. After the appropriate volume copies are restored, the outstanding logs are applied to the databases to recover the data to the designated point in time.

Recovery to the point in time of a backup
This recovery restores the appropriate volume copies of the data and logs. The logs are used only to back out inflight transactions on restart.

Remote site recovery from disaster at a local site
This recovery is determined by what you keep at the remote site. For example, in addition to offsite tapes of backups, current logs might have been transmitted electronically. If so, the current logs can be applied to the databases after the data and logs are restored from the offsite tapes.

The BACKUP SYSTEM utility

The BACKUP SYSTEM online utility provides the following types of backups:

- A data-only system backup, which contains only the databases. The RESTORE SYSTEM online utility uses these backups to recover the system to an arbitrary point in time.
- A full system backup, which contains both logs and databases. The RESTORE SYSTEM online utility uses these backups to recover the system to the point in time when the copy was taken by using normal Db2 restart recovery.

The RESTORE SYSTEM utility

The RESTORE SYSTEM utility recovers a Db2 subsystem to a prior point in time. The utility restores volume copies that are provided by the BACKUP SYSTEM online utility with the DATA ONLY option. After restoring the data, you can use the RESTORE SYSTEM online utility to recover to an arbitrary point in time.

Related concepts
Point-in-time recovery with system-level backups (Db2 Administration Guide)
Related reference
RESTORE SYSTEM (Db2 Utilities)
BACKUP SYSTEM (Db2 Utilities)

Recovering a data sharing group in case of a disaster

To develop a procedure for recovering a data sharing group by using a remote site, you can use the disaster recovery procedure as a base. In most cases, you must complete those steps for each member of the data sharing group.

As an alternative, you can set up the remote data sharing group as a tracker site. The advantage of a tracker site is that it dramatically reduces the amount of time that is needed for takeover if a disaster

occurs at the primary site. The disadvantage is that the tracker site must be dedicated to shadowing the primary site. You cannot use the tracker site to perform transactions of its own.

Related concepts

Recovery procedure differences
The procedure for data sharing at the recovery site differs in that extra steps exist for cleaning out old information in the coupling facility.

Related tasks

Creating essential disaster recovery elements (Db2 Administration Guide)
Recovering from disasters by using a tracker site (Db2 Administration Guide)

Configuration of the recovery site

The hardware configuration can be different at the recovery site, as long as it supports data sharing.

The recovery site must have a data sharing group that is identical to the group at the local site. It must have the same name, the same number of members, and the members must have the same names as those at the local site. The CFRM policies at the recovery site must define the coupling facility structures with the same names as those at the local site (although the sizes can be different).

You can run the data sharing group on as few or as many z/OS systems as you want.

Conceptually, there are two ways to run the data sharing group at the recovery site. Each way has different advantages that can influence your choice:

- Run a multi-system data sharing group.

  The local site is most likely configured this way. You have a Parallel Sysplex containing many CPCs, z/OS systems, and Db2 subsystems. This configuration requires a coupling facility, the requisite coupling facility channels, and the Sysplex Timer.

  Using a multi-system data sharing group at the recovery site, you have the same availability and growth options that you have at the local site.
- Run a single-system data sharing group.

  In this configuration, you centralize all of your Db2 processing within a single, large processor, such as an IBM System z10 or subsequent 64-bit z/Architecture® processor. As the following figure shows, you must *install* a multi-member data sharing group. After the group starts up, you shut down all but one of the members and access data through the remaining member.

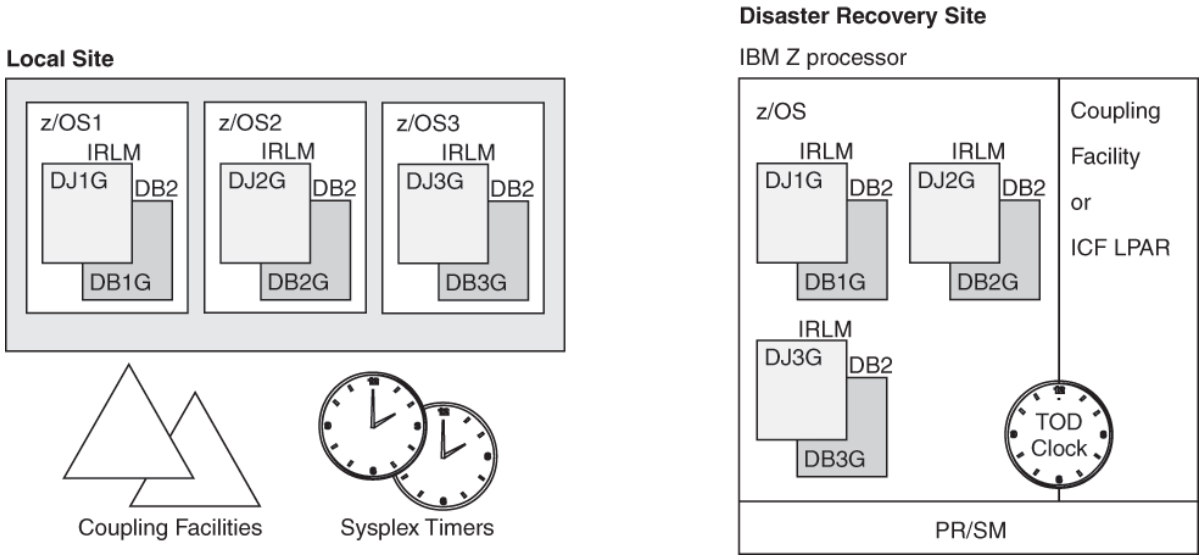


Figure 26. Example of local and disaster recovery site configurations

With a single-system data sharing group, you lose the availability benefits of the Parallel Sysplex, but the group has fewer hardware requirements:

– The Sysplex Timer is not needed; you can use the CPC's time-of-day clock.
– You can use any available coupling facility configuration for the recovery site system, including Integrated Coupling Facilities (ICFs).

  With a single-system data sharing group, there is no longer inter-Db2 read/write interest, and the requirements for the coupling facility are as follows:

  - A lock structure (which can be smaller)
  - An SCA

Group buffer pools are not needed for running a single-system data sharing group. However, you do need to have at least small group buffer pools for the initial startup of the group. Db2 allocates them and uses them to do its damage assessment processing. When you are ready to do single-system data sharing, you can remove the group buffer pools by stopping all members and then restarting the member that is handling the workload at the disaster recovery site.

Related reference

zSeries Processor Resource/Systems Manager Planning Guide

What to send to the recovery site

You must send to the recovery site the same information as for single-system remote recovery: logs and BSDSs, image copies, and so on.

To prepare the logs for the remote site, you have three options:

- Use the command ARCHIVE LOG with the MODE (QUIESCE) option to ensure a point of consistency for each of the log data sets. If the quiesce is not successful, the command fails and the logs are not archived.

  At the recovery site (just as for non-data-sharing disaster recovery), the ENDRBA value you use for restarting each member is the end RBA +1 of the latest archive log data set from each member in the data sharing group.
- Use the command ARCHIVE LOG SCOPE(GROUP). This version of the command does not ensure a point of consistency for all members' logs, but the logs are archived on each of the active members of the data sharing group. You can use the ENDLRSN option of the DSNJU003 (change log inventory) utility on the remote site to truncate all logs to the same point in time.

  To determine the truncation value, look at the print log map output from the latest copies of the archived BSDS.

  Another way to determine the truncation value is to ship the SYSLOG containing message DSNJ003I with your archive log data sets to the recovery site. This message is issued when archive log data sets are created (when you issue the ARCHIVE LOG command). The message contains the starting and ending LRSN and RBA values for the archive log data set. For example, the following messages appear when the command ARCHIVE LOG SCOPE(GROUP) is issued from one of the members at the local site:

```
DSNJ003I -DB1A DSNJOFF3 FULL ARCHIVE LOG VOLUME
DSNAME=DSNC510.ARCHLOG1.A0000003, STARTRBA=000001C68000,
ENDRBA=000001D4FFFF, STARTLRSN=ADFA208AA36C, ENDLRSN=AE3C45273A77,
UNIT=SYSDA, COPY1VOL=SCR03, VOLSPAN=00, CATLG=YES
```

```
DSNJ003I -DB2A DSNJOFF3 FULL ARCHIVE LOG VOLUME
DSNAME=DSNC518.ARCHLOG1.A0000001, STARTRBA=000000000000,
ENDRBA=0000000D6FFF, STARTLRSN=ADFA00BB70FB, ENDLRSN=AE3C45276DD7,
UNIT=SYSDA, COPY1VOL=SCR03, VOLSPAN=00, CATLG=YES
```

  Compare the ending LRSN values for all members' archive logs, and choose the lowest LRSN as the truncation point. For the two members shown previously, the lowest LRSN is AE3C45273A77. To get the last complete log record, you must subtract 1 from that value, so you would enter AE3C45273A76 as the ENDLRSN value in the CRESTART statement of the DSNJU003 utility for each of the members at the remote site. All log records with a higher LRSN value are discarded during the conditional restart.
- Use the command SET LOG SUSPEND if you are using the IBM RAMAC Virtual Array (RVA) storage control with the peer-to-peer remote copy (PPRC) function or Enterprise Storage Server® Flashcopy to

create point-in-time backups of entire Db2 subsystems for faster recovery at a remote site. Using either of these methods to create a remote copy requires the suspension of logging activity, which prevents database updates. The SUSPEND option of the SET LOG command suspends logging and update activity until a subsequent SET LOG command with the RESUME option is issued.

Important: Make sure that all members of the group are active when you archive the logs. If you have a quiesced member whose logs are necessary for a recovery base at the disaster recovery site, you must start that member with ACCESS(MAINT) to archive its log.

For read-only members, Db2 periodically writes a log record to prevent those members from keeping the LRSN value too far back on the log.

Related tasks

Performing remote site recovery from a disaster at a local site (Db2 Administration Guide)

Recovery procedure differences

The procedure for data sharing at the recovery site differs in that extra steps exist for cleaning out old information in the coupling facility.

Old information exists in the coupling facility from any practice startups. In addition, you must prepare each subsystem (rather than just a single system) for a conditional restart.

Related tasks

Performing conditional restart (Db2 Administration Guide)

Recovery of pages on the logical page list

Db2 attempts to automatically recover pages when they are added to the logical page list. Recovered pages are deleted from the logical page list when recovery processing successfully completes.

In some situations—such as a channel failure between the group buffer pool and the processor, a transient disk problem, or Db2 in restart—the automatic LPL recovery processor cannot be initiated, or the recovery cannot complete. In these situations, the pages are kept in the logical page list. The following tasks show several ways to perform manual LPL recovery:

- Start the object with access (RW) or (RO). This command is valid even if the table space is already started.

  When you issue the command START DATABASE, Db2 issues messages to indicate the status of the recovery, including:

  - DSNI005I, which indicates an error
  - DSNI006I, which indicates the start of recovery
  - DSNI021I, which indicates successful completion
  - DSNI022I, which provides the status of recovery operations
  - DSNI051I, which indicates that second pass log apply has started
- Run the RECOVER utility on the object.

  The only exception to this is when a logical partition of a type 2 nonpartitioned index has both LPL and RECP status. If you want to recover the logical partition using RECOVER INDEX with the PART keyword, you must first use the command START DATABASE to clear the logical page list pages.
- Run the LOAD utility with the REPLACE option on the object.
- Issue an SQL DROP statement for the object.
- Use the utility REPAIR SET with NORCVRPEND. This can leave your data in an inconsistent state.
- Use START DATABASE ACCESS(FORCE). This can leave your data in an inconsistent state.

None of the items in the preceding list work if retained locks are held on the object. You must restart any failed member that is holding those locks.

When a table space or partition is placed in the LPL because undo processing is needed for a NOT LOGGED table space, either at restart time or during rollback processing, automatic LPL recovery is not initiated. A -START DATABASE command identifying this table space will have no effect on the LPL status

Related concepts

Pages in error and the logical page list

The *logical page list (LPL)* contains a list of pages (or a page range) that could not be read or written for some reason. Reasons could include transient disk read and write problems that can be fixed without redefining new disk tracks or volumes.

Related tasks

Displaying the logical page list (Db2 Administration Guide)

Removing pages from the logical page list (Db2 Administration Guide)

How Db2 recovers from coupling facility failures

Failures of the coupling facility can be classified into two main groups: structure failures and connectivity failures.

Important: To ensure high availability for Db2 data sharing and reduce the time and effort for recovering from failures, follow the practices described in Planning for coupling facility availability for data sharing.

Structure failures occur when structures such as the group buffer pool, SCA, or lock structures in the coupling facility become damaged, but the coupling facility continues to operate.

A *connectivity failure* is a total failure of the coupling facility. They can be caused by the following types of problems:

- Problems with the attachment of the z/OS system to the coupling facility
- A power failure that affects the coupling facility but leaves one or more z/OS systems running
- Deactivation of the coupling facility partition
- Failure of the coupling facility control code
- Failure of the coupling facility CPC or LPAR

The following table summarizes how Db2 recovers from structure and connectivity failures for different structures and configurations.

Failure type	Recovery if duplexed	Recovery if simplexed
Group buffer pool structure	The failing structure is deallocated, and processing continues with the running structure. This recovery is usually fast and unnoticeable.	Recovery from the log can occur manually, as the result of a START DATABASE command, or if the group buffer pool is defined with the AUTOREC(YES) option, it can occur automatically.
SCA and lock structure	The failing structure is deallocated, and processing continues on the running structure.	Db2 uses information that is contained in its virtual storage to quickly rebuild the structures. Recovery of a simplexed SCA is usually fast and has a minimal impact on performance. Recovery of a simplexed lock structure is also fast in normal cases, but the performance impact depends on the number of held locks at the time of the failure. Db2 can rebuild a simplexed SCA and lock structure in the same coupling facility or in an alternate coupling

Failure type	Recovery if duplexed	Recovery if simplexed
		facility, assuming that the following conditions are true: • You specified the alternate coupling facility in the CFRM policy preference list. • You allocated enough storage in the alternate coupling facility to rebuild the structures there. If Db2 fails to rebuild the SCA and lock structure from virtual storage, all active members in the group terminate abnormally, and you must perform a group restart to recover the necessary information from the logs.
Connectivity	Db2 switches to the structure with good connectivity	Db2 rebuilds simplexed structures on the alternate coupling facility that is specified in the CFRM policy. Recovery of a simplexed SCA is fast and has little or no impact on performance. The performance of the recovery of a lock structure depends on the number of modify locks in the lock structure at failure time, but it is also fast in most cases. However, recovery of simplexed group buffer pool structures is very disruptive to the system. In rebuilding these structures, Db2 attempts to allocate storage on the alternate coupling facility. Db2 uses the current size of the structure for the initial size of the structure on the alternate coupling facility. If Db2 cannot allocate the storage for the SCA or lock structure, the rebuild fails. If z/OS cannot allocate the storage for the group buffer pools, the changed pages are written to disk.

Another type of failure is a *channel failure*, which occurs when the channel connecting a CPC to a coupling facility no longer operates. Without dual channels (sometimes called links), a channel failure is more likely to occur than a failure in the coupling facility. Losing connectivity to the SCA or lock structure can bring that particular member down, unless you specify duplexing or an alternative coupling facility in the CFRM policy preference list.

Related concepts

When the coupling facility structure fails (z/OS MVS Setting Up a Sysplex)

Coupling facility failure scenarios

Coupling facility failure scenarios include connectivity failures and structure failures.

Connectivity failure to the SCA or lock structure

The following table summarizes what happens when there are connectivity failures to the SCA.

Table 18. Summary of connectivity failures to the SCA

Active SFM policy?	Weighted loss < REBUILDPERCENT?	Db2 response	Operational response
No	Not applicable	Each affected member issues: `DSN7501A 00F70600` Db2 comes down. Connection is deleted; structure remains allocated.	Options include: • Fix problem • Restart failed member on system that is connected to coupling facility • Manually rebuild onto another coupling facility.
Yes	Yes	Each affected member issues: `DSN7501A 00F70600` Db2 comes down. Connection is deleted; structure remains allocated.	Options include: • Fix problem • Restart failed member on system that is connected to coupling facility • Manually rebuild onto another coupling facility.
Yes	No	Automatic rebuild. `DSN7503I`	None needed.

The following table summarizes what happens when there are connectivity failures to the lock structure.

Table 19. Summary of connectivity failures to the lock structure

Active SFM policy?	Weighted loss < REBUILDPERCENT?	Db2 response	Operational response
No	Not applicable	Each affected member issues: `DXR136I 00E30105` Db2 comes down. Connection is failed-persistent; structure remains allocated.	Options include: • Fix problem • Restart failed member on system that is connected to coupling facility • Manually rebuild onto another coupling facility.
Yes	Yes	Each affected member issues: `DXR136I 00E30105` Db2 comes down. Connection is failed-persistent; structure remains allocated.	Options include: • Fix problem • Restart failed member on system that is connected to coupling facility • Manually rebuild onto another coupling facility.

Table 19. Summary of connectivity failures to the lock structure (continued)

Active SFM policy?	Weighted loss < REBUILDPERCENT?	Db2 response	Operational response

Yes	No	`DXR143I`
		Automatic rebuild.
		`DXR146I`
			None needed.

Connectivity failure to non-duplexed group buffer pools

The following table summarizes what happens when there are connectivity failures to the group buffer pools.

Table 20. Summary of connectivity failures for non-duplexed group buffer pools

Connectivity lost from	Weighted loss < REBUILDPERCENT?	Db2 response	Operational response

100% of members connected to a group buffer pool that is defined with GBPCACHE(YES)	No	Each affected member issues:
		`DSNB303E`
		`DSNB228I`
		Add pages to logical page list, if necessary.
		`DSNB250E`
		`DSNB314I* rsn=100%`
		`DSNB304I*`
		Damage assessment, GRECP page sets.
		`DSNB320I`
		`DSNB321I`
		`DSNB353I`
		`DSNI006`
		`DSNI021`
		`DSNI051`
		`DSNB354I`
		`DSNB305I*`
			None needed if the group buffer pool is defined with AUTOREC(YES), and Db2 successfully recovers the page sets. Otherwise, enter START DATABASE commands.

100% of members connected to a group buffer pool that is defined with GBPCACHE(NO)	No	Automatic rebuild.
		`DSNB331I`
		`DSNB338I`
			None needed.

Table 20. Summary of connectivity failures for non-duplexed group buffer pools (continued)

Connectivity lost from	Weighted loss < REBUILDPERCENT?	Db2 response	Operational response

A subset of members connected to some or all group buffer pools.	Yes	Each affected member issues:
		`DSNB303E`
		`DSNB228I`
		`DSNB313I rsn=LOSSCONN`
		Quiesce applications that use the group buffer pool. Add pages to logical page list if necessary.
		`DSNB250E`
		`DSNB311I, DSNB312I`
		Disconnect GBPx failed-persistent.
		`DSNB309I`
			Options include:
			• Fix the problem.
			• Manually rebuild the structure onto another coupling facility.
			• Stop and restart Db2 on a system that is connected to the coupling facility.
			Enter START DATABASE commands to recover logical page list entries.

A subset of members connected to some or all group buffer pools.	No	Automatic rebuild.
		`DSNB331I`
		`DSNB332I`
		`DSNB333I` [1]
		`DSNB338I`
			None needed.

Note:

1. Issued by the structure owner.

Structure failures

The following table summarizes what happens to each structure if there is a structure failure. This information is for non-duplexed group buffer pools.

Table 21. Summary of structure failures, by structure type

Failed structure	Db2 response	Operational response

SCA	`DSN7502I`
	Automatic rebuild
	`DSN7503I`
		None needed.

Lock structure	`DXR143I`
	Automatic rebuild
	`DXR146I`
		None needed.

Table 21. Summary of structure failures, by structure type (continued)

Failed structure	Db2 response	Operational response

Group buffer pool that is defined with GBPCACHE(YES)	`DSNB228I`
	`DSNB314I rsn=STRFAIL`
	Add pages to logical page list, if necessary.
	`DSNB250E`
	Damage assessment, GRECP page sets.
	`DSNB304I`
	`DSNB320I DSNB321I`
	`DSNI006`
	`DSNI021`
	`DSNB305I`
		None needed if the group buffer pool is defined with AUTOREC(YES) and Db2 successfully recovers the page set. Otherwise, enter START DATABASE commands.

Group buffer pool that is defined with GBPCACHE NO	Automatic rebuild.
	`DSNB331I`
	`DSNB338I`
		None needed.

Connectivity failures for the duplexed group buffer pools

For duplexed group buffer pools, a failure response is the same for both a loss of structure and for lost connectivity, as shown in the following table.

Table 22. Summary of scenarios for both a loss of structure and lost connectivity for duplexed group buffer pools

Structure loss	Db2 response	Operational response

Primary	Switch to secondary in simplex mode
	`DSNB744I`
	`DSNB745I`
	If DUPLEX(ENABLED) then reduplexing is attempted.
		If the system did not automatically reduplex, correct the problem with the failed coupling facility. If you want to restart duplexing, use the z/OS SETXCF command.

Secondary	Revert to primary in simplex mode
	`DSNB743I`
	`DSNB745I`
	If DUPLEX(ENABLED) then reduplexing is attempted.
		If the system did not automatically reduplex, correct the problem. If you want to restart duplexing, use the SETXCF command.

Both (loss of structure or 100 % LOSSCONN)	Damage assessment, GRECP page sets.	None needed if the group buffer pool is defined with AUTOREC(YES) and Db2 successfully recovers the page set. Otherwise, enter START DATABASE commands.

Related concepts

Coupling facility recovery scenarios
Looking at a few problem scenarios might help if you need to recover a coupling facility.

Problem: Loss of group buffer pool structure (non-duplexed)

In your data sharing environment, you might encounter an error that is caused by the loss of a non-duplexed group buffer pool structure.

Problem: Loss of lock structure
In your data sharing environment, you might encounter an error that is caused by a loss of lock structure.

Problem: Loss of SCA structure
In a data sharing environment, you might encounter an error that is caused by a loss of SCA structure.

Coupling facility recovery scenarios
Looking at a few problem scenarios might help if you need to recover a coupling facility.

Problem: loss of coupling facility (CF)
You might encounter a failure that is treated by z/OS and Db2 as a total loss of connectivity to the coupling facility and coupling facility structures.

In these example scenarios, an active SFM policy is assumed.

Problem: Loss of group buffer pool structure (non-duplexed)
In your data sharing environment, you might encounter an error that is caused by the loss of a non-duplexed group buffer pool structure.

Symptom

Some or all of the following messages appear, depending on which structures Db2 tries to access:

```
DSNB303E -DB1A csect-name A LOSS OF CONNECTIVITY
WAS DETECTED TO GROUP BUFFER POOL gbpname.
```

```
DSNB228I csect-name GROUP BUFFER POOL gbpname
CANNOT BE ACCESSED FOR function
MVS IXLCACHE REASON CODE=reason
```

```
DSNB314I csect-name DAMAGE ASSESSMENT TO BE TRIGGERED FOR
GROUP BUFFER POOL gbpname REASON=100%LCON
```

```
DSNB250E csect-name A PAGE RANGE WAS ADDED TO THE
LOGICAL PAGE LIST
DATABASE NAME = dbn
SPACE NAME = spn
DATA SET NUMBER = dsno
PAGE RANGE = lowpg TO highpg
START LRSN = startlrsn
END LRSN = endlrsn
START RBA = startrba
```

System action

A group buffer pool failure restricts access to the data assigned to that group buffer pool; it does not cause all members of the group to fail. For a loss of connectivity to the group buffer pools, applications that need access to group buffer pool data are rejected with a -904 SQL return code. Any of the following reason codes can be returned:

00C20204
00C20205
00C20220

- Db2 puts the group buffer pool in "damage assessment pending" status. The following message appears:

  ```
  DSNB304I -DB1A  csect-name GROUP BUFFER POOL
  gbpname WAS SET TO 'DAMAGE ASSESSMENT PENDING' STATUS
  ```

- Db2 adds entries to the logical page list, if necessary.

- Db2 marks the affected table spaces, indexes, or partitions as group buffer pool recovery pending (GRECP) (DSNB320I or DSNB321I), indicates that the group buffer pool is recovering page sets (DSNB353I), and initiates recovery for the page set (DSNI006I). Message DSNI051I indicates the beginning of second pass GRECP recovery for the object.
- As each page set is recovered, the castout owner for the page set issues DSNI021I. After the last page set is recovered, any member that has issued message DSNB353I now issues message DSNB354I.
- After damage assessment is complete, the structure owner issues DSNB305I.

  The first new connection to the group buffer pool causes z/OS to reallocate the group buffer pool in the same or an alternate coupling facility as specified on the preference list in the CFRM policy.

System programmer action

Correct the coupling facility failure. For lost connectivity to group buffer pools, Db2 automatically recovers data that was in any group buffer pool defined with AUTOREC(YES). If the automatic recovery is not successful or if any pages remain in the logical page list after recovery, notify the database administrator to recover the data from the group buffer pool by using the command START DATABASE (*dbname*) SPACENAM (*spacename*) to remove the GRECP status. You must issue separate START DATABASE commands in the following order:

1. **DSNDB01**
2. **DSNDB06**

If any table or index space that is required for authorization checking is unavailable, Installation SYSADM or Installation SYSOPR authority is required to issue the START DATABASE commands.

Problem: Loss of group buffer pool structure (duplexed)
In your data sharing environment, you might encounter an error that is caused by the loss of a duplexed group buffer pool structure.

System action

The following messages will appear if the secondary group buffer pool structure is lost:

```
DSNB743 csect-name DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL gbpname
            FALLING BACK TO PRIMARY,
            REASON = STRFAILSEC
            DB2 REASON CODE = xxxxxxxxx
```

```
DSNB745 csect-name THE TRANSITION BACK TO SIMPLEX MODE HAS COMPLETED FOR
        GROUP BUFFER POOL gbpname
```

The following messages will appear if the primary group buffer pool structure is lost:

```
DSNB744 csect-name DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL gbpname
            SWITCHING TO SECONDARY,
            REASON = STRFAILPRI
            DB2 REASON CODE = xxxxxxxxx
```

```
DSNB745 csect-name THE TRANSITION BACK TO SIMPLEX MODE HAS COMPLETED FOR
        GROUP BUFFER POOL gbpname
```

System programmer action

No action is required at this time.

If there are 2 coupling facilities originally: When the CF that failed is brought back online, the structures will duplex themselves automatically if DUPLEX(ENABLED) was specified as recommended in the CFRM policy.

If there are more than 2 coupling facilities specified in the preference list of the CFRM policy for the structure: The structure will duplex itself using the remaining coupling facilities if DUPLEX (ENABLED) was specified in the structure statement of the CFRM policy.

After all coupling facilities have been restored to service, you should issue the SETXCF START,REALLOCATE at a low time of activity, to ensure that the group buffer pools are restored to their preferred coupling facilities.

Problem: Loss of lock structure
In your data sharing environment, you might encounter an error that is caused by a loss of lock structure.

Symptom

Locking requests are suspended until the lock structure is rebuilt. If the lock structure cannot be rebuilt, the following messages appears:

```
DXR136I irlmx HAS DISCONNECTED FROM THE DATA SHARING GROUP
```

```
DXR143I irlmx REBUILDING LOCK STRUCTURE BECAUSE IT HAS FAILED OR AN IRLM
        LOST CONNECTION TO IT
```

System action

DXR146I is issued for a successful rebuild of the lock structure. If the structure cannot be rebuilt, all active members of the group terminate abnormally with a 00E30105 abend code.

Important: If the lock structure cannot be rebuilt, the lost connectivity causes the members of the group to abend. In this case, a group restart is required.

System programmer action

You must fix the problem that is causing the loss of connectivity. If the problem is not an obvious one (such as a power failure), call IBM Support. See message DXR135E for the root cause of the problem and the corrective procedure.

After the problem is fixed, restart any failed members as quickly as possible to release retained locks. For the loss of lock structure, if the automatic rebuild occurred normally, processing can continue while you wait for the coupling facility problem to be fixed.

Related tasks
Planning for coupling facility availability for data sharing

Problem: Loss of SCA structure
In a data sharing environment, you might encounter an error that is caused by a loss of SCA structure.

Symptom

The following message appears:

```
DSN7501I -DB1A csect-name SCA STRUCTURE sca-structure-name CONNECTIVITY FAILURE.
```

Db2 suspends processing until the SCA is rebuilt, using information contained in Db2 memory.

If the SCA cannot be rebuilt, the following message appears:

```
DSN7504I -DB1A csect-name SCA STRUCTURE sca-structure-name REBUILD UNSUCCESSFUL.
```

System action

DSN7503I is issued for a successful rebuild of the SCA. If the rebuild is unsuccessful, all members of the group terminate abnormally with a 00F70600 abend code.

Important: If the SCA cannot be rebuilt, the lost connectivity causes the members of the group to abend. This can happen, for example, if both coupling facilities are volatile and lose power. In this case, a group restart is required.

To avoid situations in which a group restart is necessary, put structures in nonvolatile coupling facility structures.

System programmer action

You must fix the problem that is causing the loss of connectivity. If the problem is not an obvious one (such as a power failure), call IBM Support. Check the termination code for the reason that the rebuild was unsuccessful.

After the problem is fixed, restart any failed members as quickly as possible to release retained locks. For the loss of SCA structure, if the automatic rebuild occurred normally, processing can continue while you wait for the coupling facility problem to be fixed.

Related tasks

Planning for coupling facility availability for data sharing

Problem: a subset of members have lost connectivity (non-duplexed)

In a data sharing environment with simplexed structures, one or more members might lose connectivity to the coupling facility, while other members do not. This might happen if a link is detached between a system and the coupling facility.

This scenario assumes that the group buffer pool, lock, and SCA structures are simplexed, and have REBUILDPERCENT(1) specified in the CFRM policy, which is the recommended setting. With REBUILDPERCENT(1) specified, the group buffer pool, lock or SCA structures will automatically be rebuilt into an alternate coupling facility whenever a member loses connectivity to the structure.

Symptom

Some or all of the following messages appear, depending on which structures Db2 tries to access:

```
DSNB303E -DB1A csect-name A LOSS OF CONNECTIVITY
WAS DETECTED TO GROUP BUFFER POOL gbpname
```

```
DSNB331 csect-name REBUILD STARTED FOR GROUP BUFFER POOL gbpname
REASON = LOSSCON
```

```
DSN7501A -DB1A csect-name SCA STRUCTURE sca-structure-name
CONNECTIVITY FAILURE
```

```
DXR143Iirlm-subsys-name REBUILDING LOCK STRUCTURE
BECAUSE IT HAS FAILED OR AN IRLM LOST CONNECTION TO IT
```

```
DSN7503Icsect-name SCA STRUCTURE sca-strname
REBUILD SUCCESSFUL
```

System action

The structures are automatically rebuilt into the alternate coupling facility.

When Db2 rebuilds the group buffer pool, it writes changed pages from the group buffer pool to the alternate structure specified in the CFRM policy. If Db2 determines that there is not enough space to hold the changed pages, it casts the pages out to disk instead and the following messages are issued:

DSNB331I
DSNB332I
DSNB333I
DSNB338I

For the SCA structure, the following message is issued:

```
DSN7503Icsect-name SCA STRUCTURE sca-strname
REBUILD SUCCESSFUL
```

For the lock structure, the following message is issued:

```
DXR143Irlm-subsys-name REBUILDING LOCK STRUCTURE
BECAUSE IT HAS FAILED OR AN IRLM LOST CONNECTION TO IT
```

Problem: a subset of members have lost connectivity (duplexed)

In a data sharing environment with duplexed structures, one or more members might lose connectivity to the coupling facility, while other members do not. This problem might happen if a link is detached between a system and the coupling facility.

This scenario assumes that the group buffer pool, lock, and SCA structures are duplexed. For a duplexed structure, the system will automatically fall out of duplexing when a member loses connectivity to one of the structure instances, and the system will use the structure instance with full connectivity in a simplexed state.

If DUPLEX(ENABLED) is specified in the CFRM policy, then the system will attempt to automatically reestablish duplexing, if another coupling facility exists that has full connectivity and that coupling facility exists in the structure's preference list. If DUPLEX(ENABLED) is specified, but no preferred and connected coupling facility exists, then the system will attempt to automatically reestablish duplexing when full connectivity is restored to the original coupling facility.

Symptom

The following messages will appear if the secondary group buffer pool structure is lost:

```
DSNB743 csect-name DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL gbpname
            FALLING BACK TO PRIMARY,
            REASON = LOSSCONNSEC
            DB2 REASON CODE = xxxxxxxxx
```

```
DSNB745 csect-name THE TRANSITION BACK TO SIMPLEX MODE HAS COMPLETED FOR
        GROUP BUFFER POOL gbpname
```

The following messages will appear if the primary group buffer pool structure is lost:

```
DSNB744 csect-name DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL gbpname
            SWITCHING TO SECONDARY,
            REASON = LOSSCONNPRI
            DB2 REASON CODE = xxxxxxxxx
```

```
DSNB745 csect-name THE TRANSITION BACK TO SIMPLEX MODE HAS COMPLETED FOR
        GROUP BUFFER POOL gbpname
```

System action

The duplexed group buffer pool, lock, and SCA structures will fall back to simplex when either copy of the duplexed structure is lost.

System programmer action

No action is required at this time.

If there are 2 coupling facilities originally: When the CF that failed is brought back online, the structures will duplex themselves automatically if DUPLEX(ENABLED) was specified as recommended in the CFRM policy.

If there are more than 2 coupling facilities specified in the preference list of the CFRM policy for the structure: The structure will duplex itself using the remaining coupling facilities if DUPLEX (ENABLED) was specified in the structure statement of the CFRM policy.

After all coupling facilities have been restored to service, you should issue the SETXCF START,REALLOCATE at a low time of activity, to ensure that the group buffer pools are restored to their preferred coupling facilities.

Problem: allocation failure of the group buffer pool

One problem that you might encounter is an allocation failure of the group buffer pool.

Symptom

The following message appears:

```
DSNB301E -DB1A csect-name GROUP BUFFER POOL
gbpname CANNOT BE CONNECTED
DB2 REASON CODE =  reason1
MVS IXLCONN REASON CODE =  xxxx0C08
```

System action

Applications needing access to group buffer pool data are rejected with a -904 SQL return code (reason code 00C20204).

If the group buffer pool cannot be allocated in an alternate coupling facility as specified on the preference list of the CFRM policy, no inter-Db2 RW activity can exist on the table spaces, indexes, or partitions that are assigned to this buffer pool. If the group buffer pool that cannot be allocated is group buffer pool 0, no update activity can exist on the Db2 catalog and directory.

System programmer action

Use IFCID 0250 in performance class 20 to determine the reason for the allocation failure. If the trace indicates that the reason for the allocation failure is inadequate storage in the coupling facility, you can:

- Change the CFRM policy to decrease the amount of storage for the group buffer pool, or redefine that group buffer pool to a different coupling facility that has more storage.
- Have the database administrator reassign some of the table spaces or indexes that are using that group buffer pool to a different group buffer pool.

Related concepts

Changing the size of the group buffer pool

You can use two methods to change the size of the group buffer pool.

Problem: storage shortage in the group buffer pool

In a data sharing environment, you might encounter a problem that is caused by a storage shortage in the group buffer pool.

Symptoms

The following message appears when the group buffer pool is 75% full:

```
DSNB319A -DB1A csect-name THERE IS A SHORTAGE OF SPACE IN GROUP
BUFFER POOL gbpname
```

If you do not do anything to relieve the shortage, the following message appears when the group buffer pool is 90% full:

```
DSNB325A -DB1A csect-name THERE IS A CRITICAL SHORTAGE OF
          SPACE IN GROUP BUFFER POOL gbpname
```

If the group buffer pool is full, Db2 cannot write to it, and the following message appears:

```
DSNB228I csect-name GROUP BUFFER POOL gbpname
CANNOT BE ACCESSED FOR function
MVS IXLCACHE REASON CODE=xxxx0C17
```

Performance problems are evidence that the group buffer pool is not large enough. You can avoid having writes to the group buffer pool fail because of a lack of storage.

System action

Some critical functions that cannot be completed can cause one or more members of the group to come down with reason code 00F70609.

System programmer action

If you cannot increase the size of the group buffer pool, use the ALTER GROUPBUFFERPOOL command to decrease the castout thresholds. If decreasing the castout threshold negatively impacts performance, this should only be used as a temporary solution.

Related concepts

Group buffer pool size is too small

Pages in the group buffer pool need to be refreshed from disk more often because they are not in the group buffer pool.

Problem: storage shortage in the SCA

In a data sharing environment, you might encounter a problem that is caused by a storage shortage in the SCA.

Symptoms

The following message appears:

```
DSN7505I -DB1A csect-name THERE IS A SHORTAGE OF FREE STORAGE
          IN SCA STRUCTURE sca-structure-name.
```

If you do not do anything to reclaim space, such as recovering pages from the logical page list, the following message appears when the SCA is 90% full:

```
DSN7512A -DB1A csect-name THERE IS A CRITICAL SHORTAGE OF
          FREE STORAGE IN SCA STRUCTURE sca-structure-name.
```

System action

Some critical functions that cannot be completed can cause one or more members of the group to come down with reason code 00F70609.

System programmer action

Perform the following steps:

1. Reclaim space in the SCA by removing exception conditions. You can issue START DATABASE commands with the SPACENAM option, or use the RECOVER utility to remove pages from the logical page list.
2. Restart any failed members.

If these actions do not free up enough space, or if this problem continues to occur, you have the following options, depending on what level of z/OS and coupling facility you have.

- If **both** of the following conditions are true:
  - The SCA is allocated in a coupling facility with a CFLEVEL greater than 0.
  - The currently allocated size of the SCA is less than the maximum structure size as defined by the SIZE parameter of the CFRM policy.

  You can enter the following command to increase the size of the SCA:

```
SETXCF START,ALTER,STRNAME=DSNDB0A_SCA,SIZE=newsize
```

This example assumes that the group name is DSNDB0A, and that *newsize* is less than or equal to the maximum size defined in the CFRM policy for the SCA structure.

- If **any** of the following conditions are true:
  - The SCA is allocated in a coupling facility with CFLEVEL=0.
  - The allocated size of the structure is already at the maximum size defined by the SIZE parameter in the CFRM policy.

  Then you must:

  1. Increase the storage for the SCA in the CFRM policy SIZE parameter.
  2. Use the z/OS command SETXCF START,POLICY to start the updated policy.
  3. Use the following z/OS command to rebuild the structure:

     ```
     SETXCF START,REBUILD,STRNAME=DSNDB0A_SCA
     ```

- If **all** members are down, and you cannot increase the size of the SCA, you must do the following:
  1. Delete the SCA structure by using the following command:

     ```
     SETXCF FORCE,STRUCTURE,STRNAME=DSNDB0A_SCA
     ```

  2. Increase the size of the SCA in the CFRM policy.
  3. Restart Db2 to rebuild the SCA using group restart.

Related concepts

Group restart phases

Group restart requires scanning the logs of each member to rebuild the SCA or retained lock information. It is recommended that you have an alternate coupling facility on which these vital structures can be automatically rebuilt in the event of a coupling facility failure.

Problem: storage shortage in the lock structure

In a data sharing environment, you might encounter a problem that is caused by a storage shortage in the lock structure.

Symptom

A DXR170I message indicates when storage reaches 50, 60, and 70% full. However, because this problem is detected along with other timer driven function, some messages with lower percentages might be missing or skipped if the coupling facility is filling up rapidly. The following message appears at increasing thresholds starting at 80% full:

```
-DB1A DXR142I irlmx THE LOCK STRUCTURE structure-name
      IS zzz%  IN USE
```

System action

Db2 continues processing, but some transactions might obtain a "resource unavailable" code because they are unable to obtain locks.

System programmer action

First, make sure that no members are down and holding retained locks. Restarting any failed members can remove the locks retained in the coupling facility lock structure and release that space.

If a failed member is not the problem, you have two courses of action:

- Lower the lock escalation values to get fewer locks. You do this by taking one of the following actions:
  - Lowering the value of the LOCKS PER TABLE(SPACE) field of installation panel DSNTIPJ (NUMLKTS subsystem parameter)
  - Specifying or changing the value of the LOCAKMAX clause of CREATE or ALTER TABLESPACE

- Increase the size of the lock structure.

Related concepts

Changing the size of the lock structure
You can change the size of a lock structure dynamically or by rebuilding the structure after making a CFRM policy change.

Related reference

LOCKS PER TABLE(SPACE) field (NUMLKTS subsystem parameter) (Db2 Installation and Migration)
CREATE TABLESPACE (Db2 SQL)
ALTER TABLESPACE (Db2 SQL)

Deallocating structures by force

In a few exceptional cases, you might need to deallocate a structure by force to get Db2 restarted.

When you forcibly deallocate an SCA or lock structure, it causes a group restart on the next startup of Db2. Db2 can then reconstruct the SCA or lock structure from the logs during group restart.

To deallocate structures, use z/OS SETXCF FORCE commands to delete persistent structures or connections. Each Db2 structure requires a different set of commands.

- For the group buffer pools:

```
SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
```

- For the SCA:

```
SETXCF FORCE,STRUCTURE,STRNAME=strname
```

- For the lock structure:

```
SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
SETXCF FORCE,STRUCTURE,STRNAME=strname
```

Important: If your site is running z/OS with APAR OA02620 applied, you cannot delete failed-persistent connections to the lock structure unless you also deallocate the lock structure. Deleting failed-persistent connections without also deallocating the associated structure can result in a loss of coupling facility data. This situation can then cause undetectable losses of data integrity. APAR OA02620 protects your site from data corruption problems that can occur as a result of deleting retained locks. In doing so, the APAR also prevents extended outages that would result from long data recovery operations.

APAR OA02620 makes deleting persistent connections and structures easier. When you forcibly deallocate the lock structure, the operating system deletes failed-persistent connections to the structure for you. Instead of issuing the SETXCF FORCE command twice (once for failed-persistent connections to the lock structure and once for the lock structure itself), you need issue it only one time:

```
SETXCF FORCE,STRUCTURE,STRNAME=strname
```

Resolution of transaction manager indoubt units of recovery

A *global transaction* is a unit of work that involves operations on Db2. All of the operations that comprise a global transaction are managed by a transaction manager, such as WebSphere Application Server.

When the transaction manager receives transactionally demarcated requests from an application, it translates the requests into a series of transaction control commands, which it directs to the participating resource managers.

When a global transaction for a local application runs in a data sharing group, all parts of the global transaction run on the same data sharing member.

Example: An application requests updates to multiple databases. The transaction manager translates these update requests into transaction control commands that are passed to several resource managers. Each resource manager then performs its own set of operations on a database. When the application

issues a COMMIT, the transaction manager coordinates the commit of the distributed transaction across all participating resource managers by using the two-phase commit protocol. If any resource manager is unable to complete its portion of the global transaction, the transaction manager directs all participating resource managers to roll back the operations that they performed on behalf of the global transaction.

If a communication failure occurs between the first phase (prepare) and the second phase (commit decision) of a commit, an indoubt transaction is created on the resource manager that experienced the failure. When an indoubt transaction is created, a message like this is displayed on the console of the resource manager:

```
00- 17.24.36 STC00051  DSNL405I  = THREAD
  - G91E1E35.GFA7.00F962CC4611.0001=217
  - PLACED IN INDOUBT STATE BECAUSE OF
  - COMMUNICATION FAILURE WITH COORDINATOR 9.30.30.53.
  - INFORMATION RECORDED IN TRACE RECORD WITH IFCID=209
  - AND IFCID SEQUENCE NUMBER=00000001
```

After a failure, the transaction manager, such as WebSphere Application Server, is responsible for resolving indoubt transactions and for handling any failure recovery. To perform these functions, the server must be restarted and the recovery process initiated by an operator. You can also manually resolve indoubt transactions with the RECOVER INDOUBT command.

Resolving indoubt transactions

After a failure, the transaction manager, such as WebSphere Application Server, is responsible for resolving indoubt transactions and for handling any failure recovery. Alternatively, you can manually resolve indoubt transactions.

Procedure

GUPI To manually resolve indoubt transactions:

1. Display indoubt threads from the resource manager console with the following command:

   ```
   -DISPLAY THD(*) T(I) DETAIL
   ```

   The command produces output like the following example:

   ```
   =dis thd(*) t(i) detail
   DSNV401I  = DISPLAY THREAD REPORT FOLLOWS -
   DSNV406I  = INDOUBT THREADS -
   COORDINATOR               STATUS      RESET URID         AUTHID
   ::FFFF:9.30.30.53..4007   INDOUBT           0000111F049A ADMF002
    V437-WORKSTATION=jaijeetsvl, USERID=admf002,
         APPLICATION NAME=db2jccmain
    V440-XID=7C7146CE 00000014 00000021 F6EF6F8B
             F36873BE A37AC6BC 256F295D 04BE7EE0
             8DFEF680 D1A6EFD5 8C0E6343 67679239
             CC15A350 65BFB8EA 670CEBF4 E85938E0
             06
    V467-HAS LUWID G91E1E35.GFA7.00F962CC4611.0001=217
    V466-THREAD HAS BEEN INDOUBT FOR    00:00:17
   DISPLAY INDOUBT REPORT COMPLETE
   ```

   In this example, only one indoubt thread exists. A transaction is identified by a transaction identifier, called an XID. The first 4 bytes of the XID (in this case, 7C7146CE) identify the transaction manager. Each XID is associated with a logical unit of work ID (LUWID) at the Db2 server. Record the LUWID that is associated with each transaction, for use in the recovery step.
2. Query the transaction manager to determine whether a commit or abort decision was made for each transaction.
3. Based on the decision of the transaction manager, recover each indoubt thread from the resource manager console by either committing or aborting the transaction. Specify the LUWID from the DISPLAY THREAD command, which you recorded in step 1.

   Use one of the following commands:

   ```
   -RECOVER INDOUBT ACTION(COMMIT) LUWID(217)
   ```

```
-RECOVER INDOUBT ACTION(ABORT) LUWID(217)
```

These commands produce output like the following example:

```
=RECOVER INDOUBT ACTION(COMMIT) LUWID(217)
DSNV414I  = THREAD
LUWID=G91E1E35.GFA7.00F962CC4611.0001=217 COMMIT
SCHEDULED
DSN9022I  = DSNVRI '-RECOVER INDOUBT' NORMAL COMPLETION
```

4. Display indoubt threads again from the resource manager console with the following command:

```
 -DISPLAY THD(*) T(I) DETAIL
```

This command produces output like the following example:

```
=dis thd(*) t(i) detail
DSNV401I  = DISPLAY THREAD REPORT FOLLOWS -
DSNV406I  = INDOUBT THREADS -
COORDINATOR                STATUS      RESET URID         AUTHID
::FFFF:9.30.30.53..4007   COMMITTED-H     0000111F049A ADMF002
 V437-WORKSTATION=jaijeetsvl, USERID=admf002,
      APPLICATION NAME=db2jccmain
 V440-XID=7C7146CE 00000014 00000021 F6EF6F8B
          F36873BE A37AC6BC 256F295D 04BE7EE0
          8DFEF680 D1A6EFD5 8C0E6343 67679239
          CC15A350 65BFB8EA 670CEBF4 E85938E0
          06
 V467-HAS LUWID G91E1E35.GFA7.00F962CC4611.0001=217
DISPLAY INDOUBT REPORT COMPLETE
```

Notice that the transaction now appears as a heuristically committed transaction.

5. If the transaction manager does not recover the indoubt transactions in a timely manner, reset the transactions from the resource manager console to purge the indoubt thread information. Specify the IP address and port that you identified from the DISPLAY THREAD command.

```
-RESET INDOUBT IPADDR(::FFFF:9.30.30.53..4007) FORCE
```

This command produces output like the following example:

```
=RESET INDOUBT IPADDR(::FFFF:9.30.30.53..4007) FORCE
DSNL455I  = DB2 HAS NO INFORMATION RELATED TO
IPADDR 9.30.30.53:4007
DSNL454I  = QUALIFYING INDOUBT INFORMATION FOR
IPADDR 9.30.30.53:4007 HAS BEEN PURGED
```

GUPI

Restarting Db2 after termination in a data sharing environment

After a failure or after a normal shutdown of Db2, you can restart Db2 with the -START DB2 command.

You can also choose to have Db2 automatically restart after a failure by using the automatic restart manager of z/OS.

During restart, Db2 resolves inconsistent states. Restart is different in a data sharing environment in the following ways:

- Database exception states exist on both the SCA and the log in a data sharing environment. The database exception states exist solely on the log in non-data-sharing environments.
- Locks that are retained in the event of a failure must be processed.
- If the SCA or the lock structure is lost and cannot be rebuilt on another coupling facility, all members of the group come down. If this unlikely event occurs, then Db2 must perform *group restart*. Group restart is distinguished from normal restart by the activity of rebuilding the information that was lost from the SCA or lock structure. Group restart does not necessarily mean that all members of the group start up again, but information from all non-starting members must be used to rebuild the lock structure or SCA.

- You can set the DEL_CFSTRUCTS_ON_RESTART subsystem parameter to YES to automatically attempt to delete and rebuild the coupling facility structures whenever Db2 is restarted. For systems that are used for disaster recovery, set this parameter to YES to ensure that the coupling facility structures are up to date when the restart is complete. For production systems, set this parameter to NO.

Related concepts

Automatic restart of z/OS (Db2 Installation and Migration)

Related reference

-START DB2 (Db2) (Db2 Commands)

DEL CF STRUCTS field (DEL_CFSTRUCTS_ON_RESTART subsystem parameter) (Db2 Installation and Migration)

Active and retained locks for Db2data sharing members

Active locks, retained locks, and retained utility locks can be held by a failed member. Locks that are held by a failed member can affect the availability of data to the other members of the group that are still running Db2 applications.

Active locks

When a member is active, the locks that it holds are called *active locks*. For transaction locks (L-locks), the normal concurrency mechanisms apply, including suspensions and timeouts when incompatible locks are requested for a resource. For physical locks (P-locks), Db2 uses a negotiation process to control access.

Retained locks

The particular concern for availability is what happens to locks when a member fails. For data sharing, active locks used to control updates to data (modify locks) become *retained* in the event of a failure. This means that information about those locks is stored in the coupling facility until the locks are released during restart. Retained locks are necessary to protect data in the process of being updated from being accessed by another active member of the group.

Db2 has various types of retained locks, among them are L-locks, page set P-locks, and page P-locks. As long as a retained lock is held by a failed member, another member cannot lock the resource in a mode that is incompatible with the mode of the retained lock on that resource. Incompatible requests from other members are suspended if you specify a non-zero value for the RETAINED LOCK TIMEOUT parameter of installation panel DSNTIPI. Incompatible requests from other members are immediately rejected if you specify a value of 0 for this parameter.

In the case of a page set P-lock, it is conceivable that an entire page set could be unavailable. For example, an X mode page set P-lock is retained if the page set is non-GBP-dependent at the time of the failure. Incompatible lock requests from other members can be processed after the retained locks are removed, which occurs during restart. To keep data available for all members of the group, **restart failed members as soon as possible**, either on the same or another z/OS system.

You can restart a failed member in "light" mode to quickly recover the locks held by that member. Restarting in light mode is primarily intended for a cross-system restart in the event of a failed z/OS system.

Retained utility locks

When a member is running a utility, it holds a lock on the utility ID (UID) for that utility. That lock is also retained if the member fails. This means that you cannot restart a utility until the failed member is restarted and the retained lock is converted to an active lock.

When retained locks are reacquired or purged

During the restart process, Db2 removes its retained locks in one of two ways:

- Db2 converts the lock to active. This is called *reacquiring* the lock, and it is what Db2 does for page set P-locks.
- Db2 purges the lock. This is what Db2 does for page P-locks and for L-locks.

The following table shows how the restart processing for locks can happen at different times.

Table 23. Restart processing for locks

Lock type	Processing
Page set P-lock	Reacquired when page sets are opened for log apply. This generally happens during forward recovery or before restart completes. The earliest that a page set P-lock can become negotiable is at the end of forward recovery. If no log apply is needed, it can happen later, such as the end of restart processing.
Page P-lock	Purged at the end of forward recovery.
L-lock	Purged at the end of restart processing.

If a member has a non-zero value for RETAINED LOCK TIMEOUT and it requests a lock that is incompatible with a retained lock, its applications can be suspended as long as the retained locks are held. Its request can go through as soon as the retained lock is purged or becomes negotiable. For example, if an application is suspended because of an incompatible retained page set P-lock, that retained lock most likely becomes active and available for negotiation at the end of forward log recovery.

Related concepts

Concurrency and locks in data sharing environments
Concurrency and transaction locks have additional implications in data sharing environments.

Protection of retained locks: failed-persistent connections
Use extreme care when deleting failed-persistent connections to the lock structure. IRLM and XES use failed-persistent connections to the lock structure to track retained lock information. Retained locks might be lost and data integrity might be exposed by arbitrarily deleting a failed-persistent lock structure connection.

Restart light
The LIGHT(YES), LIGHT(NOINDOUBTS), and LIGHT(CASTOUT) clauses of the -START DB2 command restart a member in "light" mode.

Page set P-Locks
P-locks are used on physical objects stored in buffers (table spaces, index spaces, and partitions).

Restart light

The LIGHT(YES), LIGHT(NOINDOUBTS), and LIGHT(CASTOUT) clauses of the -START DB2 command restart a member in "light" mode.

Restart light mode does the following:

- Minimizes the overall storage required to restart the member.
- Removes retained locks as soon as possible, except for the following:
  - Locks that are held by postponed-abort units of recovery, if the LBACKOUT subsystem parameter is set to LIGHT or LIGHTAUTO.
  - IX and SIX mode page set P-locks, if LIGHT(YES) or LIGHT(NOINDOUBTS) is specified. These locks do not block access by other members; however, they do block drainers, such as utilities.
- Terminates the member normally after forward and backward recovery is complete. No new work is accepted.

Restart light mode is not recommended for a restart in place. It is intended only for a cross-system restart of a system that does not have adequate capacity to sustain the Db2 and IRLM in the event of a failed z/OS system.

Indoubt units of recovery

When you restart a member by using the LIGHT(NOINDOUBTS) clause of the -START DB2 command, the member terminates without waiting for resolution of indoubt units of recovery (URs).

When you restart a member by using the LIGHT(YES) or LIGHT(CASTOUT) clause and indoubt URs exist, Db2 issues message DSNR052I at the end of restart. The member continues to run in light mode to allow the indoubt URs to be recovered.

When Db2 for z/OS operates in restart light mode, it resolves indoubt URs in environments in which Db2 for z/OS is a participant relative to a local or a remote coordinator that makes the decision to commit or abort an indoubt UR. Db2 for z/OS can also be a coordinator relative to local or remote participants. As a coordinator, Db2 provides commit and abort decisions to remote participants as long as Db2 and DDF remain started while resolving participant-related indoubt URs. In either of those environments, Db2 automatically terminates after all indoubt URs have been resolved.

XA recovery processing for a remote XA Transaction Manager (XA TM) requires that the XA TM connect to the SQL port (tcpport) of the data sharing group to retrieve a list of transaction identifiers (XIDs) that are indoubt. Because the SQL port is unavailable for the restart light member, another member of the data sharing group must be available and accessible through a group DVIPA. After the available member is contacted, that member returns the member DVIPA and resynchronization port (resport) for any indoubt XIDs that are owned by the restart light member. The XA TM resolves work at the restart light member because its resynchronization port is available.

Indoubt URs can be resolved in two ways: automatically through resynchronization processing or manually.

- Automatically: Ensure that the appropriate commit coordinators (IMS or CICS subsystems that have indoubt URs on the restart-light member) are started so that they can resynchronize with the member to resolve the indoubt URs.
- Manually: Use the RECOVER INDOUBT command to manually resolve the indoubt URs.

Use the DISPLAY THREAD command to monitor the progress of indoubt UR resolution.

After the last indoubt UR is resolved, the member terminates. Alternatively, if you want to stop the member before all indoubt URs are resolved, you can issue the -STOP DB2 command. Db2 issues message DSNR046I to indicate that incomplete URs still exist.

Db2 commands and restart-light mode

The following commands are not allowed when a member is started in light mode:

- DISPLAY DATABASE, START DATABASE, STOP DATABASE
- DISPLAY RLIMIT, START RLIMIT, STOP RLIMIT
- SET SYSPARM
- START DDF, STOP DDF

Related tasks

Creating an automatic restart policy (Db2 Installation and Migration)

Related reference

-START DB2 (Db2) (Db2 Commands)

Group restart phases

Group restart requires scanning the logs of each member to rebuild the SCA or retained lock information. It is recommended that you have an alternate coupling facility on which these vital structures can be automatically rebuilt in the event of a coupling facility failure.

The automatic rebuild that occurs during a coupling facility failure does not require the log scans that group restart does.

During group restart, all restarting members update the SCA or lock structure from information contained in their logs. If you do not issue a START DB2 command for all members of the group, the started members perform group restart on behalf of the non-starting members by reading their logs.

Although one member can perform restart on behalf of the group, you should restart all of the non-quiesced members together, perhaps by using an automated procedure. This shortens the total restart time. Also, because retained locks are held for non-starting members, it is best to start all members of the group for maximum data availability.

Because all members must synchronize at the end of current status rebuild (CSR) and at the end of forward log recovery, the time taken for group restart done in parallel is determined by the member that has the longest CSR and, if the lock structure is lost, by the member that has the longest forward log recovery.

When the members are synchronized after forward log recovery, backward log recovery proceeds in parallel for the started members.

The phases of group restart are generally the same as in a non-data-sharing environment, with the addition of function for group restart. The phases of group restart vary based on whether the SCA, lock structure, or both are lost, and whether information is needed from the logs of inactive members. The following table summarizes the phases, depending on which structure is lost.

Table 24. Summary of group restart phases based on which structure is lost

SCA lost	Lock structure lost
Initialization	Initialization
CSR (rebuild SCA)	CSR (reacquire page set P-locks)
Peer CSR (rebuild SCA)	Peer CSR (rebuild page set P-locks)
Forward-log recovery (rebuild locks)	Forward-log recovery (rebuild locks) or Peer forward recovery (rebuild locks)
Backward-log recovery	Backward-log recovery

Db2 initialization

The initialization phase verifies BSDSs, logs, and the integrated catalog facility catalog.

In this phase, the RBA of the last log record is retrieved, and logging is set to begin at the next control interval (CI) following the last RBA. Also during this phase, Db2 determines if the lock structure or SCA is lost and needs to be recovered.

The following message output shows a group restart controlled by member DB3A on behalf of members DB1A and DB2A. During initialization, you see messages similar to the following messages:

```
 $HASP373 DB3AMSTR STARTED
⋮
 DSNJ127I @DB3ADB2 SYSTEM TIMESTAMP FOR BSDS= 95.040 13:03:05.32
 DSNJ001I @DB3ADB2 DSNJW007 CURRENT COPY 1 ACTIVE LOG 753
 DATA SET IS DSNAME=DSNC410.THIRD.LOGCOPY1.DS01,
 STARTRBA=000000000000,ENDRBA=000000167FFF
 DSNJ001I @DB3ADB2 DSNJW007 CURRENT COPY 2 ACTIVE LOG
 DATA SET IS DSNAME=DSNC410.THIRD.LOGCOPY2.DS01,
 STARTRBA=000000000000,ENDRBA=000000167FFF
 DSNJ099I @DB3ADB2 LOG RECORDING TO COMMENCE WITH
 STARTRBA=000000010000
```

```
⋮
$HASP373 DB3ADBM1 STARTED
```

Current status rebuild

When a restarting member has completed its own current status rebuild (CSR), it checks and waits for every other member to finish CSR. If non-starting members exist, peer CSR is performed.

During current status rebuild, the following tasks are accomplished:

- The SCA is rebuilt from the log by reading it forward from the last checkpoint.

  All restarting members add entries to the indoubt transaction ID (XID) list in the SCA from information contained in their logs. If an indoubt XID entry cannot be added to the SCA, the member abnormally terminates with reason code 00F70606.
- Db2 determines all outstanding units of recovery (URs) that were interrupted by the previous termination.
- If the lock structure is lost, all partition and page set P-locks are reacquired by reading information from the log. These locks are retained locks until the end of restart.

During CSR, you see messages similar to the following messages. (The phrase in parentheses is not part of the output.)

```
DSNR001I @DB3ADB2 RESTART INITIATED
DSNR003I @DB3ADB2 RESTART...PRIOR CHECKPOINT RBA=00000000DC4E
DSNR004I @DB3ADB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of current status rebuild for member DB3A)

DSNR021I @DB3ADB2 DSNRRGRC DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
```

Peer CSR

Peer CSR is skipped unless it is necessary to perform group restart on behalf of non-starting members. Peer CSR is not performed on non-starting members that are normally quiesced.

A restarting member can select an inactive member on which to perform peer initialization and peer CSR:

- If the SCA is lost, the restarting member rebuilds SCA information from the information contained in the non-starting member's logs.
- If the lock structure is lost, the restarting member reacquires page set and partition P-locks (as retained locks) for the non-starting member. Those locks are now retained locks.

During peer CSR, you see messages similar to the following messages. (The phrases in parentheses are not part of the output.)

```
DSNR023I @DB3ADB2 DSNRRGRC GROUP RESTART INITIATED TO
RECOVER THE SCA FOR GROUP MEMBER DB1A
DSNR003I @DB3ADB2 RESTART...PRIOR CHECKPOINT RBA=00000201CC4E
DSNR004I @DB3ADB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
DSNR024I @DB3ADB2 DSNRRGRC GROUP RESTART COMPLETED TO
RECOVER THE SCA FOR GROUP MEMBER DB1A
(End of peer current status rebuild for member DB1A)

DSNR023I @DB3ADB2 DSNRRGRC GROUP RESTART INITIATED TO
RECOVER THE SCA FOR GROUP MEMBER DB2A
DSNR003I @DB3ADB2 RESTART...PRIOR CHECKPOINT RBA=000000009C4E
DSNR004I @DB3ADB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
DSNR024I @DB3ADB2 DSNRRGRC GROUP RESTART COMPLETED TO
RECOVER THE SCA FOR GROUP MEMBER DB2A
(End of peer current status rebuild for DB2A)

DSNR022I @DB3ADB2 DSNRRGRC DB2 SUBSYSTEM HAS
COMPLETED GROUP RESTART FOR PEER MEMBERS
(End of peer processing)
```

When all members have completed CSR—either by performing it on their own or by having a peer perform it for them—the SCA has been rebuilt, and page set and partition P-locks have been reacquired.

Forward-log recovery
In the forward-log recovery phase, Db2 applies log records and completes any database write operations that were outstanding at the time of the failure.

It also rebuilds retained locks during this phase by reading that information from the log. Restart time is longer when lock information needs to be recovered during a group restart, because Db2 needs to go back to the earliest begin_UR for an inflight UR belonging to that subsystem. This is necessary to rebuild the locks that member has obtained during the inflight UR. (A normal restart goes back only as far as the earliest RBA that is needed for database writes or is associated with the begin_UR of indoubt units of recovery.)

If a problem exists that prevents an object's log record from being applied (for example, if the disk version of the data could not be allocated or opened), or if the page set is deferred, Db2 adds the relevant pages and page ranges to the logical page list. Only pages affected by the error are unavailable.

When each restarting member has completed its own forward-log recovery, it checks and waits for all members to finish. If there are non-starting members, peer forward-log recovery is performed.

During forward-log recovery, you see messages similar to the following messages:

```
DSNR005I @DB3ADB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR021I @DB3ADB2 DSNRRGRH DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
```

Peer forward-log recovery
Peer forward-log recovery is skipped unless it is necessary to rebuild lock information from information contained in inactive, non-quiesced members' logs.

Peer retained-lock recovery requires that Db2 do a peer initialization, a partial CSR phase to rebuild UR status, and then do the forward-log recovery for the non-started member.

During peer forward-log recovery, you see messages similar to the following messages. (The phrases in parentheses are not part of the output.)

```
DSNR025I @DB3ADB2 DSNRRGRH GROUP RESTART INITIATED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB1A
DSNR003I @DB3ADB2 RESTART...PRIOR CHECKPOINT RBA=00000201CC4E
DSNR004I @DB3ADB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of peer partial current status rebuild for DB1A)

DSNR005I @DB3ADB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR026I @DB3ADB2 DSNRRGRH GROUP RESTART COMPLETED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB1A
(End of peer forward-log recovery for member DB2A)

DSNR025I @DB3ADB2 DSNRRGRH GROUP RESTART INITIATED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB2A
DSNR003I @DB3ADB2 RESTART...PRIOR CHECKPOINT RBA=000000009C4E
DSNR004I @DB3ADB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of partial current status rebuild for member DB2A)

DSNR005I @DB3ADB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR026I @DB3ADB2 DSNRRGRH GROUP RESTART COMPLETED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB2A
DSNR022I @DB3ADB2 DSNRRGRH DB2 SUBSYSTEM HAS
COMPLETED GROUP RESTART FOR PEER MEMBERS
(End of peer forward-log recovery for member DB1A)
```

When all members have completed forward-log recovery—either by performing it on their own or by having a peer perform it for them—the lock structure has been rebuilt.

Backward-log recovery
During the backward-log recovery phase, Db2 completes recovery processing by reversing all changes performed for inflight and in-abort units of work.

At this point, all forward-log applies have been performed, and inflight, indoubt, and in-abort transactions are protected by locks. Any updates that cannot be externalized to the group buffer pool or to disk cause the affected pages to be added to the logical page list.

During backward-log recovery, you see messages similar to the following messages. (The phrase in parentheses is not part of the output.)

```
DSNR006I @DB3ADB2 RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0
(End of backward-log recovery for member DB3A)
DSNR002I @DB3ADB2 RESTART COMPLETED
DSN9022I @DB3ADB2 DSNYASCP 'START DB2' NORMAL COMPLETION
```

The backward-log recovery messages for other members do not appear until those members are actually started. Backward-log recovery can occur in parallel for all the started members. No peer backward-log recovery exists; all members must be started to complete backward-log recovery and to release the locks held by inflight and in-abort transactions.

Protection of retained locks: failed-persistent connections

Use extreme care when deleting failed-persistent connections to the lock structure. IRLM and XES use failed-persistent connections to the lock structure to track retained lock information. Retained locks might be lost and data integrity might be exposed by arbitrarily deleting a failed-persistent lock structure connection.

You should delete failed-persistent lock structure connections only in the following situations:

- Disaster recovery. All Db2 and IRLM-related failed-persistent connections and structures must be deleted before restarting the data sharing group at the remote site. During the restart, Db2 uses the group restart process to rebuild the retained locks from the logs.
- A Parallel Sysplex-wide outage when the lock structure is forced. Failed-persistent connections can be safely forced when all members are down and the lock structure is also forced. During the restart phase of a disaster recovery process, Db2 uses the group restart process to rebuild the retained locks from the logs.
- After a hard failure occurs, such as a check-stop or abnormal re-IPL of a z/OS image that contains an active member and the Db2 or IRLM member has not been restarted.

Important: This information about deleting failed-persistent connections is not relevant for sites running z/OS with APAR OA02620 applied. With this APAR, you cannot delete failed-persistent connections to the lock structure unless you also deallocate the lock structure. Deleting failed-persistent connections without also deallocating the associated structure can result in a loss of coupling facility data. This situation can then cause undetectable losses of data integrity. APAR OA02620 protects your site from data corruption problems that can occur as a result of deleting retained locks. In doing so, the APAR also prevents extended outages that would result from long data recovery operations.

Do not delete a member's failed-persistent connection just because that member was normally quiesced.

When a member is shut down while holding retained locks, those retained locks are transferred to another member to hold until the original member is restarted. Therefore, although a normally quiesced member does not hold retained locks for itself, it might hold retained locks for another member that was shut down. The following situations can cause a transfer of retained locks:

- Member failure. Assume that three members, DB1A, DB2A, and DB3A, are running normally. If z/OS incurs a hard failure that takes down DB2A, DB2A's retained locks are transferred to one of the other

members; assume that it was DB1A. If DB1A is subsequently shut down normally, you might assume that no locks are held by DB1A and that you can safely delete DB1A's failed-persistent connection. In actuality, because DB1A is holding the retained locks from DB2A, deleting DB1A's failed-persistent connection deletes DB2A's retained locks and exposes the Db2 data to potential data integrity errors.

- A lock structure rebuild when one or more members are down and holding retained locks. This could be caused by either the z/OS SETXCF command or a coupling facility-related failure.

A coupling facility structure rebuild deletes any failed-persistent connections that existed before the rebuild. The retained locks belonging to failed members are re-created and held during the rebuild by one of the active members until the failed members are restarted.

Handling coupling facility connections that hang

If a connection to a coupling facility structure hangs, an operator should complete some actions to recover from the hanging connection.

About this task

When a member abnormally terminates, its connections to coupling facility structures are put into a FAILING state by cross-system extended services for z/OS (XES). The member remains in this FAILING state until all surviving members of the group have responded to the XES Disconnected/Failed Connection (DiscFailConn) event for each structure. XES sends this event to each surviving member of the group so that the surviving members can take the necessary recovery actions in response to the failed member.

After all surviving members of the group perform the necessary recovery actions and provide DiscFailConn responses to XES for a given coupling facility structure, XES changes the failed member's connection status for that coupling facility structure from FAILING to FAILED PERSISTENT. The member can reconnect to the coupling facility structure during restart when the member's status is FAILED PERSISTENT.

When you restart the member immediately following a connection failure, the member can attempt to reconnect to a coupling facility structure while its connection is still in a FAILING state. If this occurs, XES denies the reconnect request with a 0C27 reason code. Db2 responds to this by entering a connection-retry loop until the connection succeeds or until it reaches the maximum retry count.

For the SCA, the maximum retry count is 200 times with a 3-second interval between each attempt. For the group buffer pools, the maximum retry count is 5 times with a 10-second interval between each attempt. You might notice a message similar to the following message, which indicates a failed connection attempt:

```
IXL013I IXLCONN REQUEST FOR STRUCTURE DB2GR0W_SCA FAILED.
JOBNAME: DB2VMSTR ASID: 05E1 CONNECTION NAME: DB2_DB2V
IXLCONN RETURN CODE: 0000000C,   REASON CODE: 02010C27
```

The preceding message might be displayed multiple times while Db2 is in a connection-retry loop. This is normal.

In rare cases, one or more of the surviving members of a group encounters difficulties in providing the DiscFailConn response to XES for a given coupling facility structure. When this happens, XES issues a message similar to the following message for each member from which it does not receive a response within two minutes:

```
IXL041I CONNECTOR NAME: DB2_DB2M, JOBNAME: DB2MMSTR, ASID: 0086
HAS NOT RESPONDED TO THE DISCONNECTED/FAILED CONNECTION EVENT FOR
SUBJECT CONNECTION: DB2_DB2V.
DISCONNECT/FAILURE PROCESSING FOR STRUCTUR DB2GR0W_SCA
CANNOT CONTINUE.
MONITORING FOR RESPONSE STARTED: 08/08/2002 23:50:23.
DIAG: 0000 0000 00000000
```

In extreme cases, the maximum number of connection retries might be reached. If encountered for the SCA, this situation prevents the failed member from restarting and Db2 issues a message similar to the following message:

```
DSN7506A  -DB2V DSN7LSTK
CONNECTION TO THE SCA STRUCTURE DB2GR0W_SCA FAILED.
 MVS IXLCONN RETURN CODE = 0000000C,
 MVS IXLCONN REASON CODE = 02010C27.
```

Procedure

To recover from coupling facility structure connections that hang:

1. Save a dump of all Db2 and IRLM members along with SDATA= (COUPLE, XESDATA) so that IBM Support can determine what is causing the hung connections. See message II10850 for more information.
2. Attempt a REBUILD of the lock structure.

   This can sometimes clear the condition that is causing the DiscFailConn response to hang. If the REBUILD of the lock structure works, XES issues a message similar to the following message for each group member as it provides the required DiscFailConn response:

   ```
   IXL043I CONNECTION NAME: DB2_DB2M, JOBNAME: DB2MMSTR, ASID: 0086
                  HAS PROVIDED THE REQUIRED RESPONSE. THE REQUIRED RESPONSE
                  FOR THE DISCONNECTED/FAILED CONNECTION EVENT
                  FOR SUBJECT CONNECTION DB2_DB2V,
                  STRUCTURE DB2GR0W_SCA IS NO LONGER EXPECTED.
   ```

   If the REBUILD does not work, proceed to step 3.
3. Issue the `D XCF,STR,STRNM=<strname>,CONNM=<conname>` command for the structure or connector that is in the FAILING state.

   Alternatively, issue the `D XCF,STR,STRNM=<strname>,CONNM=ALL` command. Both commands display the status of the structures and connectors that are used by XES.

   If this command identifies the unresponsive members, skip to Step 6. If it does not identify the unresponsive members, proceed to Step 4.
4. Attempt a structure REBUILD for the affected structure, if you have not already done this.
5. If the REBUILD hangs, issue the `D XCF,STR,STRNM=<strname>` command to identify the unresponsive connector.

   This identifies the members that are unresponsive to the REBUILD. These members are probably the same members that are unresponsive to the DiscFailConn event.
6. Cancel and recycle the unresponsive members.

   The `STOP D` command might not work because internal Db2 processes are hung, so cancel IRLM or Db2 MSTR.

   As each member terminates, verify that XES issues message IXL043I to indicate that it no longer expects a DiscFailConn response from that member. When all members that owe responses have been stopped, all connections to the SCA should be ACTIVE or FAILED PERSISTENT.
7. Issue the `D XCF,STR,STRNM=<sca>,CONNM=ALL` command to verify the status of the connections to SCA.
8. Restart all members with FAILED PERSISTENT connections.

   As each member successfully reconnects to the SCA, XES issues message IXL014I. If a problem still exists, proceed to step 9.
9. Stop and restart the systems on which the unresponsive members are running. If restarting the system does not fix the unresponsive members, proceed to step 10.
10. Cancel and recycle all connectors to the coupling facility structure. If a problem still exists, proceed to step 11.
11. Stop and restart all systems.

Related information

z/OS MVS Recovery and Reconfiguration Guide

Postponed backout in a data sharing environment

Two options on the DSNTIPL1 installation panel allow you to enable and control postponed backout.

You can postpone backout processing for in-abort and inflight units of recovery (URs). The advantage of postponing backout processing is that Db2 can be up and accepting new work before handling backout processing.

The following fields on installation panel DSNTIPL1 enable and control postponed backout:

- LIMIT BACKOUT

  The AUTO, YES, LIGHT, and LIGHTAUTO values of the LIMIT BACKOUT field enable postponed backout. AUTO and LIGHTAUTO are the recommended values because you do not have to remember to issue commands to start backout processing. The AUTO and LIGHTAUTO values result in the same behavior for normal Db2 restart. However, the LIMIT BACKOUT field is ignored when AUTO is specified during Db2 restart with the LIGHT(YES) or LIGHT(NOINDOUBTS) option.
- BACKOUT DURATION

  A multiplier that is used to calculate how much log processing to back out during restart. This option is valid only when AUTO, YES, LIGHT, or LIGHTAUTO is specified for the LIMIT BACKOUT field.

Related tasks

Resolving postponed units of recovery (Db2 Administration Guide)

Related reference

LIMIT BACKOUT field (LBACKOUT subsystem parameter) (Db2 Installation and Migration)

BACKOUT DURATION field (BACKODUR subsystem parameter) (Db2 Installation and Migration)

Why postponed backout works in a data sharing environment

While a member is restarting, it cannot accept new work. If a restarting member has much backout work to perform, and the work is not postponed, the restart can be time-consuming.

In a data sharing environment with spare capacity, you can reroute work to another member of the group. However, if the other system does not have enough capacity, or if you are not able to switch workloads because your configuration is such that there is a strong one-to-one relationship between a member and the workload that runs on that member, postponing backout processing for long-running URs can shorten the outage significantly.

What data is unavailable?

One difference between the non-data-sharing and the data sharing implementation of postponed backout is the degree of data unavailability.

In non-data-sharing, Db2 places any page set or partition that has pending backout work into a restrictive status called *restart pending*, which blocks access to data at the page set or partition level.

In data sharing, no restrictive status is set. Access to data with pending backout work is blocked by transaction locks that persist through restart. The following retained locks persist through restart when postponed backout processing is active:

- Retained transaction locks held on page sets or partitions for which backout work has not been completed
- Retained transaction locks held on tables, pages, rows, or LOBs of those table spaces or partitions

The retained transaction locks on any particular page set or partition are freed when all URs using that page set or partition have completed their backout processing. Until that happens, the page set or partition is placed in an advisory status called *advisory restart-pending* (AREST).

Identifying objects in advisory restart-pending status

Objects with backout work pending display a status of AREST (advisory restart-pending).

Procedure

To identify objects in advisory restart-pending status:

1. Use the DISPLAY GROUP command to determine if a member has work pending,

   The following statuses indicate that work is pending for that member:

   A I
   This active member has indoubt URs, URs for which backout work is postponed, or both.

   Q I
   This quiesced member has indoubt URs, URs for which backout work is postponed, or both.
2. Use the DISPLAY THREAD TYPE(INDOUBT) and DISPLAY THREAD TYPE(POSTPONED) commands to determine whether the pending work is indoubt URs or postponed backout URs.
   - To recover indoubt URs, use the RECOVER INDOUBT command.
   - To recover postponed-abort URs, use the RECOVER POSTPONED command. (If you specify AUTO or LIGHTAUTO for the LBACKOUT subsystem parameter, Db2 automatically recovers the postponed URs after a normal restart. Starting Db2 with the LIGHT(YES) or LIGHT(NOINDOUBTS) option causes postponed-abort URs to be automatically recovered unless LIGHTAUTO or LIGHT is specified for the LBACKOUT subsystem parameter.)
3. Use the DISPLAY DATABASE command to determine which objects have backout work pending (status of AREST).

   The AREST status is removed when the backout processing is complete for the object. You cannot use the command START DATABASE with ACCESS(FORCE) to remove the advisory status if any indoubt or postponed backout URs exist for the object.

   Utilities are not restricted by the AREST status, but any write claims that are held by postponed-abort URs on the objects in AREST status prevent draining utilities from accessing that page set.

Related reference

-DISPLAY GROUP (Db2) (Db2 Commands)
-DISPLAY THREAD (Db2) (Db2 Commands)
LIMIT BACKOUT field (LBACKOUT subsystem parameter) (Db2 Installation and Migration)
-DISPLAY DATABASE (Db2) (Db2 Commands)
-START DB2 (Db2) (Db2 Commands)

Restarting a member with conditions

Some installations use conditional restart to bypass a long active UR backout, such as might occur when a long-running batch job fails without having issued interim commits. In data sharing environments, this use of conditional restart is not recommended.

It is safer and provides better availability to reroute work to another member or to postpone backout processing rather than suffer the total outage necessary for a conditional restart.

If you do perform conditional restart, you need to stop all members of the group except the one that is conditionally restarting to ensure that applications on those other members do not change the data that is not locked by the restarting member.

You might, in unusual circumstances, choose to make inconsistent data available for use without recovering it. This might be the case for certain test groups, for example, where data consistency is not important.

Related tasks

Using locks for data consistency (Managing Security)

Performing a cold start

In some cases, for example when advised to do so by IBM Support, you might need to perform a cold start in a data sharing environment. A cold start occurs when STARTRBA and ENDRBA are equal on the CRESTART statement of the DSNJU003 (change log inventory) utility.

Procedure

To cold start a Db2 subsystem in a data sharing environment:

1. Stop all other members of the data sharing group.
2. Cold start the chosen member using ACCESS(MAINT). The cold start deallocates the group buffer pools to which this member was connected.
3. Resolve all data inconsistency problems resulting from the cold start.
4. Start all the other members and restart this one without ACCESS(MAINT).

Related tasks

Resolving inconsistencies (Db2 Administration Guide)

Related information

Recovering from a failure resulting from total or excessive loss of log data (Db2 Administration Guide)

Conditionally restarting without a cold start

Use this procedure for a conditional restart when you are truncating the log but you are not performing a cold start.

Procedure

To conditionally restart a Db2 subsystem without a cold start:

1. Stop all other members of the data sharing group.
2. Conditionally restart the chosen member using ACCESS(MAINT).
3. Resolve all data inconsistency problems resulting from the conditional start.
4. After you have resolved the data inconsistencies resulting from the conditional restart, start all the other members and restart the chosen member without ACCESS(MAINT).

Related tasks

Resolving inconsistencies (Db2 Administration Guide)

Deferring recovery during restart

It is possible to defer recovery for an object during restart. If you use the DEFER installation option, that defers the log apply processing of the object for only the member who specified that option.

All the pages that would have been applied to disk or the group buffer pool are instead added to the logical page list. This can affect the rest of the group; any member who needs a page that is on the logical page list will not be able to access that page until the object is restarted.

To make those pages available after Db2 restarts, use the START DATABASE command with the SPACENAM option.

Deferring recovery does not change restart time significantly.

Starting duplexing for a structure

When duplexing starts, a period exists in which activity to the structure is quiesced. For this reason, you should start duplexing during a period of low activity in the system.

About this task

While duplexing is being established, and for the entire time duplexing is in effect, a display of the structure shows the structure as being in DUPLEXING REBUILD. The rebuild phase is called DUPLEX ESTABLISHED, which means that duplexing is truly active for the structure.

Procedure

To start duplexing for a structure:

- Complete one of the following actions:
  - Activate a new CFRM policy with DUPLEX(ENABLED) for the structure. The structure must have one or more actively-connected Db2 instances that support duplexing, and no connectors that do not support duplexing. If the structure is currently allocated, z/OS can automatically initiate the process to establish duplexing as soon as you activate the policy. If the structure is not currently allocated, the duplexing process can be initiated when the structure is allocated.
  - Activate a new CFRM policy with DUPLEX(ALLOWED) for the structure. If the structure is currently allocated, use the following command to start the duplexing rebuild:

    ```
    SETXCF START,REBUILD,DUPLEX,STRNAME=strname
    ```

    If the structure is not currently allocated, wait until it is allocated before starting the duplexing rebuild.

Related concepts

Group buffer pool monitoring with the z/OS DISPLAY XCF,STR command
You can monitor group buffer pools with the DISPLAY XCF,STR command.

Stopping duplexing for a structure

If you need to stop duplexing, you must first decide which instance of the structure is to remain as the surviving simplexed structure.

About this task

If you have a choice, use the primary structure as the surviving one. For example, the primary group buffer pool has intact page registration information. If you choose the secondary group buffer pool—which does not have intact page registration information—as the surviving structure, all this information is lost. As a result, all locally cached pages revert to an invalid state and these pages must then be refreshed from the group buffer pool or disk.

Procedure

To **temporarily** switch to using one of the duplexed structures as a simplexed structure:

1. Optional: If DUPLEX(ENABLED) is specified for the active CFRM policy, activate a new policy specifying DUPLEX(ALLOWED). For the new DUPLEX value to take effect, all other CFRM policy parameters must remain unchanged.
2. Use the SETXCF STOP,REBUILD command, specifying KEEP=OLD to use the primary structure as a simplexed structure, or specifying KEEP=NEW to use the secondary structure as a simplexed structure.

   For example, the following command reverts to using the primary structure as the simplexed structure:

   ```
   SETXCF STOP,RB,DUPLEX,STRNAME=strname,KEEP=OLD
   ```

What to do next

If you do not plan to reestablish duplexing for the structure in the near future, activate a new CFRM policy specifying DUPLEX(DISABLED) for the structure. Doing so, **permanently** switches to using one of the duplexed structures as a simplexed structure.

If you want to perform maintenance on a coupling facility that contains primary or secondary structure instances, use the SETXCF STOP,RB,DUPLEX command and specify the target coupling facility.

Related tasks

Shutting down the coupling facility
Create a plan for those cases in which it is necessary to shut down a coupling facility to apply maintenance or perform another type of reconfiguration.

Related reference

z/OS SETXCF STOP Command (MVS System Commands)

Shutting down the coupling facility

Create a plan for those cases in which it is necessary to shut down a coupling facility to apply maintenance or perform another type of reconfiguration.

About this task

For the least disruptive shutdown, move all of your structures to another coupling facility before shutting it down. See the guidelines below for handling this event. For other structures in the coupling facility, see the appropriate product documentation.

Procedure

To shut down a coupling facility, consider the following guidelines:

1. Prepare for the move:
   - Ensure that you have enough room on the alternate coupling facility for all structures you intend to move.
   - Ensure that the preference list for the group buffer pool, SCA, and lock structures contains the alternate coupling facility information.
2. Move simplexed structures to the **new** coupling facility by using the following command:

   ```
   SETXCF START,REBUILD,CFNAME=newcf,LOC=OTHER
   ```

   This command rebuilds all structures that allow rebuild in the alternate coupling facility.
3. Deallocate duplexed structures on the original (**target**) coupling facility by using the following command:

   ```
   SETXCF STOP,REBUILD,DUPLEX,CFNAME=targetcf
   ```

   If the CFRM policy specifies DUPLEX(ALLOWED) or DUPLEX(DISABLED), the structure goes into simplex mode.

   If the CFRM policy specifies DUPLEX(ENABLED), z/OS might try to automatically restart duplexing. If you have a third coupling facility specified in the CFRM policy, it is possible to continue duplexing during the outage of the target coupling facility. When an operator command causes the structure to drop from duplex to simplex mode, z/OS avoids automatically reduplexing the structure back into the same coupling facility from which one of the duplexed instances of the structure was just deallocated. Instead, it duplexes the structure in the third coupling facility.
4. Return the coupling facility to service by performing the following steps:
   a) Evacuate the target coupling facility.

   ```
   SETXCF START,REBUILD,CFNAME=cfname
   ```

   b) Perform the maintenance operation on the target coupling facility and bring it back into the Parallel Sysplex.
   c) Repopulate the target coupling facility with the structures that were in it before the coupling facility was brought down.

   ```
   SETXCF START,REBUILD,POPULATECF,CFNAME=cfname
   ```

Related reference

z/OS SETXCF STOP Command (MVS System Commands)

z/OS SETXCF START command (MVS System Commands)

Chapter 6. Performance monitoring and tuning for data sharing environments

Actions such as tuning your use of locks and managing your group buffer pools can improve the performance of your data sharing group.

One of the main objectives of the data sharing function of Db2 for z/OS is to increase processing capacity while using the lower cost IBM Z Parallel Sysplex technology. Workload capacity is increased by allowing many Db2 subsystems to access shared Db2 data with full integrity. Db2 data sharing is designed to address this objective while providing balanced performance for a broad range of SQL applications.

Db2 gives you the power of data sharing while avoiding overhead, whenever possible, for such things as global locking and data caching. However, you can take additional action to reduce the performance cost of data sharing.

Monitoring tools for Db2 data sharing

Monitoring tools that are available for data sharing environments include Resource Measurement Facility (RMF) reports, Db2 trace, and Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS .

Resource Measurement Facility reports

The Resource Measurement Facility (RMF) provides single system and Parallel Sysplex views by reporting on collected resource usage data

- The Sysplex Summary report provides an integrated view of the entire Parallel Sysplex on one screen.
- The Response Time Distribution report contains details about the distribution of response times on a Parallel Sysplex level and includes the capability of zooming into a single system that indicates problems.
- The Coupling Facility reports include information about storage allocation, structure activity, and subchannel activity which allows you to plan for better resource utilization. For an example report, see “Group buffer pool monitoring with the coupling facility activity report of RMF” on page 176
- The Shared Device report provides information about how disks and tape resources are shared among the different systems in the Parallel Sysplex.

For an example REMF report

Db2 trace

Db2writes trace records to help you monitor events in the data sharing group. Many trace classes include information about locking and use of the group buffer pool. For more information about Db2 trace, see Db2 trace (Db2 Performance).

Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS

Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS can monitor the use of shared Db2 resources such as the catalog and directory, group buffer pools, databases for entire data sharing groups, and databases for individual members of the group. For more information, see Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS .

- Reports and traces with *member scope* present instrumentation for each member of the data sharing group, without merging the data. These reports and traces are similar to those generated in a non-data-sharing environment, where each report or trace is produced for an individual Db2 subsystem.
- Reports and traces with *group scope* merge instrumentation data for individual members and present the data for the entire data sharing group. The traces show events in chronological order and indicate which member is reporting the event, and the reports show events summarized for the entire group.

© Copyright IBM Corp. 1994, 2020

The following report sets provide both group-scope and member-scope reporting:

- Accounting
- Audit
- Locking
- Statistics

Group-scope reporting is also available in exception processing and graphics: you can define exception thresholds for groups, and you can create graphs showing performance trends for an entire data sharing group.

With Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS , you can have processor times reported in service units. This allows times from different central processor complex (CPC) models in the group to be normalized before they are reported.

Related reference

Facilities and tools for Db2 performance monitoring (Db2 Performance)

Improving the performance of data sharing applications

Several actions can help your resource-intensive applications run at their best in the Parallel Sysplex.

Many of the things you currently do for a single Db2 subsystem to improve response time or reduce processor consumption also hold true in the Parallel Sysplex. However, if the processing speed of the CPU is slower than that on which you are currently running, be sure to plan carefully for applications or Db2 utility processes where processor resource consumption represents the significant part of the elapsed time. Some examples are some batch applications, complex queries, and many Db2 utilities.

General recommendation

Take advantage of data partitioning and design for parallel processing whenever possible. Db2 can use parallel processing effectively only when the data is partitioned.

Related concepts

Concurrency and locks in data sharing environments
Concurrency and transaction locks have additional implications in data sharing environments.

Db2 address spaces involved in distributed data processing

With data sharing, as you increase throughput by spreading work across more systems, you can expect some processing increases. These increases are caused by the cost of managing and controlling locks, buffers, and data sets. However, these increases are limited to the percentage of your workload that is accessing data with inter-Db2 read/write interest.

The z/OS address spaces that are involved in distributed database processing with Db2 for z/OS and the purpose of each address space is as follows (*ssnm* is a placeholder for the actual Db2 subsystem name):

- *ssnm*MSTR

  The system services address space is responsible not only for starting and stopping Db2 for z/OS, but for controlling local access to it.

  Activities that occur in this address space include commit processing after updates, inserts, and deletes; logging; backout processing; and archiving. With Db2 data sharing, writes to the group buffer pool and the global unlocking that occurs during commit processing both occur under service request blocks (SRBs) in this address space.

- *ssnm*DBM1

  The database services address space is responsible for accessing relational databases controlled by Db2 for z/OS. The input and output to database resources is performed on behalf of SQL application programs in this space.

Activities that occur in this address space include prefetching, space management, and deferred write. With Db2 data sharing, the following activities occur under SRBs in this address space:

  - Castouts
  - Prefetch interactions with the coupling facility
  - P-lock negotiation
  - Updates to SYSLGRNX
  - Group buffer pool checkpoints

- *ssnm*DIST

  The distributed services address space is responsible for that portion of Db2 for z/OS that provides distributed database capabilities: Distributed Data Facility (DDF).

  When a distributed database request is received, DDF passes the request to *ssnm*DBM1, so that the required database I/O operations can be performed.

- *ssnm*SPAS

  The Db2 stored procedures address space is responsible for processing stored procedures.

- IRLM

  The IRLM address space is responsible for controlling access to database resources.

  Activities that occur in this address space include global lock conflict resolution. When a system is running normally, IRLM's SRB time is substantially lower than either *ssnm*DBM1 or *ssnm*MSTR.

- Allied address space

  The allied agent's address space is responsible for handling global lock requests and requests to read from the group buffer pool.

Migration of batch applications

When migrating a batch application to a data sharing environment, ensure that you account for any changes in processor speed.

Be aware of contention that can occur when there are hot spots in the data as it is updated from multiple members of the group.

Parallel scheduling

If a significant portion of the elapsed time of a long-running batch application is due to processor resource consumption and you are currently having scheduling problems, consider running more than one copy of the same program in parallel. You can run each copy on a different key range, typically the partitioning key of the main table.

If your batch application cannot be redesigned to run on separate partitions, consider running batch jobs on the CPC in the group that has the fastest single-central processor speed. This approach is recommended only if the application does not access the coupling facility at a high intensity.

To avoid disk contention, make sure the data partitions are placed on disk devices with separate I/O paths. Consider using the PIECESIZE option of the CREATE or ALTER INDEX SQL statements to give you more control over the data set size of nonpartitioning indexes.

Designing table spaces for heavy insert and update activity

MEMBER CLUSTER option

If your application does heavy sequential insert processing from multiple members of the group, consider putting the data in a table space that is defined with the MEMBER CLUSTER option. The MEMBER CLUSTER option causes Db2 to insert the data, based on available space rather then respecting the clustering index or the first index. For sequential insert applications, specifying MEMBER CLUSTER can reduce P-lock contention and give better insert performance at the cost of possibly higher query times.

TRACKMOD option

When an application is rapidly updating a single table space, contention can exist on the space map pages as Db2 tracks changes. Db2 tracks these changes to help improve the speed of incremental image copies. With TRACKMOD NO, Db2 does not track changes and thus avoids contention on the space map. This option is not recommended if you depend on fast incremental image copies for your backup strategy. If your application does heavy sequential inserts, consider also using the MEMBER CLUSTER option.

Related concepts

Options for reducing space map page contention

When you define table spaces and indexes, you have several options to help reduce space map hot spots. These hot spots can occur when a large amount of update, insert, or delete activity occurs on a page set from multiple members of a group.

Related reference

Statements (Db2 SQL)

Resource limit facility implications for data sharing

The resource limit facility controls the amount of processor time that SQL statements can consume in Db2.

Different members of a data sharing group can use the same or different resource limit facility tables. Each resource limit table must have a unique name within the data sharing group.

The following controls for the resource limit facility apply only to a single member:

- The following Db2 commands apply only to the member where the command was issued:
  - START RLIMIT
  - STOP RLIMIT
  - DISPLAY RLIMIT
- The following subsystem parameters apply only for the member where the value is set:
  - RLF
  - RLFERR
  - RLFERRD

While a resource limit table is active in any member, the following actions are restricted:

- DROP commands for any object that is associated with the resource limit table.
- STOP DATABASE commands for a database or table space that contains the resource limit tables.
- START DATABASE commands with the ACCESS(UT) option for a database or table space that contains the resource limit tables.

Related concepts

The resource limit facility (Introduction to Db2 for z/OS)

Related tasks

Setting limits for system resource usage by using the resource limit facility (Db2 Performance)

Related reference

Resource limit facility tables (Db2 Performance)

-DISPLAY RLIMIT (Db2) (Db2 Commands)

-START RLIMIT (Db2) (Db2 Commands)

-STOP RLIMIT (Db2) (Db2 Commands)

RLF AUTO START field (RLF subsystem parameter) (Db2 Installation and Migration)

Improving batch processing performance in data sharing

You might improve the performance of large batch processes in a data sharing environment by removing group buffer pool dependency for a table space, index space, or partition.

Procedure

On the same member that runs the batch process, issue an ACCESS DATABASE command for a table space, index space, or partition, and specify the MODE(NGBPDEP) keyword.

Db2 attempts to remove group buffer pool dependency for the specified page set or partition. This operation drains the page set or partition.

Example

The following command physically closes the entire non-partitioned table space DSN9003 of database DSN9001 and removes group buffer pool dependency.

```
-ACCESS DATABASE(DSN9001) SPACENAM(DSN9003) MODE(NGBPDEP)
```

Output similar to the following output indicates that the command completed successfully:

```
  -DSNTDDIS 'ACCESS DATABASE' NORMAL COMPLETION
```

Related reference

-ACCESS DATABASE (Db2) (Db2 Commands)

Resolving synchronous read I/O problems in data sharing environments

If your data sharing environment experiences a high volume of unexpected synchronous read I/O activity, you can take steps to alleviate the problem.

Symptoms

An unexpected and high volume of synchronous read I/O activity in a data sharing environment occurs.

Causes

Two Db2 subsystem parameters establish a threshold that can affect the level of synchronous read I/O activity:

- PCLOSET specifies the number of minutes that can elapse after a page set or partition is updated.
- PCLOSEN specifies the number of consecutive checkpoints that are allowed after a page set or partition is updated.

This situation leads to GBP dependency changes, which means that the pages in the local buffer pool are treated as invalid. If those pages are referenced again, they need to be refreshed, perhaps by synchronous read I/O activity. This situation might be worse if the local buffer pool is defined with PGSTEAL(NONE) because this setting disables prefetch.

Environment

When this problem occurs for a page set or partition in a data sharing environment, the PCLOSEN or PCLOSET subsystem parameter values are reached as a result of activity on that page set or partition.

Resolving the problem

System programmer response: To resolve this problem, you can take either of the following actions:

- Increase the PCLOSET and PCLOSEN subsystem parameter values to avoid frequent GBP-dependency changes.
- Change the definition of the buffer pool so that it specifies PGSTEAL(LRU), which ensures that the prefetch engines are always enabled. Refreshing invalid pages by the prefetch engines for asynchronous read I/O has less performance impact to applications.

Physical open of a page set of partition

To eliminate the overhead of a physical open for an SQL thread, you may want to force a physical open of a table space, index space, or partition to improve performance in a data sharing environment.

You can do this by issuing the ACCESS DATABASE MODE(OPEN) command on the data sharing member where you plan to run your applications. This command will force the physical opening of the specified pageset or partition on the local member only. This will move the overhead of the physical open from a SQL thread to the command thread.

Concurrency and locks in data sharing environments

Concurrency and transaction locks have additional implications in data sharing environments.

Global transaction locking

With data sharing, concurrency control exists both within a specific member and among all members of a data sharing group. This means that locks used in data sharing are *global transaction locks* with a scope that includes the entire data sharing group. Many of these locks are processed by each of the following facilities:

- The local IRLM
- The z/OS cross-system extended services (XES)
- The coupling facility lock structure

Relationship between L-locks and P-locks

Transaction locks are often called *logical locks* (L-locks) in data sharing.

Data sharing also uses *physical locks* (P-locks). But, P-locks are related to caching, not concurrency. These locks use different mechanisms than the transaction locks you are familiar with in Db2.

Inter-member interest occurs between more than one Db2 member in a data sharing group. Inter-member interest is possible on an L-lock but not on a P-lock that is held on the same object. Inter-member interest is also possible on the P-lock but not on the L-lock. The inter-member interest on the page set P-lock, in locking protocol 2, controls both GBP-dependency and child lock propagation.

Locking optimizations for data sharing

Db2 uses the following optimizations to reduce the necessity of processing locks beyond the local IRLM whenever possible.

- Explicit hierarchical locking makes certain optimizations possible. When no inter-member read/write interest exists in an object, it is possible to avoid processing certain locks beyond the local IRLM.
- If a single member with update interest and multiple members with read-only interest exist, Db2 propagates fewer locks than when all members have update interest in the same page set.
- All locks (except L-locks) that are held beyond the local IRLM are owned by a member, not by an individual work unit. This ownership scope reduces lock propagation. It requires that only the most restrictive lock mode for an object on a given member is propagated to XES and the coupling facility. A new lock that is equal to, or less restrictive than, the lock currently being held is not propagated.
- When the lock structure is allocated in a coupling facility of CFLEVEL=2 or higher, IRLM can release many locks with just one request to XES. This release can occur, for example, after a transaction commits and has two or more locks that need to be unlocked in XES. It also can occur at Db2 shutdown or abnormal termination when the member has two or more locks that need to be unlocked.

Partition locks in data sharing

In a partitioned table space, locks are obtained at the partition level. Individual partitions are locked as they are accessed. This locking behavior enables greater concurrency.

Partition locks are always acquired, even if the table space was defined LOCKPART NO clause in a version of Db2 prior to Version 8. The LOCKPART NO clause is deprecated and no longer has any effect.

Each locked partition is a separate parent lock. Therefore, Db2 and IRLM can detect when no inter-Db2 read/write interest exists on that partition and thus do not propagate child L-locks unnecessarily.

Restrictions: If any of the following conditions are true, Db2 cannot selectively lock the partitions. It must lock **all** the partitions:

- The table space is defined with LOCKSIZE TABLESPACE.
- LOCK TABLE IN EXCLUSIVE MODE is used (without the PART option).

Related concepts

Concurrency and locks (Db2 Performance)

Lock size (Db2 Performance)

Related tasks

Improving concurrency (Db2 Performance)

Explicit hierarchical locking in data sharing

In Db2 data sharing, Db2 uses explicit hierarchical locking to determine whether propagating L-locks beyond the local IRLM to XES and to the coupling facility is necessary.

With *explicit hierarchical locking*, IRLM grants child locks locally when no inter-Db2 read/write interest exists on the parent. Granting lock requests locally, versus globally, improves performance.

Locks on the higher objects are *parent locks*. Locks on lower level objects are *child locks*, with the caveat that a child lock can be the parent of another child lock. While a lock is held, the first lock on the top parent is always propagated to XES and the lock structure. Thereafter, only more restrictive locks are propagated. When the lock is released, the process begins again. For partitioned table spaces, each locked partition is a parent of the child locks that are held for that partition. Explicit hierarchical locking is based on the lock hierarchy that is shown in the table below.

Table 25. Lock hierarchy. Indexes are not included in the hierarchy because index pages are protected by locks on the corresponding data.

Parent	Children or child
UTS or partitioned table space	Data pages and rows
Segmented table space	Data pages and rows
Tables in a segmented table space	N/A
LOB table space	LOB
XML table space	XML document
Simple table space	Data pages and rows

When you use locking protocol 1, locks on child objects are based on contention on the parent L-lock. When you use locking protocol 2, locks on child objects are propagated depending on the compatibility of the page set P-lock with the page set P-locks that are held by other members for the table space.

The following table shows the conditions that cause the child locks to be propagated when locking protocol 1 is used.

Table 26. Determining when child locks are propagated to XES when locking protocol 1 is used

Maximum page set L-lock mode of this member is ...	And the maximum page set L-lock mode of other members is...	Are X, L-child locks propagated by this member?	Are S, L-child locks propagated by this member?	Are U, L-child locks propagated by this member?
IS, S	None, IS, S	N/A	No	N/A
X	None	N/A	N/A	N/A
IS	IX, SIX	N/A	Yes	N/A
IX, SIX	IS	Yes	No	Yes
IX	IX	Yes	Yes	Yes
IX, SIX	None	No	No	No

The following table shows the conditions that cause the child locks to be propagated when locking protocol 2 is used.

Table 27. Determining when child locks are propagated to XES when locking protocol 2 is used

Maximum page set P-lock mode of this member is ...	And the maximum page set P-lock mode of the other members is...	Are X, L-child locks propagated by this member?	Are S, L-child locks propagated by this member?	Are U, L-child locks propagated by this member?
IS, S	None, IS, S	N/A	No	Yes
X	None	No	No	No
IS	IX, SIX	N/A	Yes	Yes
IX, SIX	IS	Yes	No	Yes
IX	IX	Yes	Yes	Yes
SIX	None	Yes	No	No
IX	None	No	No	No

Notes:

- Some child L-locks might be acquired before the page set P-lock is obtained. When this happens, child L-locks are automatically propagated.
- When a page set switches from inter-system read/write interest to no inter-system read/write interest, a short period of time exists when the page set remains GBP-dependent, before the P-lock reverts to X state. During this time, child L-locks continue to be propagated.

Related concepts

A locking scenario
Looking at an example of locking activity between two members of a data sharing group can help you to understand locking in a data sharing environment.

Lock size (Db2 Performance)

Lock modes and compatibility of locks (Db2 Performance)

A locking scenario

Looking at an example of locking activity between two members of a data sharing group can help you to understand locking in a data sharing environment.

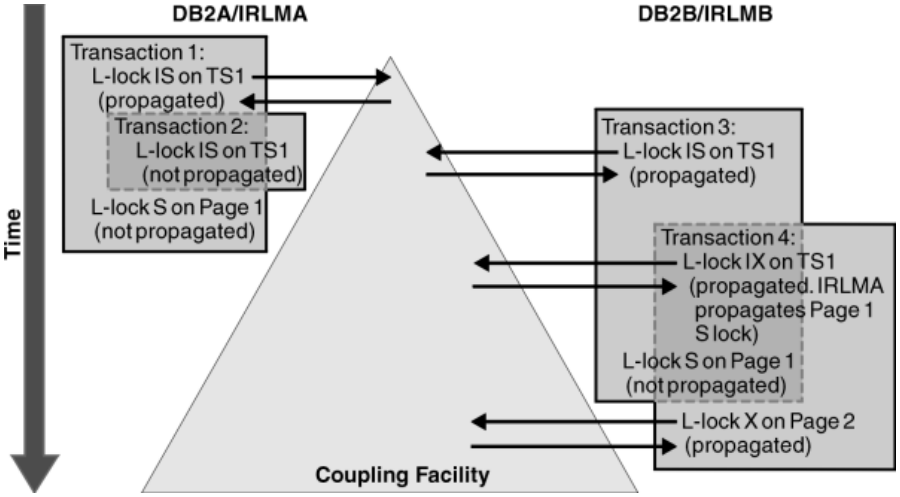


Figure 27. A lock propagation scenario

In the figure above:

1. The L-lock on table space 1 (TS1), which is associated with transaction 2, is not propagated because DB2A already holds a lock of an equal restriction on that object. (The L-lock on TS1, which is associated with transaction 3, is propagated because that lock is from another member.)
2. The child L-lock on page 1, which is associated with transaction 1, is not propagated at the time that it is requested because its parent lock is IS on both members: no inter-Db2 read/write interest exists on the parent.
3. When Transaction 4 upgrades its lock to IX on TS1, its X lock on Page 2 must be propagated because inter-Db2 read/write interest now exists on the parent. Also, the child lock on Page 1, which is associated with transaction 1, must be propagated.

   The child lock is propagated under an IRLM SRB, not under the transaction's TCB. This propagation is counted in the statistics report as an asynchronous propagation, as shown in B in the Db2 OMEGAMON statistics trace.

Related reference

Lock monitoring with the Db2 statistics trace
The Db2 statistics trace provides counters that track the amount of global locking activity and contention that each member in the data sharing group is encountering.

Traces that indicate whether locks have been propagated

The statistics and accounting traces indicate the number of global locks that have been propagated to XES.

The ratio of this number to the total number of global locks requested reflects the effects of explicit hierarchical locking and other locking optimizations.

Improving concurrency in data sharing environments

You can apply certain recommendations to reduce lock contention and improve concurrency in data sharing environments.

Before you begin

Before applying these particular data-sharing recommendations, ensure that you have applied the basic recommendations for improving concurrency. For information about the basic recommendations, see Improving concurrency (Db2 Performance).

About this task

Lock requests, along with group buffer pool requests, are the most critical factors for data sharing performance. To reduce locking contention in data sharing environments, you can apply the same approaches for improving concurrency outside of data sharing. Some of the general recommendations are repeated below with additional emphasis, because lock avoidance is more important in data sharing environments.

Procedure

You can use any of the following approaches to improve concurrency in a data sharing environment:

- Specify the TRACKMOD NO and MEMBER CLUSTER options when you create table spaces.

  These options can reduce p-lock and page latch contention on space map pages during heavy inserts into GBP-dependent table spaces. TRACKMOD NO cannot be used when incremental image copies are used for the table spaces.
- When the MEMBER CLUSTER option is used, consider the use of LOCKSIZE ROW for insert-intensive workloads.

  Row-level locking might provide additional relief in the following forms:

  – Reduced page P-lock and page latch contention on data pages
  – Better space usage
  – Reduced working set of pages in the buffer pool

  However, do not use LOCKSIZE ROW without the MEMBER CLUSTER option for insert-intensive workloads that frequently insert data at the end of the table space. Doing so might result in excessive page p-lock contention on data pages and space map pages, and the extra locking protocol from taking page p-locks.
- Use LOCKSIZE ANY or PAGE as a design default. Consider LOCKSIZE ROW only when applications encounter significant lock contention, including deadlock and timeout.

  LOCKSIZE ANY is the default for CREATE TABLESPACE. It allows Db2 to choose the lock size, and Db2 usually chooses LOCKSIZE PAGE and LOCKMAX SYSTEM for non-LOB/non-XML table spaces. For LOB table spaces, Db2 chooses LOCKSIZE LOB and LOCKMAX SYSTEM. Similarly, for XML table spaces, Db2 chooses LOCKSIZE XML and LOCKMAX SYSTEM.

  Page-level locking generally results in fewer requests to lock and unlock data for sequential access and manipulation, which translates to reduced CPU cost. Page-level locking is also more likely to result in sequentially inserted rows in the same data page. Row-level locking with MAXROWS=1 can suffer from data page p-locks in data sharing environments. However, page-level locking can avoid the data page p-locks when MAXROWS=1.

  Row-level locking provides better concurrency because the locks are more granular. However, the cost of each lock and unlock request is roughly the same for both page and row-level locking. Therefore, row-level locking is likely to incur additional CPU cost. Row-level locking might also result in more data page latch contention. Sequentially inserted rows, by concurrent threads, are less likely to be in the same data page under row-level locking.
- Use the RELEASE(DEALLOCATE) bind option to avoid the cost of repeatedly releasing and reacquiring locks for applications that use frequent commit points for repeated access to the same table spaces.

  For objects that do not have much concurrent activity within a member, this option avoids the cost of repeatedly releasing and reacquiring the same parent locks. You can also reduce the amount of false contention for transactions that use the thread.
- Use the RELEASE(COMMIT) bind option for plans or packages that are used less frequently to avoid excessive increases to the EDM pool storage, lock storage, and thread-related storage.
- Consider using randomized index key columns.

  In a data sharing environment, you can use randomized index key columns to reduce locking contention at the possible cost of more CPU usage, from increased locking and getpage operations, and more index page read and write I/Os.

This technique is effective for reducing contention on certain types of equality predicates. For example, if you create an index on a timestamp column, where the timestamp is always filled with the current time, every insertion on the index would be the greatest value and cause contention on the last index leaf page. An index on a column of sequential values, such as invoice numbers, causes similar contention, especially in heavy transaction workload environments. In each case, using the RANDOM index order causes the values to be stored at random places in the index tree, and reduce the chance that consecutive insertions hit the same index leaf page and cause contention.

Although the randomized index can relieve contention problems for sets of similar or sequential values, it does not help with identical values. Identical values encode the same and each are inserted at the same place on the index tree.

Related tasks

Choosing a RELEASE option (Db2 Performance)

Programming for concurrency (Db2 Performance)

Disabling update locks for searched UPDATE and DELETE

You can use the XLKUPDLT subsystem parameter to disable update locks on searched UPDATE and DELETE statements.

Avoiding false contention

The Db2 coupling facility lock structure has two parts: a lock table, which is used to determine whether inter-Db2 read/write interest exists on a particular resource, and a list of the update locks that are currently held.

When considering false contention, you must consider the size of the lock table. The total size of the lock structure determines the size of the lock table. Assuming that you specify an INITSIZE value on the CFRM policy that is a power of 2, the lock table is allocated to one-half the total size of the lock structure. The value that you specify for the lock table entries (LTE) parameter in the IRLMPROC or with the MODIFY irlmproc,SET,LTE command can control how the lock structure is partitioned.

The number of members in the group determines the size of each entry in that lock table. IRLM uses the value that you specify in the LOCK ENTRY SIZE field of installation panel DSNTIPJ to determine the initial size of the lock table entries. IRLM also uses the value that you specify in the NUMBER OF LOCK ENTRIES field of installation panel DSNTIPJ to determine how the lock structure is initially partitioned.

IRLM assigns locked resources to an entry value in the lock table. This is how it can quickly check to see if a resource is already locked. If the lock structure is too small (thereby making the lock table portion too small), many locks can be represented by a single value. Thus, "false" lock contention can exist. *False lock contention* is where two different locks on different resources hash to the same lock entry. The second lock requester is suspended until it is determined that no real lock contention exists on the resource.

False contention can be a problem for workloads that have heavy inter-Db2 read/write interest.

One way to reduce false contention is to increase the size of the lock structure or to increase the proportional size of the lock table.

Related reference

LOCK ENTRY SIZE field (Db2 Installation and Migration)

Monitoring for false contention

You can determine the amount of false contention by using the RMF Coupling Facility Activity reports.

Db2 also provides necessary information in its accounting and statistics trace classes. More detailed information can be found in the performance trace.

Related reference

Lock monitoring with the coupling facility structure activity report

The Coupling Facility Activity Report of RMF describes activity to all structures in the coupling facility for a given time period.

Lock monitoring with the Db2 performance trace

The Db2 performance trace gives more detailed information about which shared resources are experiencing contention.

Lock monitoring with the Db2 accounting trace
Use the Db2 accounting trace to determine which users or plans are experiencing global lock contention.

Lock monitoring with the Db2 statistics trace
The Db2 statistics trace provides counters that track the amount of global locking activity and contention that each member in the data sharing group is encountering.

How much contention is acceptable

For the best performance, you want to achieve the least possible amount of global lock contention, both real and false.

Global lock contention includes false contention, XES contention, and IRLM contention. Aim for total global lock contention of less than 5%. When the global lock contention is less than 5%, the individual breakdown of false contention, XES contention, and IRLM contention is not significant.

Use the following guidelines to monitor your global lock contention.

- If the global lock contention is less than 3%, no action is necessary.
- If the global lock contention is 3 - 5%, no immediate action is necessary, but you should monitor your system.
- If the global lock contention is greater than 5%, identify the greatest contributor among false contention, XES contention, and IRLM contention, and tune your system accordingly.
- Ignore individual intervals with very low locking rates but very high false contention. This situation is not significant, and it often occurs on systems that use a one-minute statistics interval.

Related concepts

Ways to monitor Db2 locking activity
With data sharing, it is essential to control the volume of global lock requests that are propagated to the coupling facility and to control the amount of lock contention, both real and false.

How to reduce false contention

Several tips and recommendations can help you reduce false contention.

The following tips can help you reduce false contention:

- As much as possible, reduce the amount of real lock contention in your applications.
- Specify a larger size for the lock structure and manually rebuild it.
- Ensure that the value for LOCK ENTRY SIZE is not too large for the number of members in your group.

  The LOCK ENTRY SIZE parameter for the first IRLM to join the group determines the size of each lock entry in the lock table.

  A lock entry size of 2 allows twice as many lock entries as a lock entry size of 4, as illustrated in the figure below.

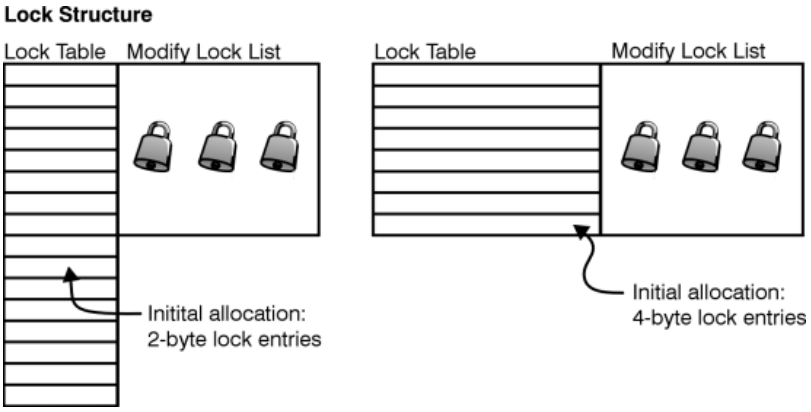


Figure 28. Initial lock entry size

IRLM automatically rebuilds the structure when it needs to. For example, assume MAXUSRS is set to 7. When the 8th member joins the group, IRLM automatically rebuilds the lock structure to create the 4-byte lock entries to handle up to 23 members. IRLM automatically rebuilds the lock structure again when the 24th member joins the group, to create 8-byte lock entries to handle up to 32 members.

For this reason, even if you anticipate your group growing beyond seven members, you can start out with a lock entry size of 2 make the most efficient use of lock structure storage.

Recommendation: Set MAXUSRS for all IRLM instances to an appropriate value for the number of members that you expect to have in a data sharing group. For example, if you plan to have between 8 and 23 members in your data sharing group on a regular basis, you should set MAXUSRS=23 in the IRLMPROC procedure for the IRLM instance that is associated with each member of the data sharing group. Alternatively, you can set the LOCK ENTRY SIZE field value to 4 in installation panel DSNTIPJ when you install each member of the data sharing group. Doing this causes the Db2 installation process to set the MAXUSRS value to 23 in the associated IRLMPROC procedures. Lock structure allocation occurs when the first member joins the data sharing group after an IPL of the operating system, or after any situation that causes the lock structure to be deallocated.

If you initially set MAXUSRS=23 for all IRLM instances, the lock structure size is already adequate for up to 23 members, no matter which IRLM is the first to connect. A lock structure rebuild is not necessary when the eighth member joins the group.

If you increase the lock entry size (and thereby increase MAXUSRS), increase the lock structure size, to maintain the number of lock table entries and record list entries. If you do not increase the lock structure size, IRLM obtains the storage that it needs for the increased lock entry size from storage for lock table entries or record list entries.

- Using locking protocol 2 can reduce false lock contention for parent L-locks when you request lock modes IS or IX. Locking protocol 2 reduces overhead from data sharing processing but might increase child lock propagation to the lock structure in the coupling facility. Propagation is based on the page set P-lock instead of the parent L-lock.

  If a data sharing group uses locking protocol 1, and you want to enable locking protocol 2, you must first quiesce that data sharing group. To activate locking protocol 2, you must initiate a group-wide shutdown. The restart of the first member after a group-wide shutdown changes the group protocol level from 1 to 2.

  To determine whether locking protocol 2 is enabled, issue the DISPLAY GROUP command.

Related tasks

Changing the size of the lock structure by rebuilding
One way to change the size of a lock structure is to rebuild the structure.

Related reference

LOCK ENTRY SIZE field (Db2 Installation and Migration)
-DISPLAY GROUP (Db2) (Db2 Commands)

Decreasing lock entry size

IRLM does not automatically rebuild if the number of members decreases.

About this task

As part of the process for decreasing lock entry size, you must restart the members, which results in a group restart. Because you quiesce work before changing the lock entry size, the group restart should be relatively quick. Nonetheless, decreasing the lock entry size is a disruptive procedure. Consider doing this only in situations when the lock entry size is set too high for the number of members in the group, and you cannot alleviate the false contention within the storage boundaries you have.

Procedure

To decrease the lock entry size:

1. Quiesce all members of the group, using the following command:

   ```
   -DB1A STOP DB2 MODE(QUIESCE)
   ```

2. If IRLM is not automatically stopped along with Db2, enter the z/OS command:

   ```
   STOP irlmproc
   ```

3. Force all connections to the lock structure to disconnect by issuing the following z/OS command:

   ```
   SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
   ```

4. Force the deallocation of the lock structure by issuing the following z/OS command:

   ```
   SETXCF FORCE,STRUCTURE,STRNAME=strname
   ```

5. Change the lock entry size for at least one IRLM. (You should change the value for all of them.)

   If you change the IRLM startup procedure directly, the parameter that you change is called MAXUSRS. The value of LOCK ENTRY SIZE is translated during the Db2 installation or migration process. The value that you enter on the IRLM parameter directly is not the same as the value you put in the LOCK ENTRY SIZE field of installation panel DSNTIPJ.

6. Start the member and IRLM that have the updated value. (You must start the updated member first.)
7. Start all other members.

Related reference

-STOP DB2 (Db2) (Db2 Commands)
STOP irlmproc (z/OS IRLM) (Db2 Commands)
z/OS SETXCF FORCE command (MVS System Commands)
LOCK ENTRY SIZE field (Db2 Installation and Migration)
START irlmproc (z/OS IRLM) (Db2 Commands)

How z/OS resolves contention problems

When contention exists on a hash class, z/OS uses XCF messages to resolve the conflict.

This is how it determines which specific resources are involved in the contention, or if the contention is false. For speedy resolution of contention situations, ensure no queuing of messages exists for XCF message buffers. You can use the XCF Activity Report of RMF to detect this queuing.

Related information

Tuning a Sysplex (z/OS MVS Setting Up a Sysplex)

Disabling update locks for searched UPDATE and DELETE

You can use the XLKUPDLT subsystem parameter to disable update locks on searched UPDATE and DELETE statements.

About this task

PSPI

By specifying the use of X-locks for searched UPDATE and DELETE operations, you can save the cost of additional requests to upgrade locks that are used for searching when rows are updated or deleted. The cost savings of this approach are greatest in data sharing environments because the main cost of the request to upgrade the locks is from recording the request in the coupling facility.

Procedure

Set the value of the XLKUPDLT subsystem parameter when most or all searched UPDATE and DELETE statements use an index or can be evaluated by stage-1 processing.

- When you specify YES, Db2 uses an X-ock on rows or pages that qualify during stage 1 processing. With CS isolation, the lock is released if the row or page is not updated or deleted because it is rejected by stage 2 processing. With RR isolation or RS isolation, Db2 acquires an X-lock on all rows that fall within the range of the selection expression. Thus, a lock upgrade request is not needed for qualifying rows, though the lock duration is changed from manual to commit. The lock duration change is not as costly as a lock upgrade.
- When you specify TARGET, Db2 treats the rows or pages of the specific table targeted by the update or delete as if the value of XLKUPDLT was YES. It treats rows or pages of other tables referenced by the query, such as those in referenced only in the WHERE clause, as if XLKUPDLT was set to NO. By specifying this blended processing, you can prevent time outs caused by strong lock acquisition of the read-only non-target objects referenced in the update or delete statement.
- When you specify NO, Db2 might use lock avoidance when scanning for qualifying rows. When a qualifying row is found, an S lock or a U lock is acquired on the row. The lock on any qualifying row or page is then upgraded to an X-lock before performing the update or delete. For stage-1 non-qualifying rows or pages, the lock is released if CS or RS isolation is used. For RR isolation, an S-lock is retained on the row or page until the next commit point. This option is best for achieving the highest rates of concurrency. PSPI

Related concepts

Lock modes and compatibility of locks (Db2 Performance)

Predicates and access path selection (Db2 Performance)

Related tasks

Improving concurrency for update and delete operations (Db2 Performance)

Choosing an ISOLATION option (Db2 Performance)

Related reference

X LOCK FOR SEARCHED U/D field (XLKUPDLT subsystem parameter) (Db2 Installation and Migration)

ISOLATION bind option (Db2 Commands)

isolation-clause (Db2 SQL)

Deadlock detection and resource timeouts in data sharing environments

The deadlock detection intervals and resource timeout value of your data sharing environment should be based on how deadlock detection and resource timeouts work in data sharing environments.

Recommendations: Quick detection of deadlocks and timeouts is necessary in a data sharing environment to prevent a large number of waiters on each system. A large numbers of waiters can cause much longer wait times for timeouts and deadlocks. The following are two recommendations to help prevent a large number of waiters from developing on each system:

- If your non-data-sharing Db2 subsystem has a problem with deadlocks, consider reducing the deadlock time to prevent a long lists of waiters from developing. (If you do not have a problem with deadlocks, you most likely will not have to change any parameters for data sharing.)
- If you have stringent timeout limits that must be honored by Db2, consider decreasing the deadlock time before moving to data sharing, as illustrated in this example:

  Assume that you have set the timeout period for your non-data-sharing Db2 subsystem to 55 seconds because you want the wait time for timeout to be at or before 60 seconds. (This assumes that your deadlock time value is five.) In a data sharing environment, reduce the timeout period to 40 seconds. This makes it more likely that your actual wait time for timeouts is at or before 60 seconds.

Global deadlock processing
In a data sharing environment, deadlocks can occur between transactions on different members.

The term *global deadlock* refers to the situation where two or more members are involved in the deadlock. *Local deadlock* refers to the situation where all of the deadlocked transactions reside on a single member.

Controlling deadlock detection
Use the DEADLOCK parameter in the IRLM startup procedure to control how often IRLM does its deadlock detection processing.

Specify the parameter as follows:

```
DEADLOCK='x,y'
```

x
The number of seconds between two successive scans for a local deadlock (DEADLOCK TIME value on installation panel DSNTIPJ). The default is 1 second. Values can range 1 - 5 seconds, or 100 to 5000 milliseconds.

y
The number of local scans that occur before a scan for global deadlock starts (DEADLOCK CYCLE value on installation panel DSNTIPJ). IRLM always uses a value of 1.

Global deadlock detection requires the participation of all IRLM members in the data sharing group. Each IRLM member has detailed information about the locks that are held and are being waited for by the transactions on its associated Db2 member. However, to provide global detection and timeout services, each IRLM is informed of all requests that are waiting globally so that the IRLM can provide information about its own blockers. That IRLM also provides information about its own waiters. The IRLM members use XCF messages to exchange information so that each member has this global information.

The global deadlock manager
To coordinate the exchange of information, one IRLM member assumes the role of the global deadlock manager.

As IRLM members join or leave the group, the global deadlock manager might change.

The local deadlock detector
Each IRLM member in the group must participate in the global deadlock detection process.

Each IRLM member (including the one designated as the global deadlock manager) has the role of local deadlock detector.

Relationship between local and global deadlock detection
The four XCF messages represent one global detection cycle, which usually takes two to four *x*-second intervals to complete (where *x* is the number of local cycles).

Four XCF messages are required to gather and communicate the latest information from the local deadlock detectors:

1. The local deadlock detector sends its information about lock waiters to the global deadlock manager.

2. The global deadlock manager takes that information from all local deadlock detectors and sends messages to each of the IRLMs in the group. (Because the global deadlock manager is also a local deadlock detector, it receives the same information, although somewhat quicker than the rest of the IRLMs.)
3. Each local deadlock detector checks the global view of resources and determines if it has blockers for other waiters. It passes that information along to the global deadlock manager with its list of waiters.
4. The global deadlock manager, from the information it receives from the local deadlock detectors, determines if a global deadlock or timeout situation exists. If a global deadlock situation exists, Db2 chooses a candidate for the deadlock. The global deadlock manager also determines if any timeout candidate is blocked by an incompatible waiter or holder and, if so, presents that candidate to the owning IRLM, along with any deadlock candidates belonging to that IRLM. When Db2 receives this information, it determines if it should request that IRLM reject any given timeout candidate waiter.

The following figure illustrates an example in which the deadlock time value is set to 5 seconds.

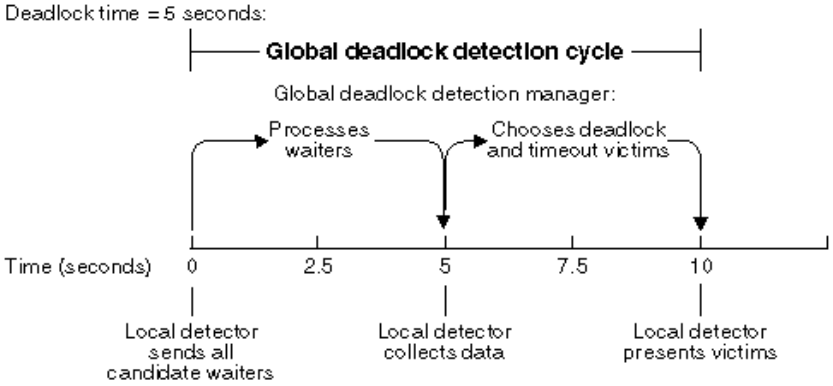


Figure 29. Global deadlock detection cycle

Deadlock detection might be delayed if any of the IRLMs in the group encounter any of the following conditions:

- XCF signaling delays
- IRLM latch contention (can be encountered in systems with extremely high IRLM locking activity)
- A large number of global waiters

Global timeout processing
Just as in a non-data-sharing environment, Db2 calculates the timeout period based on the RESOURCE TIMEOUT and DEADLOCK TIME installation parameter values.

Db2 calculates the timeout period as follows:

1. Divide RESOURCE TIMEOUT by DEADLOCK TIME
2. Round to the next largest integer
3. Multiply that integer by DEADLOCK TIME

In non-data-sharing systems, the actual time that a transaction waits on a lock before timing out varies between the timeout period and the timeout period plus one DEADLOCK TIME interval.

For example, if the timeout period for a given transaction is 60 seconds and the DEADLOCK TIME value is 5 seconds, the transaction waits between 60 and 65 seconds before timing out, with the average wait time of 62.5 seconds. This is because timeout is driven by the deadlock detection process, which is activated on a timer interval basis.

Elapsed time until timeout, non-data-sharing
The timeout period for a local timeout is typically shorter than that of a global timeout.

The actual time a process waits until timing out usually falls within the following range:

```
MIN LOCAL TIMEOUT = timeout period
MAX LOCAL TIMEOUT = timeout period + DEADLOCK TIME value
AVERAGE LOCAL TIMEOUT = timeout period + DEADLOCK TIME value/2
```

However, the maximum or average values can be larger, depending on the number of waiters in the system or if a heavy IRLM workload exists.

Elapsed time until timeout, data sharing

In a data sharing environment, because the deadlock detection process sends inter-system XCF messages, a given transaction typically waits somewhat longer before timing out than in a non-data-sharing environment.

How much longer a transaction waits depends on where in the global deadlock detection cycle that the timeout period actually expired. However, the length of time a process waits until timing out generally falls within the following range:

```
MIN GLOBAL TIMEOUT = timeout period + DEADLOCK TIME value
MAX GLOBAL TIMEOUT = timeout period + 4 * DEADLOCK TIME value
AVERAGE GLOBAL TIMEOUT = timeout period + 2 * DEADLOCK TIME value
```

Again, the maximum or average values might be larger.

Ways to monitor Db2 locking activity

With data sharing, it is essential to control the volume of global lock requests that are propagated to the coupling facility and to control the amount of lock contention, both real and false.

You must monitor both the amount and type of locking that your applications are doing, and you must also make sure that any locking problems are not caused by data sharing resources, such as an undersized lock structure, or the overuse of the coupling facility or coupling facility channels (links).

The z/OS command D XCF,STRNAME can also be used to monitor lock structure activity.

Related reference

Displaying information about specific structures
The STRNAME keyword of the z/OS D XCF, STR command displays detailed information about specific structures.

Lock monitoring with the DISPLAY DATABASE command

Use the LOCKS ONLY option on DISPLAY DATABASE to display information about page set, partition, or table locks that are held on resources.

The "lock" column of the display describes the type and duration of locks used by corresponding agents.

GUPI

The following figure is an example of output of DISPLAY DATABASE for a table space. The application identified as LSS001 on member DB1A has locked partitions 1 and 2. LSS002 on member DB2A has locked partitions 1 and 3. Partition 4, which has no locks, is not on the display because the ONLY option of DISPLAY DATABASE was used.

```
NAME     TYPE PART STATUS              CONNID   CORRID       LOCKINFO
-------- ---- ---- ------------------ -------- ------------ ---------
TSPART   TS    01  RO                 LSS001   DSN2SQL      H-IS,P,C
     -                  MEMBER NAME DB1A
TSPART   TS    01  RO                                       H-S,PP,I
     -                  MEMBER NAME DB1A
TSPART   TS    01  RO                 LSS002   DSN2SQL      H-IS,P,C
     -                  MEMBER NAME DB2A
TSPART   TS    01  RO                                       H-S,PP,I
     -                  MEMBER NAME DB2A
TSPART   TS    02  RW                 LSS001   DSN2SQL      H-IS,P,C
     -                  MEMBER NAME DB1A
TSPART   TS    02  RW                                       H-S,PP,I
     -                  MEMBER NAME DB1A
TSPART   TS    03  RW                 LSS002   DSN2SQL      H-IS,P,C
     -                  MEMBER NAME DB2A
TSPART   TS    03  RW                                       H-S,PP,I
     -                  MEMBER NAME DB2A
```

Figure 30. Example DISPLAY DATABASE LOCKS for a table space

GUPI

Lock monitoring with the coupling facility structure activity report

The Coupling Facility Activity Report of RMF describes activity to all structures in the coupling facility for a given time period.

The following figure shows a partial report, giving information about:

- A: Total number of lock-related requests.
- B: Number of requests that were deferred because of contention.
- C: The number of deferred requests that were caused by false contention.

```
                                    COUPLING FACILITY STRUCTURE ACTIVITY

STRUCTURE NAME = DSNDB0A_LOCK1      TYPE = LOCK
          #REQ    --------------    REQUESTS ------------  ------ DELAYED REQUESTS
-----------
SYSTEM    TOTAL           #     % OF -SERV TIME(MIC)- REASON  #  % OF  ---AVG TIME(MIC)---
EXTERNAL REQUEST
NAME      AVG/SEC         REQ    ALL      AVG  STD_DEV        REQ  REQ   /DEL   STD_DEV /ALL
CONTENTIONS
STLABC2    126K  SYNC   126K   51.0%     61.0   22.5
REQ TOTAL  A 162K
          701.7  ASYNC   0     0.0%       0.0    0.0   NO SCH 0   0.0%   0.0   0.0   0.0
REQ Deferred 612
                 CHNGD   0     0.0%  INCLUDED IN ASYNC
-CONT     B  621

-FALSE CONT C 212
```

Figure 31. Partial RMF Coupling Facility Activity report for lock structure

Calculating contention percentages

You can use a formula to calculate the percentage of global contention and false contention in your data sharing environment.

Use the following calculations:

- Total contention is the number of deferred requests (B) divided by the total number of requests (A), multiplied by 100. So, for this example:

  ```
  (621 / 162000) × 100 = .387%
  ```

  This indicates that the global contention rate is approximately 0.39 % (a good figure).

- False contention is the number of false contentions (C) divided by the total number of requests (A) multiplied by 100. For this example:

```
(212 / 162000) × 100 = 0.13%
```

  Thus, the rate of false contention is 0.13 % (a very good figure).

Lock monitoring with the Db2 statistics trace

The Db2 statistics trace provides counters that track the amount of global locking activity and contention that each member in the data sharing group is encountering.

This trace runs with low overhead. Keep the Db2 statistics trace turned on to allow continuous monitoring of each subsystem.

The following figure shows the type of information that is provided by a statistics trace.

```
DATA SHARING LOCKING            QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------       --------  -------  -------  -------
GLOBAL CONTENTION RATE (%)          0.19  A
FALSE CONTENTION RATE (%)           0.04  B
P/L-LOCKS XES RATE (%)             36.47

LOCK REQUESTS (P-LOCKS)            107.4K    44.74     0.93     0.18
UNLOCK REQUESTS (P-LOCKS)          105.2K    43.82     0.91     0.18
CHANGE REQUESTS (P-LOCKS)         1453.00     0.61     0.01     0.00

SYNCH.XES - LOCK REQUESTS    C     1478.8K   616.16    12.77     2.48
SYNCH.XES - CHANGE REQUESTS  D    45450.00    18.94     0.39     0.08
SYNCH.XES - UNLOCK REQUESTS  E      810.0K   337.48     7.00     1.36
BACKGROUND.XES -CHILD LOCKS  F        0.00     0.00     0.00     0.00
ASYNCH.XES -CONVERTED LOCKS  G      108.00     0.04     0.00     0.00

SUSPENDS - IRLM GLOBAL CONT  H     3236.00     1.35     0.03     0.01
SUSPENDS - XES GLOBAL CONT.  I      334.00     0.14     0.00     0.00
SUSPENDS - FALSE CONT. MBR   J      863.00     0.36     0.01     0.00
SUSPENDS - FALSE CONT. LPAR  J         N/A      N/A      N/A      N/A
REJECTED - XES                    31897.00    13.29     0.28     0.05
INCOMPATIBLE RETAINED LOCK            0.00     0.00     0.00     0.00
NOTIFY MESSAGES SENT                344.00     0.14     0.00     0.00
NOTIFY MESSAGES RECEIVED            106.00     0.04     0.00     0.00
P-LOCK/NOTIFY EXITS ENGINES         500.00      N/A      N/A      N/A
P-LCK/NFY EX.ENGINE UNAVAIL           0.00     0.00     0.00     0.00

PSET/PART P-LCK NEGOTIATION          38.00     0.02     0.00     0.00
PAGE P-LOCK NEGOTIATION            3026.00     1.26     0.03     0.01
OTHER P-LOCK NEGOTIATION           1520.00     0.63     0.01     0.00
P-LOCK CHANGE DURING NEG.          3072.00     1.28     0.03     0.01
```

Figure 32. Data sharing locking block of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics trace

The following table contains an explanation of the fields that are shown in the previous figure.

Table 28. Explanation of the fields in a Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics trace

Field	Explanation
A	The global contention rate.
B	The false contention rate.
C, D, E	These counters indicate the total number of lock, change, and unlock requests (including L-locks and P-locks) that were propagated to XES synchronously.
F	The number of resources (including L-locks and P-locks) that were propagated by IRLM to XES asynchronously to the original request. This situation occurs when new inter-Db2 interest occurs on a parent resource, or when a request completes after the requestor's execution unit has been suspended.

Table 28. Explanation of the fields in a Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics trace (continued)

Field	Explanation
G	The number of synchronous-to-asynchronous heuristic conversions for lock requests in XES. This conversion occurs when XES determines that it is more efficient to send the request asynchronously to the coupling facility.
H	The number of real contentions, as detected by IRLM.
I	The number of real contentions, as detected by XES, that were not IRLM-level contentions. IRLM has knowledge of more lock types than XES. Thus, IRLM often resolves contention that XES cannot. The most common example of XES-level contention is usually the intent locks (IS and IX) on the parent L-locks. IS and IX are compatible to IRLM but not to XES. Another common example is the U and S page L-locks; U and S are compatible to IRLM, but not to XES.
J	The total number of false contentions for LOCK and UNLOCK requests at the subsystem (member) level or the LPAR level. This information is helpful when the LPAR contains more than one Db2 member. The QTGSFCON flag indicates whether the false contention is reported at the subsystem level or the LPAR level.

Related concepts

Statistics trace (Db2 Performance)

Related tasks

Controlling the Db2 trace (Db2 Administration Guide)

Related reference

Statistics Report and Trace Blocks (Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS)

Related information

Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS

Calculating global contention percentages with the Db2 statistics trace

You can use the Db2 statistics trace to calculate global contention percentages for lock requests.

Procedure

To calculate global contention:

1. Determine the total number of suspends that are caused by contention.
2. Determine the total number of requests that went to XES.
3. Divide the total number of suspends by the total number of requests.
4. Multiply the result by 100.

Example

```
DATA SHARING LOCKING            QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------      --------  -------  -------  -------
GLOBAL CONTENTION RATE (%)          0.19  A
FALSE CONTENTION RATE (%)           0.04  B
P/L-LOCKS XES RATE (%)             36.47

LOCK REQUESTS (P-LOCKS)            107.4K    44.74     0.93     0.18
UNLOCK REQUESTS (P-LOCKS)          105.2K    43.82     0.91     0.18
CHANGE REQUESTS (P-LOCKS)         1453.00     0.61     0.01     0.00

SYNCH.XES - LOCK REQUESTS    C     1478.8K   616.16    12.77     2.48
SYNCH.XES - CHANGE REQUESTS  D    45450.00    18.94     0.39     0.08
SYNCH.XES - UNLOCK REQUESTS  E      810.0K   337.48     7.00     1.36
BACKGROUND.XES -CHILD LOCKS  F        0.00     0.00     0.00     0.00
ASYNCH.XES -CONVERTED LOCKS  G      108.00     0.04     0.00     0.00

SUSPENDS - IRLM GLOBAL CONT  H     3236.00     1.35     0.03     0.01
SUSPENDS - XES GLOBAL CONT.  I      334.00     0.14     0.00     0.00
SUSPENDS - FALSE CONT. MBR   J      863.00     0.36     0.01     0.00
SUSPENDS - FALSE CONT. LPAR  J         N/A      N/A      N/A      N/A
REJECTED - XES                    31897.00    13.29     0.28     0.05
INCOMPATIBLE RETAINED LOCK            0.00     0.00     0.00     0.00
NOTIFY MESSAGES SENT                344.00     0.14     0.00     0.00
NOTIFY MESSAGES RECEIVED            106.00     0.04     0.00     0.00
P-LOCK/NOTIFY EXITS ENGINES         500.00      N/A      N/A      N/A
P-LCK/NFY EX.ENGINE UNAVAIL           0.00     0.00     0.00     0.00

PSET/PART P-LCK NEGOTIATION          38.00     0.02     0.00     0.00
PAGE P-LOCK NEGOTIATION            3026.00     1.26     0.03     0.01
OTHER P-LOCK NEGOTIATION           1520.00     0.63     0.01     0.00
P-LOCK CHANGE DURING NEG.          3072.00     1.28     0.03     0.01
```

Figure 33. Data sharing locking block of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics trace

For this example statistics trace, calculate global contention as follows:

Determine the total number of suspends that are caused by contention (H + I + J).

```
3236 + 334 + 863 = 4433
```

Determine the total number of requests that went to XES (C + D + E + F + G + H + I + J).

```
1478800 + 45450 + 810000 + 0 + 108 + 3236 + 334 + 863 = 2338791
```

Divide the total number of suspends by the total number of requests.

```
4433 / 2338791 = .0019
```

Multiply the result by 100.

```
.0019 x 100 = 0.19
```

The result indicates that the global contention rate is approximately 0.19%(A). Because this is such a low rate of contention, it is probably not necessary to determine the amount of false contention.

What to do next

If the global contention rate is greater than 5%, investigate whether the following types of contention are contributing to the global contention:

IRLM contention
Lock resource contention. Use the lock trace to identify the objects. Some object definitions might need to be changed. For example, row-level locking might be needed for random access.

XES contention
XES understands only S or X locks. This is usually not an problem

False contention

If the size of the lock facility coupling size is too small, you might need to increase the size of the coupling facility lock table.

Related concepts

Statistics trace (Db2 Performance)

Avoiding false contention

The Db2 coupling facility lock structure has two parts: a lock table, which is used to determine whether inter-Db2 read/write interest exists on a particular resource, and a list of the update locks that are currently held.

Related tasks

Controlling the Db2 trace (Db2 Administration Guide)

Calculating false contention percentages with the Db2 statistics trace

You can use the Db2 statistics trace to calculate false contention percentages.

Related reference

Statistics Report and Trace Blocks (Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS)

Related information

Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS

Calculating false contention percentages with the Db2 statistics trace

You can use the Db2 statistics trace to calculate false contention percentages.

Procedure

To calculate false contention with the Db2 statistics trace:

1. Determine the number of suspends that are caused by false contention.
2. Determine the total number of requests that went to XES.
3. Divide the number of suspends that are caused by false contention by the total number of requests that went to XES.
4. Multiply the result by 100.

Example

```
DATA SHARING LOCKING            QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------      --------  -------  -------  -------
GLOBAL CONTENTION RATE (%)          0.19  A
FALSE CONTENTION RATE (%)           0.04  B
P/L-LOCKS XES RATE (%)             36.47

LOCK REQUESTS (P-LOCKS)            107.4K    44.74     0.93     0.18
UNLOCK REQUESTS (P-LOCKS)          105.2K    43.82     0.91     0.18
CHANGE REQUESTS (P-LOCKS)         1453.00     0.61     0.01     0.00

SYNCH.XES - LOCK REQUESTS    C     1478.8K   616.16    12.77     2.48
SYNCH.XES - CHANGE REQUESTS  D    45450.00    18.94     0.39     0.08
SYNCH.XES - UNLOCK REQUESTS  E     810.0K   337.48     7.00     1.36
BACKGROUND.XES -CHILD LOCKS  F       0.00     0.00     0.00     0.00
ASYNCH.XES -CONVERTED LOCKS  G     108.00     0.04     0.00     0.00

SUSPENDS - IRLM GLOBAL CONT  H    3236.00     1.35     0.03     0.01
SUSPENDS - XES GLOBAL CONT.  I     334.00     0.14     0.00     0.00
SUSPENDS - FALSE CONT. MBR   J     863.00     0.36     0.01     0.00
SUSPENDS - FALSE CONT. LPAR  J        N/A      N/A      N/A      N/A
REJECTED - XES                    31897.00    13.29     0.28     0.05
INCOMPATIBLE RETAINED LOCK            0.00     0.00     0.00     0.00
NOTIFY MESSAGES SENT                344.00     0.14     0.00     0.00
NOTIFY MESSAGES RECEIVED            106.00     0.04     0.00     0.00
P-LOCK/NOTIFY EXITS ENGINES         500.00      N/A      N/A      N/A
P-LCK/NFY EX.ENGINE UNAVAIL           0.00     0.00     0.00     0.00

PSET/PART P-LCK NEGOTIATION          38.00     0.02     0.00     0.00
PAGE P-LOCK NEGOTIATION            3026.00     1.26     0.03     0.01
OTHER P-LOCK NEGOTIATION           1520.00     0.63     0.01     0.00
P-LOCK CHANGE DURING NEG.          3072.00     1.28     0.03     0.01
```

Figure 34. Data sharing locking block of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics trace

For this example statistics trace, calculate the false contention as follows:

Determine the total number of suspends that are caused by false contention (**J**).

```
863
```

Determine the total number of requests that went to XES (**C** + **D** + **E** + **F** + **G** + **H** + **I** + **J**).

```
1478800 + 45450 + 810000 + 0 + 108 + 3236 + 334 + 863 = 2338791
```

Divide the number of suspends that are caused by false contention by the total number of requests that went to XES.

```
863 / 2338791 = .0004
```

Multiply the result by 100.

```
.0004 × 100 = .04
```

False contention is approximately 0.04% of the total number of requests that went to XES.

Related concepts

Statistics trace (Db2 Performance)

Avoiding false contention

The Db2 coupling facility lock structure has two parts: a lock table, which is used to determine whether inter-Db2 read/write interest exists on a particular resource, and a list of the update locks that are currently held.

Related tasks

Controlling the Db2 trace (Db2 Administration Guide)

Calculating global contention percentages with the Db2 statistics trace

You can use the Db2 statistics trace to calculate global contention percentages for lock requests.

Related reference

Statistics Report and Trace Blocks (Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS)

Related information

Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS

Lock monitoring with the Db2 accounting trace

Use the Db2 accounting trace to determine which users or plans are experiencing global lock contention.

The accounting trace provides a summary of thread resource usage within Db2. Db2 threads experiencing global lock contention are shown in accounting trace class 1, as shown in the figure below. The accumulated elapsed time of the suspensions are shown in accounting trace class 3.

```
LOCKING              AVERAGE     TOTAL      DATA SHARING          AVERAGE     TOTAL
--------------       --------  --------     -------------------  --------  --------
TIMEOUTS                0.00         3      GLOBAL CONT RATE(%)      0.54       N/A  B
DEADLOCKS               0.00         0      FALSE CONT RATE(%)       0.00       N/A  C
ESCAL.(SHARED)          0.00         0      P/L-LOCKS XES (%)       17.97       N/A  D
ESCAL.(EXCLUS)          0.00         2      LOCK REQ   - PLOCKS      2.92   8317127  E
MAX PG/ROW LOCKS HELD   0.92      5053      UNLOCK REQ - PLOCKS      2.76   7849772
LOCK REQUEST           92.74   264098K  A   CHANGE REQ - PLOCKS      0.01     20661
UNLOCK REQUEST          7.78  22155610      LOCK REQ   - XES        17.19  48941369  F
QUERY REQUEST           0.00         0      UNLOCK REQ - XES         8.03  22863352  G
CHANGE REQUEST          0.42   1196297      CHANGE REQ - XES         0.26    754626  H
OTHER REQUEST           0.00         0      SUSPENDS   - IRLM        0.14    396784  I
TOTAL SUSPENSIONS      16.78  47784696      SUSPENDS   - XES         0.00         0  J
 LOCK SUSPENSIONS       0.00     10897      CONVERSIONS- XES         0.05    128351  K
 IRLM LATCH SUSPENS.   16.78  47773728      FALSE CONTENTIONS        0.00         0  L
 OTHER SUSPENS.         0.00        71      INCOMPATIBLE LOCKS       0.00         0
                                            NOTIFY MSGS SENT         0.00        73
```

Figure 35. Portion of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS accounting trace showing locking activity

The following formulas are used to calculate the global contention rate, false contention rate, and percentage of P/L-lock requests that were synchronously propagated to XES:

GLOBAL CONT RATE(%) = (I+J+L) / (F+G+H+I+J+K+L)
FALSE CONT RATE(%) = (K+L) / (F+G+H+I+J+K+L)
P/L-LOCKS XES(%) = F / (A+E)

Lock monitoring with the Db2 performance trace

The Db2 performance trace gives more detailed information about which shared resources are experiencing contention.

Performance traces are generally activated on an as-needed basis because of their added overhead. Performance trace class 6 (specifically, IFCID 0045) indicates whether suspension is caused by contention.

This trace causes Db2 to write a trace record every time a lock is suspended and every time it is resumed. Part of the data that is recorded is the resource name that is experiencing the contention. By determining which shared resources are experiencing the lock contention, you might be able to make some design changes or other tuning actions to increase concurrency. Check for contention within IRLM, because IRLM indicates true contention over resources.

Changing the size of the lock structure

You can change the size of a lock structure dynamically or by rebuilding the structure after making a CFRM policy change.

When choosing which way to change the size, consider the level of the coupling facility and when you want the storage of the lock table portion of the lock structure changed.

When you change the size of the lock structure dynamically, only the modify lock list portion of the structure is changed immediately. The size of the lock table portion remains unchanged until the structure is rebuilt. When the structure is rebuilt, the structure is reallocated at the changed size with the

storage divided between the lock table and the record table based on your IRLMPROC LTE value or the value that is set using the MODIFY irlmproc,SET,LTE= command, if either of these values is nonzero. (If the structure was forced with the SETXCF FORCE command, it is reallocated at the INITSIZE that is specified in the CFRM policy and not the changed size.)

Related concepts

Coupling facility structure size allocation (Db2 Installation and Migration)

Changing the lock structure size dynamically

If certain conditions are met, you can dynamically change the size of a lock structure.

About this task

You can dynamically change the lock structure size if all the following conditions are true:

- The lock structure is allocated in a coupling facility with CFLEVEL greater than zero.
- The currently allocated size of the structure is less than the maximum size that is defined in the SIZE parameter of the CFRM policy.

Procedure

Issue the SETXCF START,ALTER command.

Example

For example, use the following command to change the lock structure size for group DSNDB0A:

```
SETXCF START,ALTER,STRNAME=DSNDB0A_LOCK1,SIZE=newsize
```

This example assumes that *newsize* is less than or equal to the maximum size that is defined in the CFRM policy for the lock structure. If the maximum size (SIZE in the CFRM policy) is still not big enough, you must increase the lock storage in the CFRM policy and rebuild the lock structure.

Important: For a duplexed lock structure, you are changing the size of both the primary and secondary structure with a single command.

Because the dynamic method affects only the record table entries portion of the lock structure, the impact of changing the lock size can have a disproportionately large effect on the record table list portion of the structure. For example, if you halve the size of the lock structure, it can result in all of the available record table entries being taken away—probably not the result you want. For the same reason, if you double the lock structure size, the increased storage space is used entirely by the record table unless a rebuild occurs or the group is shut down, the structure is forced, and the group is restarted. If either of these are done after the size is changed, the split of the new structure is determined by the number of lock table entries requested by the first IRLM to connect to the group.

Related reference

z/OS SETXCF START command (MVS System Commands)

Changing the size of the lock structure by rebuilding

One way to change the size of a lock structure is to rebuild the structure.

About this task

If any of the following conditions are true, you must rebuild the lock structure in order to change its size:

- The lock structure is allocated in a coupling facility at CFLEVEL=0.
- The allocated size of the structure is already at the maximum size defined by the SIZE parameter of the CFRM policy, and you need to increase the maximum limit.
- You want to change the size of the lock table portion of the lock structure.

Important: For duplexed lock structures, you change the size of both the primary and secondary structure with a single command.

Procedure

To change the lock structure size through rebuild, complete at least one of the following procedures:

- If you are at maximum size or want to increase the maximum size that is available, change the CFRM POLICY SIZE to the size that you want.
  - For information on increasing the maximum size that is defined by the SIZE parameter of the CFRM policy, see "Changing the size of the group buffer pool" on page 190.
  - If you are satisfied with the number of lock table entries that you have been getting, leave the INITSIZE the same. If you want more lock table entries to decrease your contention rate, change the INITSIZE to accommodate the added number of entries, by taking either of the following actions:
    - If you are allowing IRLM to determine how many LTE entries to request, ensure that when you change INITSIZE, it remains a power of 2 so that there is a 1:1 split between the lock table storage and the record table storage. For example, if your current INITSIZE is 16 MB, increase it to 32 MB.
    - If you are controlling the number of lock table entries by specifying LTE= in the IRLMPROC or by issuing the MODIFY irlmproc,SET,LTE= command, your INITSIZE does not need to be a power of 2, but it is still recommended. You do need to ensure that it is large enough to accommodate the storage required for your LTE= value and still large enough to create sufficient record table entries to handle your update volume.
- If you want to change the size of the lock table:

  a) If your contention rates are low and you want **fewer** lock table entries, complete the following procedure:

     a. If you are letting IRLM control the number of lock entries to request, you must decrease the INITSIZE by a power of 2.

        Important: This also decreases the record table size by half.

     b. If you want to control the number of lock table entries to request, issue the MODIFY irlmproc,SET,LTE= command, where the LTE= value reduces the current number of lock entries by half. For example, assume that you currently have 16 MB lock entries, specify:

        ```
        MODIFY irlmproc,SET,LTE=8
        ```

        This method increases the number of record table entries, because the INITSIZE is not altered.

  b) If your contention rates are high and you want **more** lock table entries, complete the following procedure:

     a. If you are letting IRLM control the number of lock entries to request, you should increase the INITSIZE by a power of 2. This also doubles the size of the record table.

     b. If you want to control the number of lock table entries to request, issue the MODIFY irlmproc,SET,LTE= command, where the LTE= value increases the current number of lock entries by powers of 2. For example, assume that you currently have 8 MB of lock entries, specify:

        ```
        MODIFY irlmproc,SET,LTE=16
        ```

        Important: This method decreases the number of record table entries if the INITSIZE is not altered.

        If the INITSIZE is not large enough to provide the lock table storage and adequate record table storage, you will experience failures when trying to write RLE to the coupling facility. If you do not want the record table size to change, you must increase the INITSIZE by the same amount

of storage that will be used by the additional lock table entries. For example, assume that you currently have 8 MB of lock table entries and you issue the following command:

```
MODIFY irlmproc,SET,LTE=16
```

With an INITSIZE of 32 MB, you will not have any storage left for record table entries. Determine the additional storage needed by using the following formula:

```
(lock table entries (new) - lock table entries (old)) x 2 bytes =
additional storage for lock entries
```

In this example:

```
(16MB - 8MB) x 2 bytes = 16MB
```

So adding 16 MB to the original INITSIZE of 32 MB will result in the new INITSIZE of 48 MB.

Tuning group buffer pools

With Db2 data sharing, group buffer pools are a key component to ensuring that Db2 does not read down-level data in its member pools. Physical locks (P-locks) are also used in that process. Your understanding of these processes is helpful when tuning the data sharing group for best performance.

Db2 does the following in a data sharing environment:

- Ensures that Db2 does not read down-level data that is cached in its member buffer pools (*cache coherency*).
- Enables, as much as possible, a quick refresh of a down-level page without having to go to disk.

With Db2 data sharing, a database page can reside:

- In a local buffer pool
- In a group buffer pool
- On disk

Database pages continue to be cached in each member's buffer pools before they can be referenced or updated. Each sharing member can control its own buffer pool configurations (the size and number of buffer pools). However, if inter-Db2 read/write interest exists in the data, the group buffer pool is also used for caching data (unless the group buffer pool is defined as GBPCACHE (NO) or the page set is defined with GBPCACHE NONE).

The group buffer pool contains information necessary for maintaining cache coherency. Pages of GBP-dependent page sets are registered in the group buffer pool. When a changed page is written to the group buffer pool, all members that have this page cached in their buffer pools are notified that the page has been invalidated (this notification does not cause a processing interruption on those systems). This is called *cross-invalidation*. When a member needs a page of data and finds it in its buffer pool, it tests to see if the buffer contents are still valid. If not, then the page must be refreshed, either from the group buffer pool or disk.

Related information

Db2 11 for z/OS: Buffer Pool Monitoring and Tuning (IBM Redbooks)

Subsystem and Transaction Monitoring and Tuning with Db2 11 for z/OS (IBM Redbooks)

Assigning page sets to group buffer pools

Any data sharing group can have up to fifty 4 KB page size group buffer pools. Each group can have up to ten each of 8 KB, 16 KB, and 32 KB page size group buffer pools.

About this task

Different group buffer pools can reside in different coupling facilities. The strict naming convention ensures that Db2 can map the group buffer pools to the individual member buffer pools. For example,

buffer pool BP0 maps to group buffer pool GBP0. Thus, your choice of buffer pool determines which group buffer pool is used. GBP0 is the default unless you have specified a different default for user data and indexes on installation panel DSNTIP1.

Procedure

To assign a table space to a group buffer pool:

1. Stop all access to the table space by issuing the STOP DATABASE command.
2. Change the buffer pool assignment by running the ALTER TABLESPACE SQL statement.
3. Allow access to the table space by issuing the START DATABASE command.

Example

The following example procedure shows how to assign table space DSN8S11D to GBP2. The procedure works only for altering a table space to a buffer pool that has the same page size.

1. Stop all access to the table space by issuing the following command:

```
-DB1A STOP DATABASE(DSN8D11A) SPACENAM(DSN8S11D)
```

2. Change the buffer pool assignment by running the following SQL statement:

```
ALTER TABLESPACE DSN8D11A.DSN8S11D
 BUFFERPOOL BP2;
```

3. Allow access to the table space by issuing the following command:

```
-DB1A START DATABASE(DSN8D11A) SPACENAM(DSN8S11D)
```

Related reference

DSNTIP1: Buffer pool sizes panel 1 (Db2 Installation and Migration)
-STOP DATABASE (Db2) (Db2 Commands)
ALTER TABLESPACE (Db2 SQL)
-START DATABASE (Db2) (Db2 Commands)

Recommendations for performance

For best performance, keep GBP-dependent page sets in separate buffer pools from non-GBP-dependent page sets.

For example, keep work file table spaces, which are always non-GBP-dependent, in different buffer pools than those used by GBP-dependent page sets. Assign work file table spaces to a buffer pool other than BP0. This separation helps Db2 more efficiently handle registering pages to, and unregistering pages from, the group buffer pool.

- Create a buffer pool specifically for LOBs to improve the efficiency of read LRSN value.
- Data rows that are too large for a single 4-KB page can use an 8-KB, 16-KB, or 32-KB page to improve the balance for different processing requirements. This allows you to store larger rows more efficiently and improve performance.

How to keep data from being shared

It is possible, although not necessarily recommended, to restrict access to data to a single member.

If you choose to do this, consider the following operational issues:

- You cannot do workload balancing for that data, because the other members of the group are not aware of that data. Thus, the member that has access to the data can become overloaded if access to that data increases over time.
- Availability is compromised, because if the member that owns the data goes down, no other member can access that data.

- You might have to set up special affinities to allow the application access to that data. Work cannot be automatically routed around the group to find the data.

How to define private data

If you want access to a table space named NOSHARE limited only to DB2C, you could assign NOSHARE to a previously unused buffer pool, such as BP25, using the ALTER TABLESPACE statement.

Do not define a group buffer pool that corresponds to BP25, and assign BP25 a size of zero on any other member of the group. This prevents the other members of the group from attempting to use this buffer pool and therefore prevents the other members from accessing table space NOSHARE.

Inter-Db2 interest and GBP-dependency

For Db2 data sharing environments, the concepts of inter-Db2 read/write interest and group buffer pool dependency (GBP-dependency) are closely related.

Whenever inter-Db2 read/write interest exists on a page set or partition, that object is GBP-dependent. Conversely, if no inter-Db2 read/write interest exists on a page set or partition, the object is usually not GBP-dependent. Sometimes an object still has pages cached in the group buffer pool, and it can remain GBP-dependent even after the inter-Db2 read/write interest has gone away. The following table shows you how to determine if a page set is GBP-dependent based on the inter-Db2 interest of one member and all other members.

Table 29. Determining group buffer pool dependency

One member's interest	Other members' interest	Is page set GBP-dependent?
Read-only	None, Read-only	No
Read-only	Read/Write	Yes
Read/Write	None	No [1]
Read/Write	Read-only	Yes
Read/Write	Read/Write	Yes

Exception:

1. The page set remains GBP-dependent for some time before Db2 removes the dependency. Db2 might not be able to remove the GBP-dependency if applications update the page set without issuing periodic commits.

How Db2 tracks interest

The mechanism that Db2 uses to express interest in an object is a global lock called a *physical lock* (P-lock).

These locks are "physical" in contrast to transaction locks, sometimes called "logical locks" (L-locks). The level of P-lock that tracks Db2 read/write interest is a *page set P-lock*. There are also page P-locks, which serve another role.

Although you do not have as much control over physical locks as over transaction locks, P-locks play an important part in how Db2 tracks inter-Db2 interest.

Page set P-lock operations occur on each member, and reflect that member's level of interest in a page set. Even if only one data sharing member is active, page set P-lock operations still occur. The following table shows when those operations occur on each member.

Table 30. When page set P-lock operations occur on each member

Event	Page set P-lock operation
Page set or partition data sets are physically opened.	Page set P-lock is obtained in a read-only state.

Table 30. When page set P-lock operations occur on each member (continued)

Event	Page set P-lock operation
Page set or partition is first updated.	Page set P-lock is changed to a read-write state.
No update was done within an installation-specified time period or number of checkpoints (read-only switching).	Page set P-lock is changed to a read-only state.
Page set or partition data sets are closed.	Page set P-lock is released.

In addition to the events mentioned in the table above, a special case can occur under the following conditions:

- A single member with read/write interest and any number of members with read-only interest exist.
- All members have read-only interest and the page set or partition has been GBP-dependent since the time it was physically opened.

When those conditions occur and the page set is defined with CLOSE YES, Db2 physically closes the page set or partition to remove the inter-Db2 read/write interest for any read-only members that do not reference the page set or partition again in a certain amount of time. The amount of time is at least the value of the PCLOSET subsystem parameter, but it can be up to twice the value of PCLOSET.

Related concepts

Physical locks in data sharing
A physical lock (P-lock) is a type of page set lock or page lock that is specific to data sharing.

Scenarios of P-Lock operations
Looking at a scenario can help you understand how P-Lock operations work.

Scenarios of P-Lock operations

Looking at a scenario can help you understand how P-Lock operations work.

The following figure shows a typical sequence of events for P-locking and P-lock negotiations between two members of a data sharing group.

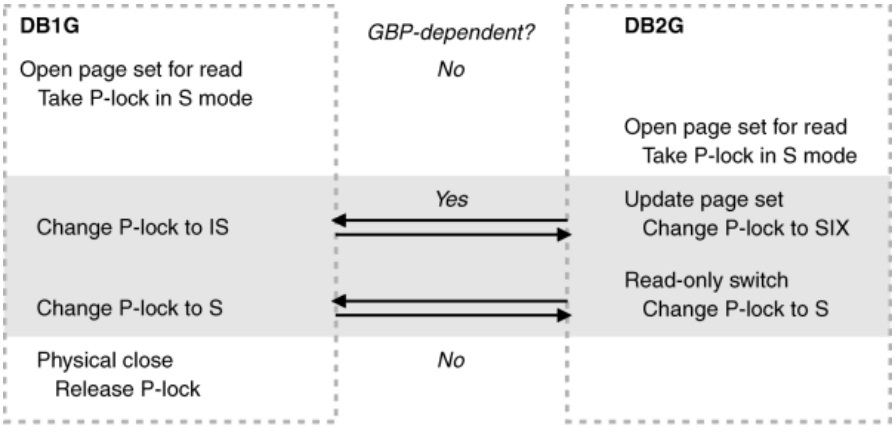


Figure 36. P-lock operations between two members

The following figure shows what happens when a single updater remains.

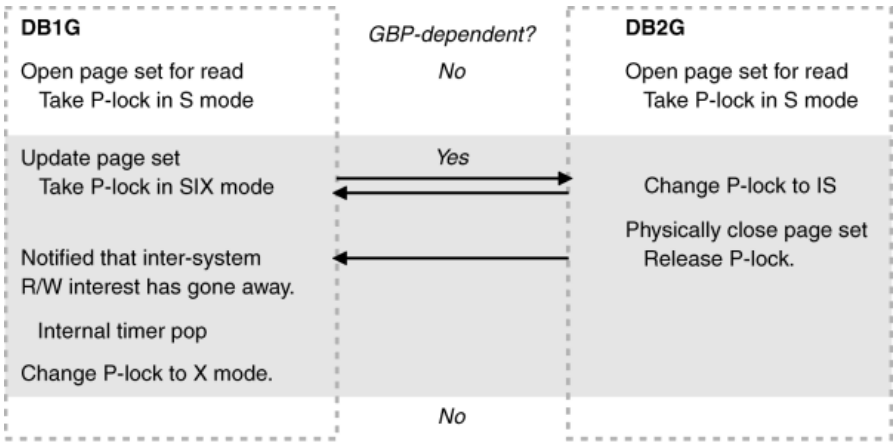


Figure 37. P-lock operations when single updater remains

Tuning recommendation to prevent Db2 from frequently going in and out of GBP-dependency
To avoid the need for Db2 to frequently switch in and out of GBP-dependency, tune the subsystem parameters that affect when data sets are switched to a different state.

These parameters can be set through the following fields on installation panel DSNTIPL1:

- CHECKPOINT TYPE
- RECORDS/CHECKPOINT
- MINUTES/CHECKPOINT
- RO SWITCH CHKPTS
- RO SWITCH TIME

Note: NOT LOGGED table spaces change their GBP-dependency more often than LOGGED table spaces, because for NOT LOGGED table spaces, RO SWITCH CHKPTS and RO SWITCH TIME are both set to one. If you run multiple jobs consecutively that modify a NOT LOGGED table space, consider scheduling them to run within an RO SWITCH TIME of one minute, in order to reduce the number of times NOT LOGGED table spaces change their GBP-dependency.

Related reference
CHECKPOINT TYPE field (CHKTYPE subsystem parameter) (Db2 Installation and Migration)
RECORDS/CHECKPOINT field (CHKFREQ and CHKLOGR subsystem parameters) (Db2 Installation and Migration)
MINUTES/CHECKPOINT field (CHKFREQ and CHKMINS subsystem parameters) (Db2 Installation and Migration)
RO SWITCH CHKPTS field (PCLOSEN subsystem parameter) (Db2 Installation and Migration)
RO SWITCH TIME field (PCLOSET subsystem parameter) (Db2 Installation and Migration)

Determining the amount of inter-system sharing
To determine the amount of inter-system sharing, you can use the DISPLAY BUFFERPOOL command with LIST option to get a snapshot of your objects and how they are being shared across members.

Using this command, you will see, by partition, which members have interest in the object, which level of interest they have (the P-lock state), and which member is the castout owner. You can also use the DBNAME, SPACENAME, GBPDEP, and CASTOWNR options to limit the scope of the report.

Issue the following command:

GUPI

```
-DISPLAY BUFFERPOOL(BP0) LIST(*)
```

GUPI

Your output will be similar to the output that is shown in the figure below.

```
DSNB401I : BUFFERPOOL NAME BP0, BUFFERPOOL ID 0, USE COUNT 21
DSNB402I : VIRTUAL BUFFERPOOL SIZE = 500 BUFFERS
             ALLOCATED       =      500   TO BE DELETED   =        0
             IN-USE/UPDATED  =        0
DSNB406I : VIRTUAL BUFFERPOOL TYPE -
             CURRENT            = PRIMARY
             PENDING            = PRIMARY
           PAGE STEALING METHOD = LRU
DSNB404I : THRESHOLDS -
            VP SEQUENTIAL    = 80
            DEFERRED WRITE   = 85   VERTICAL DEFERRED WRT  = 80,0
            PARALLEL SEQUENTIAL = 50   ASSISTING PARALLEL SEQT=  0
DSNB460I :
-------------PAGE SET/PARTITION LIST INFORMATION--------------
                                ------DATA SHARING INFO------
                           TS GBP  MEMBER  CASTOUT  USE  P-LOCK
DATABASE SPACE NAME  PART IX DEP   NAME    OWNER  COUNT STATE
======== ========== ==== == === ======== ======= ===== ======
DSNDB01  DSNLLX02         IX  Y  DB1AA                0   IS
                              DB1AB        Y       0   SIX
DSN8D61A DSN8S61E     001 TS  Y  DB1AA                0   IS
                              DB1AB        Y       0   SIX
                      002 TS  N  DB1AA                0   S
                              DB1AB                0   S
DSNDB01  DSNSPT01         IX  N  DB1AA                0   S
                              DB1AB                0   S
         SPT01            TS  N  DB1AA                0   S
                              DB1AB                0   S
DSNDB07  DSN4K01          TS  N  DB1AA                0   S
…
DSN9022I : DSNB1CMD '-DIS BPOOL' NORMAL COMPLETION
```

Figure 38. Sample DISPLAY BUFFERPOOL output showing the amount of inter-system sharing.

Displaying GBP-dependent page sets

You can use the DISPLAY DATABASE command to determine if a page set is GBP-dependent.

GUPI To determine if a particular page set is GBP-dependent, use the DISPLAY DATABASE command with the LOCKS option:

```
-DB1A  DISPLAY DATABASE(DSN8D11A) SPACE(DSN8S11D) LOCKS
```

Your output will be similar to the output that is shown in the following figure:

```
 NAME     TYPE PART STATUS             CONNID   CORRID       LOCKINFO
 -------- ---- ---- ------------------ -------- ------------ ----------
 DSN8S11D    TS        RW                                       H-IX,PP,I
                                      MEMBER NAME DB1A (CO)
 DSN8S11D    TS        RW                                       H-IX,PP,I
                                      MEMBER NAME DB2A
```

Figure 39. Sample DISPLAY DATABASE output showing GBP-dependent table spaces

Page set P-locks are identified by a member name rather than a correlation ID. They are also identified by 'PP' as the lock unit. If any of the P-locks shown in the output have a lock state of NSU, SIX, or IX, the identified page set is GBP-dependent. Thus, the output in the figure above shows that DSN8S11D is GBP-dependent. GUPI

Determining GBP-dependency for a particular member

At times you might need to know what the impact is to bring down a particular member or to disconnect a particular member from the group buffer pool.

You can use the DISPLAY BUFFERPOOL command with the GBPDEP(Y) option to discover whether a particular member has any page sets opened that are GBP-dependent:

```
-DB1A DISPLAY BUFFERPOOL(BP0) GBPDEP(Y)
```

Your output will be similar to the output that is shown in the figure below.

```
@DIS BPOOL(BP0) GBPDEP(Y)
DSNB401I @ BUFFERPOOL NAME BP0, BUFFERPOOL ID 0, USE COUNT 21
DSNB402I @ VIRTUAL BUFFERPOOL SIZE = 500 BUFFERS
             ALLOCATED       =      500   TO BE DELETED   =        0
             IN-USE/UPDATED  =        0
DSNB406I @ VIRTUAL BUFFERPOOL TYPE -
             CURRENT            = PRIMARY
             PENDING            = PRIMARY
           PAGE STEALING METHOD = LRU
DSNB404I @ THRESHOLDS -
            VP SEQUENTIAL    = 80
            DEFERRED WRITE   = 85   VERTICAL DEFERRED WRT  = 80,0
            PARALLEL SEQUENTIAL = 50   ASSISTING PARALLEL SEQT=  0
DSNB460I @
-------------PAGE SET/PARTITION LIST INFORMATION--------------
                               ------DATA SHARING INFO------
                          TS GBP  MEMBER  CASTOUT  USE  P-LOCK
DATABASE SPACE NAME  PART IX DEP   NAME    OWNER  COUNT STATE
======== =========== ==== == === ======== ======= ===== ======
DSNDB01  DSNLLX02         IX  Y  DB1AA        Y       0   IX
                              DB1AB                0   IX
DSN8D71A DSN8S71E     001 TS  Y  DB1AA        Y       0   IX
                              DB1AB                0   IX
                      003 TS  Y  DB1AA        Y       0   IX
                              DB1AB                0   IX
DSNDB01  DSNLLX01         IX  Y  DB1AA        Y       0   IX
                              DB1AB                0   IX
         SYSLGRNX         TS  Y  DB1AA        Y       0   IX
                              DB1AB                0   IX
DSN9022I @ DSNB1CMD '-DIS BPOOL' NORMAL COMPLETION
```

Figure 40. Sample DISPLAY BUFFERPOOL output indicating page sets and partitions are group-buffer-pool-dependent

Physical locks in data sharing

A physical lock (P-lock) is a type of page set lock or page lock that is specific to data sharing.

Page set P-Locks

P-locks are used on physical objects stored in buffers (table spaces, index spaces, and partitions).

P-locks have complete partition independence: it is possible to have a P-lock on one partition of a page set and not on another. A P-lock on a page set does not necessarily mean that a P-lock exists on the corresponding index.

P-locks do not control concurrency but they do help Db2 track the level of interest in a particular page set or partition and determine the need for cache coherency controls.

P-locks differ from L-locks (transaction locks) in the following ways:

- P-locks are owned by a member. After a P-lock is obtained for the subsystem, later transactions accessing the locked data do not have to incur the expense of physical locking.
- The mode of a P-lock can be negotiated. If one member requests a P-lock that another member holds in an incompatible mode, the existing P-lock can be made less restrictive. The negotiation process usually involves registering pages or writing pages to the group buffer pool, and then downgrading the P-lock to a mode that is compatible with the new request.

Retained P-locks

Just as with transaction locks, certain P-locks can be retained because of a system failure.

A retained P-lock means that other members cannot access the data that the P-lock is protecting if the accessing member requests a P-lock in an incompatible state. Thus, if a member fails holding an IX page set P-lock, it is still possible for another member to obtain an IX page set P-lock on the data.

Use the DISPLAY DATABASE command with the LOCKS option to determine if retained locks exist on a table space, index, or partition. An "R" in the LOCKINFO column indicates that a lock is retained.

The following table shows the possible modes of access for a page set and the P-lock state that is retained if the member that is represented in the first column fails.

Table 31. Determining retained P-lock state

One member's interest	Other members' interest	Retained P-lock states of single member
Read-only	None, Read-only	None
Read-only	Read/Write	None
Read/Write	None	X or NSU [1]
Read/Write	Read-only	IX [2]
Read/Write	Read/Write	IX

Notes:

1. NSU stands for "non-shared update". It acts like an X lock, but is only used during P-lock negotiation from an X to an SIX.
2. The P-lock is retained in SIX mode if the page set or partition is an index that is not on a Db2 catalog or directory table space.

Related concepts

Active and retained locks for Db2data sharing members

Active locks, retained locks, and retained utility locks can be held by a failed member. Locks that are held by a failed member can affect the availability of data to the other members of the group that are still running Db2 applications.

Page P-locks

At times, a P-lock must be obtained on a page to preserve physical consistency of the data between members. These locks are known as *page P-locks*.

Page P-locks, are used, for example, when two subsystems attempt to update the same page of data and row locking is in effect. These locks are also used for GBP-dependent space map pages and GBP-dependent leaf pages for indexes, regardless of locking level. Page P-locks can also be retained if a member fails.

If an index page set or partition is GBP-dependent, Db2 does not use page P-locks for that index page set or partition if **all** of the following are true:

- Only one member is updating the index page set or partition
- No repeatable read claimers exist on the read-only members for the index page set or partition
- The index is not on a Db2 catalog or directory table space

Because of the possible increase in P-lock activity with row locking, evaluate row locking carefully before using it in a data sharing environment. If you have an update-intensive application process, the amount of page P-lock activity might increase the overhead of data sharing.

To decrease the possible contention on those page P-locks, consider using page locking and a MAXROWS value of one on the table space to simulate row locking. You can get the benefits of row locking without the data page P-lock contention that comes with it. A new MAXROWS value does not take effect until you run REORG on the table space.

P-lock monitoring

Monitoring activity of P-locks, especially page P-locks, in a Db2 data sharing environment can help you determine if you need to control inter-Db2 read/write interest.

More overhead exists when inter-Db2 read/write interest exists in a page set. Although Db2 dynamically tracks inter-Db2 read/write interest, which helps avoid data sharing overhead when it is not needed, you will pay some cost for the advantages of data sharing.

If excessive global contention exists that cannot be resolved by any tuning measures, you might need to reduce the locking overhead by keeping some transactions and data together on a single system.

How to find information about page set P-locks

You can use the DISPLAY DATABASE command with the LOCKS option to find information about page set P-locks, including which member is holding or waiting for P-locks, and whether P-locks are being held because of a Db2 failure.

The following figure has sample output obtained from the command.Figure 39 on page 159

```
NAME     TYPE PART STATUS             CONNID   CORRID       LOCKINFO
-------- ---- ---- ------------------ -------- ------------ ----------
DSN8S11D    TS        RW                                       H-IX,PP,I
                                      MEMBER NAME DB1A (CO)
DSN8S11D    TS        RW                                       H-IX,PP,I
                                      MEMBER NAME DB2A
```

Figure 41. Sample DISPLAY DATABASE output showing GBP-dependent table spaces

A "PP" in the LOCKINFO field of the output indicates that a particular lock is a page set or partition P-lock.

You can also obtain information about P-locks, along with information about transaction locking, from the statistics and accounting traces. Performance class 20 (IFCID 0251) also contains information about P-lock negotiation requests. IFCID 0251 is mapped by DSNDQW04.

How to find information about page P-locks

Page P-locking activity is recorded, along with the rest of the data sharing locking information, in the statistics and accounting trace classes.

You can find more detail about those page P-locks in performance trace class 21 (IFCID 0259). IFCID 0259 allows you to monitor page P-locking without having to use a full Db2 lock trace. IFCID 0259 is mapped by DSNDQW04.

Options for reducing space map page contention

When you define table spaces and indexes, you have several options to help reduce space map hot spots. These hot spots can occur when a large amount of update, insert, or delete activity occurs on a page set from multiple members of a group.

The MEMBER CLUSTER option can reduce contention when doing heavy sequential inserts, and the TRACKMOD NO option can reduce contention when any type of insert, update, or delete activity occurs. The TRACKMOD NO option cannot be used when incremental image copies are used for a tablespace.

Understand the implications of each option before choosing either one, as both options have drawbacks.

Related concepts

Options that affect data sharing performance
The GBPCACHE, MEMBER CLUSTER, and TRACKMOD options all affect the performance of data sharing.

Member affinity clustering
For applications that do heavy sequential insert processing from multiple members, the contention on the space map or for the data pages at the end of the table can be considerable.

Related reference

CREATE TABLESPACE (Db2 SQL)

Member affinity clustering

For applications that do heavy sequential insert processing from multiple members, the contention on the space map or for the data pages at the end of the table can be considerable.

The MEMBER CLUSTER option of CREATE TABLESPACE causes Db2 to manage space for inserts on a member-by-member basis instead of by using one centralized space map. Table spaces defined with MEMBER CLUSTER have the following characteristics:

- Data that is inserted by the SQL INSERT statement is not clustered by the implicit clustering index (the first index) or the explicit clustering index.
- Db2 chooses where to locate the data in such a way that avoids lock and latch contention. In general, it tries to insert data in a place that is covered by the locally cached space map page. If it cannot find space there, it continues to search through space map pages until it can find a place for which the space

map page is available. As a result, space in a data set might not be fully used. However, when the data set reaches the maximum number of extents, lock contention might increase, and Db2 does use the entire space.

- For a UTS table space, the space map page has 10 segments, regardless of the page size. The number of data pages that are allocated to each Db2 member varies, depending on SEGSIZE. A large SEGSIZE value causes more data pages to be covered by the space map page, which in turn causes the table space to grow in a multiple-member data sharing environment. If SEGSIZE is small, more space map pages are required.

  For a partitioned (non-UTS) table space, each space map covers 199 data pages. Because there are more space map pages and some might be partially used, table spaces that are defined with MEMBER CLUSTER can use more disk space.
- To reduce the overhead of reacquiring page P-locks, a page P-lock is held longer for MEMBER CLUSTER table spaces.

The downside to using MEMBER CLUSTER is that data is not inserted in clustering order. If you have a query application that performs best when data is in clustering order, you should run the REORG utility on the table space before starting the query application.

Related concepts

Options that affect data sharing performance
The GBPCACHE, MEMBER CLUSTER, and TRACKMOD options all affect the performance of data sharing.

Related tasks

Altering table spaces (Db2 Administration Guide)

Related reference

REORG TABLESPACE (Db2 Utilities)

ALTER TABLESPACE (Db2 SQL)

CREATE TABLESPACE (Db2 SQL)

Indexes with randomized key columns to reduce hot spots

You can randomize the index key columns in database tables where you observe hot spots within the indexes during INSERT, DELETE, or UPDATE processing by using the new RANDOM option on the CREATE INDEX statement.

This option can be used when you want to alter an index to add a key column that is randomized. Note that even though the indexes are stored in random order, it is still possible to get index only access. Also, when the index is in random order it is only usable for equality lookups.

Example: A user defines an index on the EMPNO column of the EMP table in ASCENDING order as follows:

```
CREATE INDEX DSN8B10.XEMP3
                   ON DSN8B10.EMP
                       (EMPNO    ASC);
```

The user then continually inserts an ascending sequence of values into the EMPNO column (000010, 000020, 000030, ...). These values are always added to the end of the index, creating a hotspot at the end of the index. In this particular index, the user only wants to look up specific employees by their EMPNO (i.e. only equality predicates are applied to the EMPNO column). To reduce contention, the user changes the definition of the index to the following:

```
CREATE INDEX DSN8B10.XEMP3
                   ON DSN8B10.EMP
                       (EMPNO    RANDOM);
```

With the RANDOM specification, the EMPNO values will now be spread randomly throughout the index, preventing the contention that results from always inserting values at the end of the index. Since the common use of this index is for looking up specific employees by their EMPNO, and the user wants to avoid a hotspot at the end, the new RANDOM ordering option provides a good solution.

Control of tracking updates to reduce coupling facility overhead

The TRACKMOD NO option of CREATE or ALTER TABLESPACE can reduce coupling facility overhead caused by constant updating of the space map pages of the page sets.

With TRACKMOD NO, Db2 does not keep track of changed pages in the space map page of the page set. By choosing TRACKMOD NO and not tracking updates, less coupling facility overhead exists, but the cost of incremental image copies is much higher because Db2 must use a table space scan to read all pages to determine if the page has been changed and thus needs to be copied.

If longer copy times means you must take fewer incremental copies, monitor your active log data sets to ensure that you do not have to go to a tape archive data set to do recovery. You might need to make the active log data sets larger, specify more active log data sets, or archive to disk to avoid this possibility.

Recommendation: If you rarely or never use incremental image copies, or if you always use DFSMS concurrent copy with Db2 LOGONLY recovery, use TRACKMOD NO.

Read operations

In a data sharing environment, the process of reading data is different from the process in a non-data sharing environment.

Where Db2 looks for a page

Db2 searches for pages in buffer pools and on disk in a specific order.

Db2 searches for pages in this order:

1. In the local buffer pool. If the page is invalid, Db2 refreshes the page from the group buffer pool (or disk).
2. In the group buffer pool. Db2 checks the group buffer pool for a page if the page set is defined as GBPCACHE ALL or if the page set or the partition is GBP-dependent, unless one of the following conditions is true:
   - Group buffer pool is defined as GBPCACHE(NO).
   - Page set is defined as GBPCACHE NONE.
   - Page set is defined as GBPCACHE SYSTEM and the page being read is not a space map page.
3. On disk. If the page is not in the group buffer pool, Db2 refreshes the page in the buffer pool from disk.

For duplexed group buffer pools, read activity occurs only against the primary structure.

Testing the page validity

Part of the process of controlling cache coherency is testing to see if a page that is referenced in the buffer pool must be refreshed from the group buffer pool or disk because it might no longer be the most current version of the data.

This process is known as testing the page *validity*. Because Db2 tracks the level of interest in a page set across the group, it is not always necessary to make this test. The following table indicates when this test is performed.

For duplexed group buffer pools, only the primary structure is used for cross-invalidations.

Table 32. Determining when page validity must be tested

One member's interest	Other members' interest	Test page validity?
Read-only	None, Read-only	No
Read-only	Read/Write	Yes
Read/Write	None	No
Read/Write	Read-only	No
Read/Write	Read/Write	Yes

Prefetch processing

Db2 prefetch processing for GBP-dependent page sets and partitions varies depending on the level of coupling facility (CFLEVEL) in which the group buffer pool is allocated.

If the group buffer pool is allocated in a coupling facility with CFLEVEL=0 or 1, Db2 reads and registers one page at a time in the group buffer pool.

If the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or higher, Db2 can register the entire list of pages that are being prefetched with one request to the coupling facility. This can be used for sequential prefetch (including sequential detection) and list prefetch. On the list, Db2 does not include any valid pages that are found in the local buffer pool.

For those pages that are cached as "changed" in the group buffer pool, or those that are locked for castout, Db2 still retrieves the changed page from the group buffer pool one at a time. For large, sequential queries, changed pages most likely do not exist in the group buffer pool.

For pages that are cached as "clean" in the group buffer pool, Db2 can get the pages from the group buffer pool (one page at a time), or can include the pages in the disk read I/O request, depending on which is most efficient.

Related concepts

What to look for in a OMEGAMON statistics report
You can use Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics reports to understand group buffer pool activity.

Related reference

-DISPLAY GROUPBUFFERPOOL (Db2) (Db2 Commands)

Related information

DSNB789I (Db2 Messages)

Use of caching for group buffer pools

You can cache pages in the group buffer pool as they are read into a member's local buffer pool by specifying GBPCACHE ALL when you create or alter a table space or index.

When you choose ALL, pages are copied to the group buffer pool as they are read in from disk, even if no inter-Db2 read/write interest exists in those pages.

However, when only a single member has exclusive read/write interest in the page set (that is, only one member has the page set open for update), pages are not cached in the group buffer pool when they are read in from disk. As soon as another member in the data sharing group shows interest in the page set or partition, it becomes GBP-dependent. Changed pages are moved into the group buffer pool, and all pages read in from disk are cached in the group buffer pool.

Choosing GBPCACHE ALL does not prevent Db2 from continuing to cache changed pages in the group buffer pool before writing them to disk (the function provided by the default, GBPCACHE CHANGED).

Note: For LOB table spaces, use the default, GBPCACHE CHANGED. If you use GBPCACHE CHANGED for a LOB table space that is defined with LOG NO and the coupling facility fails, the LOB table space is placed in GRECP. When group buffer pool recovery occurs, all LOB values that were in the coupling facility at the time of the failure are marked invalid because the log records that are necessary to perform the recovery for those values are missing due to the LOG NO attribute.

Example: Use the GBPCACHE clause to cache read-only page sets in the group buffer pool:

GUPI

```
ALTER TABLESPACE DSN8D11A.DSN8S11D
  GBPCACHE ALL;
```

GUPI

Why choose GBPCACHE ALL?

By choosing GBPCACHE ALL, you prevent multiple members from reading the same page in from disk. For this reason, page sets or partitions that typically have a high degree of inter-Db2 read interest are good candidates for GBPCACHE ALL.

If you use the GBPCACHE ALL option, it increases the need for coupling facility resources: processing power, storage, and channel utilization.

Related reference

ALTER TABLESPACE (Db2 SQL)

CREATE TABLESPACE (Db2 SQL)

Write operations

With data sharing, Db2 usually writes changed data to the group buffer pool before writing that changed data to disk.

How the GBPCACHE option affects write operations

You can use several options in a CREATE TABLESPACE statement to tell Db2 how you want data to be handled through the group buffer pool.

GBPCACHE CHANGED or ALL

GBPCACHE CHANGED is the default in Db2. For both ALL and CHANGED, the write operations are the same for changed pages: changed pages are written to the group buffer pool before being written to disk.

GBPCACHE SYSTEM

This option is allowed only for LOBs. For GBPCACHE SYSTEM page sets, the only pages that are written to the group buffer pool are LOB space map pages. All other data pages are written directly to disk, similar to GBPCACHE NONE page sets.

Recommendation: In a data sharing environment, if GBPCACHE CHANGED or GBPCACHE ALL is used instead for a LOG NO LOB table space, a coupling facility failure can result in the table space being marked GRECP. Any kind of recovery of the table space marks the attached LOB values as invalid and places the table space in the AUXW state. For LOB table spaces, choose GBPCACHE SYSTEM to avoid having large LOB values overwhelm the group buffer pool. Also, for LOB table spaces with the LOG NO attribute, GBPCACHE SYSTEM ensures that LOB values are written to disk by commit time, thereby avoiding possible recovery problems caused by missing log data.

GBPCACHE NONE

For GBPCACHE NONE page sets, or for page sets defined in a group buffer pool as GBPCACHE(NO), no pages are written to the group buffer pool. The group buffer pool is used only for the purpose of buffer cross-invalidation. At every COMMIT, any pages that were updated by the transaction and have not yet been written are synchronously written to disk during commit processing. This can have a severe impact on performance for most types of transactions.

One potential advantage of not caching pages in the group buffer pool is that data does not have to be recovered from the log if the coupling facility fails. However, because Db2 still depends on the cross-invalidation information that is stored in the group buffer pool, a coupling facility failure still means some data might not be available. That is why specifying an alternate coupling facility in the CFRM policy is recommended. If you are looking for a high availability option, consider duplexing the group buffer pool rather than suffering the performance hit of writing directly to disk at every COMMIT.

Advantage of specifying GBPCACHE(NO) for group buffer pools: You can specify GBPCACHE(NO) on the group buffer pool level instead of on the page set level. Changing the group buffer pool attribute is usually less disruptive than changing the page set attribute. Because the GBPCACHE(NO) attribute takes precedence over the GBPCACHE option on the page set, you can plan to use different types of processing at different times of day. For example, assume that you want to run transactions or queries during the

day, but you want to do batch updates at night. Assume also that your batch updates would benefit from no group buffer pool data caching. You can do the following:

1. Define the table space as GBPCACHE CHANGED and put it into its own buffer pool.
2. Define the corresponding group buffer pool as GBPCACHE(YES) for daytime processing.
3. At night, use the ALTER GROUPBUFFERPOOL command to change the group buffer pool to GBPCACHE(NO).
4. Set the SETXCF START,REBUILD,STRNAME=*strname* command to enable the new attribute.
5. In the morning, use the ALTER GROUPBUFFERPOOL command to change the group buffer pool back to GBPCACHE(YES).
6. Issue the SETXCF START,REBUILD,STRNAME=*strname* command to enable the new attribute.

Reasons not to cache: A performance benefit is possible for applications in which an updated page is rarely, if ever, referenced again, such as a batch job that sequentially updates a large table. By not caching, you save the costs of transferring data to the group buffer pool and casting out from the group buffer pool. Again, this benefit is at the cost of synchronous disk I/O at COMMIT. To reduce the amount of synchronous disk I/O at COMMIT, you can lower the deferred write thresholds so that Db2 writes more pages asynchronously prior to COMMIT.

To determine if a particular group buffer pool is a candidate for GBPCACHE(NO), look at the group buffer pool statistics. If the ratio of READS, DATA RETURNED / PAGES WRITTEN is less than 1%, this page set might be a good candidate for GBPCACHE(NO), or the page sets using this group buffer pool might be a good candidate for GBPCACHE NONE. You receive the following benefits:

- Reduced coupling facility costs and faster coupling facility response time
- Reduced processor time on the host system
- Better transaction throughput at a small possible cost in higher transaction response time

If you use GBPCACHE NONE, enable DASD Fast Write and set the vertical deferred write threshold (VDWQT) to 0. By setting this threshold to 0, you let deferred writes happen continuously before the COMMIT, thus avoiding a large surge of write activity at the COMMIT.

Related concepts

How Db2 writes to the group buffer pool
With data sharing, Db2 still performs deferred writes for Db2 table spaces, indexes or partitions. However, when an update is to a page set that has inter-Db2 read/write interest, Db2 forces the updated pages to the group buffer pool before or when the transaction commits.

Use of caching for group buffer pools
You can cache pages in the group buffer pool as they are read into a member's local buffer pool by specifying GBPCACHE ALL when you create or alter a table space or index.

How Db2 writes to the group buffer pool

With data sharing, Db2 still performs deferred writes for Db2 table spaces, indexes or partitions. However, when an update is to a page set that has inter-Db2 read/write interest, Db2 forces the updated pages to the group buffer pool before or when the transaction commits.

Updated pages can be written to the group buffer pool before the updating transaction is committed when:

- One of the deferred write thresholds is reached.
- The buffer pool lacks reassignable buffers because writes to the group buffer pool cannot keep up with update activity in the buffer pool. The shortage of buffers can occur when the deferred write thresholds are too high, or if the application is not committing frequently enough—in a data sharing environment, the commits make buffers reassignable.
- An updated page has stayed in the buffer pool for a long period of time since it was last referenced or updated (such as with a long-running transaction that does not issue frequent commits). In this case, a system checkpoint can free space in the buffer pool before the commit.

- The same page is required for update by another system because no conflict on transaction locking exists (such as page sets that are using row locking, index pages, space map pages, and so on). This write is part of the page P-lock negotiation process.

When a page of data is written to the group buffer pool, all copies of that page cached in other members' buffer pools are invalidated. This means that the next time that one of those members needs that page, the page must be refreshed from the group buffer pool (or disk).

Before an updated page is written to the group buffer pool or to disk, Db2 also ensures that the last update log record for that page is externalized to the active log. This ensures that updates can be backed out when necessary.

When committing an updating transaction, Db2 synchronously writes to the group buffer pool pages that were updated but not yet written to the group buffer pool. If a group buffer pool is required and unavailable(because of a channel or hardware failure) at the time the transaction commits, Db2 places all the transaction's updated pages on the logical page list (LPL) associated with each page set. After the problem is fixed, Db2 attempts automatic LPL recovery, but you can issue a START DATABASE command with the SPACENAM option to manually recover the pages on the logical page list.

Note: When a table space or partition is placed in the LPL, at either restart time or during rollback processing, because undo processing is needed for a NOT LOGGED table space, automatic LPL recovery is not initiated and a START DATABASE command identifying this table space will have no effect on its LPL status.

Writing to a GBPCACHE(NO) group buffer pool

For GBP-dependent page sets, no data is written to a group buffer pool for the following instances:

- The group buffer pool is defined as GBPCACHE(NO)
- The page set is defined as GBPCACHE NONE
- The changed pages are non-system pages of a page set defined with GBPCACHE SYSTEM

In these cases, Db2 writes changed pages for the transaction directly to disk at or before COMMIT. Db2 batches up to 32 pages in a single I/O.

When Db2 writes a changed page to disk, it cross-invalidates the page using the group buffer pool. These cross-invalidations are called *explicit cross-invalidations*. These explicit cross-invalidations are reported in statistics separately from cross-invalidations caused by directory reclaims or by writes of a changed page to a group buffer pool.

Writing to a duplexed group buffer pool

When a group buffer pool is duplexed, page writes are performed in the following manner:

1. For some fixed number of pages that must be written:
   a. Each page is written to the secondary group buffer pool asynchronously
   b. Each page is written to the primary group buffer pool synchronously
2. After all pages have been written to the primary group buffer pool, Db2 checks to see if all pages have been written to the secondary group buffer pool. If some pages still need to be written, Db2 forces the completion of those writes.

Related concepts

Database monitoring options
Db2 attempts to automatically recover logical page list page sets.

How Db2 writes from the group buffer pool to disk

The process of writing pages from the group buffer pool to disk is called *castout*.

Because no physical connection exists between the group buffer pool and disk, the castout process involves reading the page from the group buffer pool into a particular member's private buffer (not part of the buffer pool storage), and then writing the page from the private buffer to disk. This member is the

owner of the castout process for the page set or partition. The first member with update intent on the page set or partition is assigned ownership of castout. After castout ownership is assigned, subsequent updating members become backup owners. One of the backup owners becomes the castout owner when the original castout owner no longer has read/write interest in the page set.

Other members can write this page to the group buffer pool even as the page is being cast out. Some events explicitly cause pages to be cast out to disk, such as the STOP DATABASE command.

Castout also occurs when:

- The number of changed pages for a castout class queue exceeds a class threshold value.
- The total number of changed pages for a group buffer pool exceeds a group buffer pool threshold value.
- The group buffer pool checkpoint is triggered.
- No more inter-Db2 read/write interest exists in the page set.
- The group buffer pool is being rebuilt, but the alternate group buffer pool is not large enough to contain the pages from the group buffer that is being rebuilt.

Pages that are cast out as a result of meeting a threshold remain cached in the group buffer pool, and the buffers are available for stealing. Pages that are cast out because no more shared interest exists in the page set are purged from the group buffer pool.

Casting out from a duplexed group buffer pool

Db2 casts out data to disk only from the primary structure. After a set of pages has been cast out, the same set of pages is deleted from the secondary structure. See the DELETE NAME LIST counter in the DISPLAY GROUP BUFFERPOOL MDETAIL report for how many times this event occurs. Db2 ensures that any pages that might have been written to the group buffer pool during castout processing are not deleted from the secondary structure.

Related concepts

Group buffer pool class castout threshold
Each group buffer pool contains a fixed number of castout class queues. This number is an internal value that is set by Db2.

Group buffer pool checkpoint
When a group buffer pool is damaged, all changed data that belong to GBP-dependent page sets must be recovered to the page sets from the Db2 logs.

Displaying the castout owner

You can use the DISPLAY DATABASE command or the DISPLAY BUFFERPOOL command to display the castout owner for a page set or partition.

GUPI Use the DISPLAY DATABASE command with the LOCKS option to display the current castout owner for a given page set:

```
-DB1A    DISPLAY DATABASE(TESTDB) SPACE(*) LOCKS
```

Display the castout owner for a particular page set or partition with (CO) by the member name, as shown in the following figure.

```
⋮
TBS43    TS    01  RW                                        H-SIX,PP,I
    -                  MEMBER NAME DB2A     (CO)
TBS43    TS    02  RW                  BATCH    SELEC        H-IS,P,C
    -                  MEMBER NAME DB1A
⋮
```

Figure 42. Partial DISPLAY DATABASE output showing castout owner for a partition

Use the DISPLAY BUFFERPOOL command with the CASTOWNR keyword to determine for which page sets or partitions the members hold castout ownership:

```
-DIS BUFFERPOOL(BP0) CASTOWNR(Y)
```

The following figure shows output that lists the members that hold the page set or partition P-lock in "U" state.

```
@DIS BPOOL(BP0) CASTOWNR(Y)
DSNB401I @ BUFFERPOOL NAME BP0, BUFFERPOOL ID 0, USE COUNT 40
DSNB402I @ VIRTUAL BUFFERPOOL SIZE = 500 BUFFERS
             ALLOCATED       =      500   TO BE DELETED   =        0
             IN-USE/UPDATED  =        0
DSNB406I @ VIRTUAL BUFFERPOOL TYPE -
             CURRENT            = PRIMARY
             PENDING            = PRIMARY
           PAGE STEALING METHOD = LRU
DSNB404I @ THRESHOLDS -
            VP SEQUENTIAL    = 80
            DEFERRED WRITE   = 85   VERTICAL DEFERRED WRT  = 80,0
            PARALLEL SEQUENTIAL = 50   ASSISTING PARALLEL SEQT=  0
DSNB460I @

-------------PAGE SET/PARTITION LIST INFORMATION--------------
                                ------DATA SHARING INFO------
                           TS GBP  MEMBER  CASTOUT  USE  P-LOCK
DATABASE SPACE NAME  PART IX DEP   NAME    OWNER  COUNT STATE
======== =========== ==== == === ======== ======= ===== ======
DSNDB01  DSNLLX02         IX  Y  V61A        Y        0   IX
                                 V61B                 0   IX
DSN8D71A DSN8S71E     001 TS  Y  V61A        Y        1   SIX
                      003 TS  Y  V61A        Y        1   SIX
DSNDB01  DSNLLX01         IX  Y  V61A        Y        0   IX
                                 V61B                 0   IX
         SYSLGRNX         TS  Y  V61A        Y        0   IX
                                 V61B                 0   IX
DSN9022I @ DSNB1CMD '-DIS BPOOL' NORMAL COMPLETION
```

Figure 43. Partial DISPLAY DATABASE output showing the list of members that hold the page set or partition P-lock in "U" state

GUPI

Group buffer pool checkpoint

When a group buffer pool is damaged, all changed data that belong to GBP-dependent page sets must be recovered to the page sets from the Db2 logs.

The number of log records that need to be applied to the page set is determined by the frequency of the group buffer pool checkpoint. *Group buffer pool checkpoint* is the process of writing all changed pages in the group buffer pool (only the primary one, if duplexed) to the page set. The purpose of the checkpoint is to reduce the amount of time needed to recover data in a group buffer pool. At group buffer pool checkpoint, Db2 records, in the member BSDSs and SCA, the log record sequence number from which group buffer pool recovery would need to take place. Group buffer pool checkpoint does not record anything in the log.

The group buffer pool checkpoint is triggered by the structure owner. The *structure owner* is usually the first member that connects to this group buffer pool, although the ownership can change over time. Message DSNB798I in the DISPLAY GROUPBUFFERPOOL command output shows which member is the current structure owner.

Default checkpoint frequency

You can change the default checkpoint frequency by using the ALTER GROUPBUFFERPOOL command.

The default checkpoint frequency is 4 minutes. For group buffer pools defined as GBPCACHE(NO), the checkpoint interval is ignored; no checkpointing occurs for those group buffer pools.

Related concepts

Changing the checkpoint frequency

Use the ALTER GROUPBUFFERPOOL command to change the checkpoint frequency.

How Db2 gathers checkpoint information

At the group buffer pool checkpoint, the structure owner records in the SCA and in its own BSDS, the LRSN from which group buffer pool recovery should take place, if necessary.

This LRSN is displayed in the DISPLAY GROUPBUFFERPOOL command output. Db2 has two possible ways of gathering checkpoint information:

- By issuing many "read directory info" requests

  This method is used when the Parallel Sysplex is not at the maintenance level required for the more efficient method described below. These "read directory info" requests are reported as an increase in the SRB time of the *ssnm*DBM1 address space, especially for the structure owner. This is because of the increased number of times that Db2 has to read the directory entries to compute the recovery LRSN.
- By issuing one "read castout statistics" request

  This method is used when the group buffer pool is allocated in a coupling facility at CFLEVEL=5 or higher.

  When you look at the MDETAIL report from a DISPLAY GROUPBUFFERPOOL command you will see significantly fewer "read directory info" requests when you have the proper maintenance applied for this feature.

Recommendation: Because group buffer pool checkpoint consumes processor, coupling facility, and I/O resources and can impact other work in the system, balance the performance impact of frequent group buffer pool checkpoints (the lower the checkpoint interval, the higher the system resource consumption) with the recovery impact of infrequent checkpoints (the lower the checkpoint interval, the faster Db2 can recover from a group buffer pool failure). The default checkpoint interval of 4 minutes is a good balance between the performance and recovery considerations in most cases.

Related tasks

Monitoring and tuning group buffer pool checkpoint intervals
You can monitor your group buffer pool checkpoint intervals and adjust the intervals if necessary.

Monitoring and tuning group buffer pool checkpoint intervals

You can monitor your group buffer pool checkpoint intervals and adjust the intervals if necessary.

Procedure

To monitor and tune group buffer pool checkpoint intervals:

1. Periodically monitor your group buffer pool checkpoint by using any of the following options:
   - Issue the DISPLAY GROUPBUFFERPOOL command.
   - Use IFCID 0261, which gives summary statistics for each group buffer pool checkpoint. You can use this record to estimate the processor cost and to monitor the coupling facility interactions for each group buffer pool checkpoint.
   - Use IFCID 0263, which gives summary statistics for the castouts. You can use this record to monitor the castout activity that is caused by each group buffer pool checkpoint (or triggered for any other reason).
2. Tune the group buffer pool checkpoint as necessary.
   - If the resource consumption of the group buffer pool checkpoint is higher than you prefer:
     - Apply the proper maintenance and allocate the group buffer pool in a coupling facility at CFLEVEL=5 to take advantage of the checkpoint performance enhancement.
     - Increase the checkpoint interval to have the checkpoint occur less frequently.
   - If the checkpoint is not moving the recovery LRSN forward fast enough, decrease the checkpoint interval. You can determine the LRSN by periodically issuing the DISPLAY GROUPBUFFERPOOL command.

- If the recovery LRSN for group buffer pool checkpoint is not advancing as fast as you want, determine if there have been any disk or coupling facility connectivity problems that are impairing the ability of Db2 to cast out.

Examples

In the following example, the DISPLAY GROUPBUFFERPOOL command shows which member is the structure owner and also shows the group buffer pool checkpoint recovery LRSN.

```
DSNB798I -DB1A LAST GROUP BUFFER POOL CHECKPOINT 17:23:21 OCT  4, 2002
               GBP CHECKPOINT RECOVERY LRSN              = ACD74C01EE30
               STRUCTURE OWNER                           = DB1A
```

Figure 44. Partial DISPLAY GROUPBUFFERPOOL output showing which member owns the structure and the group buffer pool checkpoint recovery LRSN

In the following example, the DISPLAY GROUPBUFFERPOOL command with MDETAIL option contains the number of checkpoints that occurred for this group buffer pool. The statistics trace also includes this information.

```
DSNB778I -DB1A CASTOUT THRESHOLDS DETECTED
               FOR CLASSES                               = 3
               FOR GROUP BUFFER POOL                     = 1
             GBP CHECKPOINTS TRIGGERED                   = 1
             PARTICIPATION IN REBUILD                    = 0
DSNB796I -DB1A CASTOUTS
               PAGES CASTOUT                             = 18
               UNLOCK CASTOUT                            = 3
               READ CASTOUT CLASS                        = 5
               READ CASTOUT STATISTICS                   = 6
               READ DIRECTORY INFO                       = 0
```

Figure 45. DISPLAY GROUPBUFFERPOOL MDETAIL report

If you are experiencing surges of coupling facility use, it could be related to group buffer pool checkpointing. Examine the number of read directory info requests. If there are many requests per checkpoint, you can probably benefit from applying the proper maintenance to use the group buffer pool checkpoint performance enhancement.

Group buffer pool thresholds

You can control the castout process with two group buffer pool thresholds.

The two group buffer pool thresholds are:

- Group buffer pool castout threshold
- Class castout threshold

These thresholds have no effect for GBPCACHE(NO) group buffer pools.

The group buffer pool castout threshold is a percentage of changed pages in the group buffer pool. The class castout threshold is a percentage of changed pages or number of changed pages in the group buffer pool per *castout queue*.

Group buffer pool class castout threshold

Each group buffer pool contains a fixed number of castout class queues. This number is an internal value that is set by Db2.

Db2 internally maps updated pages that belong to the same page sets or partitions to the same castout class queues. Because of a limited number of castout class queues, it is possible that more than one page set or partition gets mapped into the same castout class queue. This internal mapping scheme is the same across all sharing subsystems.

When Db2 writes changed pages to the group buffer pool, it determines how many changed pages are on a particular class castout queue. If the number of changed pages on a specified castout class queue exceeds the threshold, Db2 casts out a number of pages from that queue.

How Db2 determines the class castout threshold for a duplexed group buffer pool

For duplexed group buffer pools, Db2 uses the smaller of the number of data entries in the primary and secondary structures. For example, if the primary structure contains 5000 data entries, the secondary structure contains 1000 data entries, and CLASST is `10,0`, Db2 sets the class castout threshold to 100 pages (10% of 1000 pages).

Default group buffer pool class castout threshold

The default for the class castout threshold (CLASST) is `5,0`, which means that castout is initiated for a particular page set or partition when 5% of the group buffer pool contains changed pages for the class.

Related reference

-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)

Group buffer pool castout threshold

The group buffer pool castout threshold determines the total number of changed pages that can exist in the group buffer pool before castout occurs. Db2 casts out enough class castout queues to bring the number of changed pages below the threshold. Db2 periodically determines whether the threshold is exceeded.

How Db2 determines the group buffer pool castout threshold for a duplexed group buffer pool: For duplexed group buffer pools, it is possible to have smaller secondary structure because of space constraints. In that case, Db2 uses the smaller of the number of data entries in the primary and secondary group buffer pools. For example, if the primary structure contains 5000 data entries and the secondary structure contains 1000 data entries, and GBPOOLT is 30%, Db2 sets GBPOOLT to 300 pages (30% of 1000 pages).

Default group buffer pool castout threshold

The default value for the group buffer pool castout threshold is 30, which means that when the group buffer pool is 30% full of changed pages, castout is initiated.

Recommended castout threshold for LOBs

You should set the group buffer pool castout threshold to zero, or a low value, to reduce the need to have a large group buffer pool for LOBs.

Related reference

-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)

Guidelines for group buffer pool thresholds

In most cases, you should use the CLASST and GBPOOLT thresholds to be the corresponding VDWQT and DWQT thresholds for the local buffer pools if these values perform well locally.

Otherwise, you can use the default values (5% for the class threshold and 30% for the group buffer pool threshold). Depending on your workload, these values help reduce disk contention during castout.

If you find that some writes to the group buffer pool cannot occur because of a lack of storage in the group buffer pool, increase the group buffer pool size, or decrease the group buffer pool castout thresholds. One way to tell if this is happening is to see the detail report of the DISPLAY GROUPBUFFERPOOL command. The following figure shows an example report that indicates the lack of storage problem. Figure 51 on page 181

```
DSNB783I -DB1A CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 May 17, 2002
DSNB784I -DB1A GROUP DETAIL STATISTICS
                 READS
                   DATA RETURNED A                  = 3845
DSNB785I -DB1A     DATA NOT RETURNED
                     DIRECTORY ENTRY EXISTED B      = 27
                     DIRECTORY ENTRY CREATED C      = 28336
                     DIRECTORY ENTRY NOT CREATED D  = 332, 0

DSNB786I -DB1A   WRITES
                   CHANGED PAGES                    = 20909
                   CLEAN PAGES                      = 0
                   FAILED DUE TO LACK OF STORAGE E  = 8
                 CHANGED PAGES SNAPSHOT VALUE       = 974
DSNB787I -DB1A   RECLAIMS
                   FOR DIRECTORY ENTRIES F          = 18281
                   FOR DATA ENTRIES                 = 47
                 CASTOUTS                           = 16073
DSNB788I -DB1A   CROSS INVALIDATIONS
                   DUE TO DIRECTORY RECLAIMS G      = 4489
                   DUE TO WRITES                    = 3624
                   EXPLICIT                         = 0
DSNB762I -DB1A   DUPLEXING STATISTICS FOR GBP0-SEC
                   WRITES
                     CHANGED PAGES                  = 20909
                     FAILED DUE TO LACK OF STORAGE  = 8
                   CHANGED PAGES SNAPSHOT VALUE     = 974
DSNB790I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1A DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

Figure 46. Example output of group detail statistics

The field indicated by E indicates this type of problem.

Effect of GBPCACHE ALL on guidelines

If you are using a group buffer pool to cache pages as they are read in from disk (GBPCACHE ALL page sets), consider lowering the threshold values to allow more space for caching those clean pages.

Ways to tune the castout thresholds

Several facilities can help you more efficiently monitor group buffer pool castout thresholds.

Use the following facilities to help you monitor the castout thresholds:

- The DISPLAY GROUPBUFFERPOOL command with the MDETAIL option.
- The Db2 statistics trace.
- PSPI IFCID 0262, which gives summary statistics for each time that the GBPOOLT threshold is reached. You can use this record to monitor how efficiently the GBPOOLT threshold is handling the castout work.
- IFCID 0263, which gives summary statistics for the castouts done by the page set and partition castout owners. All castout work for a given page set or partition is done by the castout owner. You can use this record to monitor the efficiency with which the page set or partition castout owners are doing their work. PSPI

Related reference

-DISPLAY GROUPBUFFERPOOL (Db2) (Db2 Commands)

Example from MDETAIL report

Using the MDETAIL option of the DISPLAY GROUPBUFFERPOOL command is one option for monitoring the group buffer pool castout thresholds.

GUPI

The following example is a partial output from the command DISPLAY GROUPBUFFERPOOL (GBP0) MDETAIL

```
⋮
DSNB796I -DB1A CASTOUTS
               PAGES CASTOUT                          = 217
               UNLOCK CASTOUT                         = 35
               READ CASTOUT CLASS                     = 47
               READ CASTOUT STATISTICS                = 47
               READ DIRECTORY INFO                    = 290
⋮
```

GUPI

The UNLOCK CASTOUT counter should always be significantly less than the PAGES CASTOUT counter. If, at the very least, it is not less than half, the castout write I/O is not performed efficiently. (The number of pages written per I/O is normally close to the number that is obtained by dividing PAGES CASTOUT by UNLOCK CASTOUT). This is probably because you have random update patterns on the Db2 data.

Ways to monitor group buffer pools

You can monitor group buffer pool activity by using a z/OS or Db2 command, by using a coupling facility activity report, or by using a statistics trace.

Group buffer pool monitoring with the z/OS DISPLAY XCF,STR command

You can monitor group buffer pools with the DISPLAY XCF,STR command.

You can use z/OS command D XCF,STR to get information about coupling facility structures:

- CFRM policy definition
- Preference list
- Coupling facility name
- Connections
- Duplexing status

The following command displays information about GBP1 in group DSNDB0A:

```
D XCF,STR,STRNAME=DSNDB0A_GBP1
```

This particular group buffer pool is duplexed, so you see information about both allocations of the structure (the old structure is the primary structure, and the new structure is the secondary one). The output is similar to the output that is shown in the following figure

:

```
DISPLAY XCF
STRNAME: DSNDB0A_GBP1
 STATUS: REASON SPECIFIED WITH REBUILD START:
           OPERATOR INITIATED
         DUPLEXING REBUILD
         REBUILD PHASE: DUPLEX ESTABLISHED
  POLICY SIZE    : 204800 K
  POLICY INITSIZE: 102400 K
  REBUILD PERCENT: 1
  DUPLEX         : ALLOWED
  PREFERENCE LIST: CACHE01  LF01
  EXCLUSION LIST IS EMPTY

 DUPLEXING REBUILD NEW STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/14/2002 17:01:48
  CFNAME         : LF01
  COUPLING FACILITY: ND01...
                     PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 102400 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : AF6935AA 78004403
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 DUPLEXING REBUILD OLD STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/14/2002 17:00:38
  CFNAME         : CACHE01
  COUPLING FACILITY: ND02...
                     PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 102400 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : AF693567 9B48B802
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DB2_DB1A         01 00010001 UTEC469  DB1ADBM1 002E ACTIVE NEW,OLD
 DB2_DB2A         02 00020001 UTEC469  DB2ADBM1 0031 ACTIVE NEW,OLD
```

Figure 47. z/OS Command D XCF showing group buffer pool information

Related information

z/OS MVS System Commands

Group buffer pool monitoring with the coupling facility activity report of RMF

You can use RMF coupling facility structure reports to monitor group buffer pools.

The figure below shows a portion of an example report.

A value for CHNGD (B) is the percentage of all accesses that were supposed to be done synchronously that had to be done asynchronously. The NO SCH field (A) indicates the amount of time that requests were queued because of a lack of subchannel resources. If the value in B is over 10% or so, and A contains a non-zero value, your configuration might not have enough subchannels to handle your systems' workload.

```
STRUCTURE NAME = DSNDB0A_GBP8      TYPE = CACHE
          # REQ    -------------- REQUESTS -------------   -------------- DELAYED REQUESTS
-------------
SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC)
-----
NAME      AVG/SEC           REQ    ALL     AVG   STD_DEV            REQ    REQ   /DEL
STD_DEV   /ALL

STLABC2   66662    SYNC     61K  43.2%   107.7     32.5      A
          370.3    ASYNC  5677    4.0%   602.8    419.8     NO SCH    0   0.0%    0.0        0.0
0.0
               B  CHNGD     0    0.0%   INCLUDED IN ASYNC
                                                        DUMP      0   0.0%    0.0        0.0
```

Figure 48. Portion of RMF Coupling Facility Structure Activity report

Group buffer pool monitoring with the DISPLAY GROUPBUFFERPOOL command
Use the DISPLAY GROUPBUFFERPOOL command to display information about group buffer pools.

Assume that you want a summary report about group buffer pool zero (GBP0), including all connections to that group buffer pool. Enter the following command:

GUPI

```
-DB1A DISPLAY GROUPBUFFERPOOL(GBP0) CONNLIST(YES)
```

The following figure shows what the display might look like, assuming that the group buffer pool is duplexed.

```
DSNB750I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 FOLLOWS
DSNB755I -DB1A DB2 GROUP BUFFER POOL STATUS
                CONNECTED                                = YES
                CURRENT DIRECTORY TO DATA RATIO          = 5.4
                PENDING DIRECTORY TO DATA RATIO          = 6.0
                CURRENT GBPCACHE ATTRIBUTE               = YES
                PENDING GBPCACHE ATTRIBUTE               = YES
DSNB756I -DB1A  CLASS CASTOUT THRESHOLD                  = 5,0
                GROUP BUFFER POOL CASTOUT THRESHOLD       = 30%
                GROUP BUFFER POOL CHECKPOINT INTERVAL     = 4 MINUTES
                RECOVERY STATUS                          = NORMAL
                AUTOMATIC RECOVERY                       = Y
DSNB757I -DB1A MVS CFRM POLICY STATUS FOR DSNDB0A_GBPO     = NORMAL
                MAX SIZE INDICATED IN MVS POLICY         = 61440 KB
                DUPLEX INDICATOR IN POLICY               = ENABLED
                CURRENT DUPLEXING MODE                   = DUPLEX
                ALLOCATED                                = YES
DSNB758I -DB1A  ALLOCATED SIZE                           = 61440 KB
                VOLATILITY STATUS                        = NON-VOLATILE
                REBUILD STATUS                           = DUPLEXED
                CFNAME                                   = LF01
                OPERATIONAL CFLEVEL                      = 5
                ACTUAL CFLEVEL                           = 7
DSNB759I -DB1A  NUMBER OF DIRECTORY ENTRIES              = 61394
                NUMBER OF DATA PAGES                     = 11370
                NUMBER OF CONNECTIONS                    = 3
DSNB798I -DB1A LAST GROUP BUFFER POOL CHECKPOINT 17:31:23 MAY  9, 2002
                GBP CHECKPOINT RECOVERY LRSN             = ACD74C77388C
                STRUCTURE OWNER                          = DB1A
DSNB799I  -DB1A SECONDARY GBP IS ALLOCATED
                ALLOCATED SIZE                           = 61440 KB
                VOLATILITY STATUS                        = NON-VOLATILE
                CFNAME                                   = LF01
                OPERATIONAL CFLEVEL                      = 5
                ACTUAL CFLEVEL                           = 7
                NUMBER OF DIRECTORY ENTRIES              = 61394
                NUMBER OF DATA PAGES                     = 11370
DSNB766I -DB1A THE CONNLIST REPORT FOLLOWS
DSNB767I -DB1A CONNECTION NAME = DB2_DB1A     , CONNECTION STATUS = A
                   CONNECTOR'S RELEASE                  =     10100
DSNB767I -DB1A CONNECTION NAME = DB2_DB2A     , CONNECTION STATUS = A
                   CONNECTOR'S RELEASE                  =     10100
DSNB767I -DB1A CONNECTION NAME = DB2_DB3A     , CONNECTION STATUS = F
                   CONNECTOR'S RELEASE                  =     10100
DSNB769I -DB1A THE CONNLIST REPORT IS COMPLETE
DSNB790I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1A DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

GUPI

Figure 49. Summary report with connection list included

You can display detailed statistics using the GDETAIL and MDETAIL keywords.

Monitoring delete name requests: Use the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option to determine how many times delete-name requests occur. The following figure shows partial output of the DISPLAY GROUPBUFFERPOOL command which includes delete-name information:

GUPI

```
-DISPLAY GROUPBUFFERPOOL(GBP29) MDETAIL(*)
```

```
DSNB750I - DISPLAY FOR GROUP BUFFER POOL GBP29 FOLLOWS
       .
       .
       .
DSNB797I - OTHER INTERACTIONS
           REGISTER PAGE                              = 0
           UNREGISTER PAGE                            = 0
           DELETE NAME                                = 0
           READ STORAGE STATISTICS                    = 0
           EXPLICIT CROSS INVALIDATIONS               = 0
           ASYNCHRONOUS GBP REQUESTS                  = 0
       .
       .
       .
DSNB793I - DELETE NAME LIST                           = 0
           READ CASTOUT STATISTICS                    = 0
           DELETE NAME                                = 0
           OTHER ASYNCHRONOUS GBP REQUESTS            = 0
DSNB790I - DISPLAY FOR GROUP BUFFER POOL GBP29 IS COMPLETE
DSN9022I - DSNB1CMD '-DISPLAY GBPOOL' NORMAL COMPLETION
```

GUPI

Figure 50. DISPLAY GROUPBUFFERPOOL command with MDETAIL option showing delete name information

You can also monitor delete-name requests by using the Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS Statistics Detail report or IFCID 263 (Page set castout detail).

Recommendation:

For an easy way to collect interval statistics for performance analysis, create a batch job that issues the following command periodically:

GUPI

```
-DB1A DISPLAY GROUPBUFFERPOOL(*) GDETAIL(INTERVAL)
```

GUPI

The first time you run the batch job is the base which purges existing statistics and resets the interval. If you run the job the second time 5 minutes after the first, you can continue running the job every 5 minutes to gather meaningful statistical data on group buffer pool activity.

Related concepts

Group buffer pool size is too small
Pages in the group buffer pool need to be refreshed from disk more often because they are not in the group buffer pool.

What to look for in a OMEGAMON statistics report
You can use Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics reports to understand group buffer pool activity.

Related reference

-DISPLAY GROUPBUFFERPOOL (Db2) (Db2 Commands)

When to use Db2 statistics trace

Use Db2 statistics class 1 to do high-level monitoring of Db2 subsystem activity.

A data section mapped by DSNDQBGL records statistics for a member's use of group buffer pools. The counters are cumulative since the time the member connected to a particular group buffer pool.

Statistics reporting intervals are not synchronized across the members of the data sharing group. For counters that are pertinent to the entire data sharing group, like group buffer statistics, Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS group-scope statistics reports combine the data of the individual members and present it for the entire group. The member data is apportioned to the same user-specified interval. Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS presents the synchronized statistics intervals for each member, adds the counters across all members and presents them as statistics on a per-group basis.

Consider also using Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS to do group-scope exception reporting when a particular counter exceeds a user-specified value.

Related concepts

What to look for in a OMEGAMON statistics report

You can use Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics reports to understand group buffer pool activity.

Determining the correct size and ratio of group buffer pools

One of the critical tuning factors in a Db2 data sharing configuration is the size of the group buffer pools.

Consider the following aspects of group buffer pool (cache structure) size:

Total structure size
The total structure size of a group buffer pool is specified in the coupling facility policy definition for the cache structure.

Number of directory entries
A directory entry is used by the coupling facility to determine where to send cross-invalidation signals when a page of data is changed or when that directory entry must be reused. A directory entry contains control information for one database page, no matter in how many places that page is cached. For example, if page P1 is cached in the group buffer pool and in the buffer pools of three members, that page still has only one directory entry.

Number of data entries
Data entries are the actual places where the data page resides. These are 4 KB, 8 KB, 16 KB, or 32 KB in size (the same size as the data page).

For GBPCACHE NO group buffer pools, no data entries exists.

The number of directory entries and data entries in the coupling facility structure is determined by the size specified in the coupling facility policy and the ratio of directory entries to data pages. The ratio is automatically defined for each group buffer pool at the time that the first member of the group is installed.

For secondary group buffer pools, the ratio is the same as the ratio used for the primary group buffer pools.

After installation, you can change the ratio with the ALTER GROUPBUFFERPOOL command. However, the change does not take effect until the next time the group buffer pool is allocated.

The following sections describe the symptoms and values that are not ideal for best performance, and discuss how you can fix the problems.

Related concepts

Storage estimates for data sharing environments (Db2 Installation and Migration)

Group buffer pool sizes (Db2 Installation and Migration)

Related reference

-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)

zSeries Processor Resource/Systems Manager Planning Guide

Group buffer pool size is too small

Pages in the group buffer pool need to be refreshed from disk more often because they are not in the group buffer pool.

When the group buffer pool is too small, the following problems can occur:

- The thresholds for changed pages are reached more frequently, causing data to be cast out to disk more often.

  If castout cannot keep up with the writes to the group buffer pool, a more serious problem occurs: pages are instead written to the logical page list and are unavailable until they are recovered.

- Many cross-invalidations caused by reusing existing directory entries, which might require refreshing a page from disk later when the page is referenced again.

You can use the GDETAIL option of the DISPLAY GROUPBUFFERPOOL command to gather detailed statistical information about how often data is returned on a read request to the group buffer pool:

GUPI

```
-DB1A DISPLAY GROUPBUFFERPOOL(GBP0) GDETAIL(*)
```

The following figure shows output similar to the detail portion of the report output.

```
DSNB783I -DB1A CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 May 17, 2002
DSNB784I -DB1A GROUP DETAIL STATISTICS
                 READS
                   DATA RETURNED A                  = 3845
DSNB785I -DB1A     DATA NOT RETURNED
                     DIRECTORY ENTRY EXISTED B      = 27
                     DIRECTORY ENTRY CREATED C      = 28336
                     DIRECTORY ENTRY NOT CREATED D  = 332, 0

DSNB786I -DB1A   WRITES
                   CHANGED PAGES                    = 20909
                   CLEAN PAGES                      = 0
                   FAILED DUE TO LACK OF STORAGE E  = 8
                 CHANGED PAGES SNAPSHOT VALUE       = 974
DSNB787I -DB1A   RECLAIMS
                   FOR DIRECTORY ENTRIES F          = 18281
                   FOR DATA ENTRIES                 = 47
                 CASTOUTS                           = 16073
DSNB788I -DB1A   CROSS INVALIDATIONS
                   DUE TO DIRECTORY RECLAIMS G      = 4489
                   DUE TO WRITES                    = 3624
                   EXPLICIT                         = 0
DSNB762I -DB1A   DUPLEXING STATISTICS FOR GBP0-SEC
                   WRITES
                     CHANGED PAGES                  = 20909
                     FAILED DUE TO LACK OF STORAGE  = 8
                   CHANGED PAGES SNAPSHOT VALUE     = 974
DSNB790I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1A DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

Figure 51. Example output of group detail statistics

GUPI

What you need to determine is the *read-hit* percentage. To calculate this value, you need to determine how many of the total number of reads were successful in returning data. Use the following formula:

```
( A / ( A + B + C + D (first number)))× 100
```

In this example, the calculation is:

```
(3845 / 32540) × 100 = 11.81%
```

Data was returned in approximately 12% of the read requests to the group buffer pool. This low percentage of read hits might indicate that the average residency time for a cached page in group buffer pool is too short. You might benefit from altering the group buffer pool to increase the total size.

However, a low percentage of read hits could be caused by other factors:

- A high read-to-write ratio.

  If you are caching only changed pages, not many pages you need will be resident in the group buffer pool.
- Random reference patterns.

  Pages that are frequently referenced are most likely to be resident in the group buffer pool. If the application keeps requesting new pages, any given page is unlikely to be found in the group buffer pool.

To determine whether the low read-hit percentage is a problem, see the field indicated by B in the statistics report, shown in the following figure. Figure 56 on page 187 (The same counter also exists in the accounting report.) Ideally, that field contains 0. A non-zero value in the field, in conjunction with a low read hit percentage, can indicate that your group buffer pool is too small.

```
GROUP BP30                        QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------------   --------  -------  -------  -------
GROUP BP R/W RATIO (%)              27.26      N/A      N/A      N/A
GBP-DEPENDENT GETPAGES             651.4M    45.4K   233.60   102.02
SYN.READ(XI)-DATA RETURNED   A     1091.6K    76.12     0.39     0.17
SYN.READ(XI)-NO DATA RETURN  B      133.0K     9.28     0.05     0.02
SYN.READ(NF)-DATA RETURNED   C     1062.1K    74.07     0.38     0.17
SYN.READ(NF)-NO DATA RETURN        14515.1K  1012.21     5.21     2.27
UNREGISTER PAGE              D     4323.7K   301.52     1.55     0.68

CLEAN PAGES SYNC.WRITTEN              0.00     0.00     0.00     0.00
CLEAN PAGES ASYNC.WRTN                0.00     0.00     0.00     0.00
REG.PAGE LIST (RPL) REQUEST         4025.4K   280.71     1.44     0.63
NUMBER OF PAGES RETR.FROM GBP        750.0K    52.30     0.27     0.12
PAGES CASTOUT                E     3033.5K   211.54     1.09     0.48
UNLOCK CASTOUT               F      115.7K     8.07     0.04     0.02

READ CASTOUT CLASS           G      105.4K     7.35     0.04     0.02
READ DIRECTORY INFO          H    69375.00     4.84     0.02     0.01
READ STORAGE STATISTICS      I     9237.00     0.64     0.00     0.00
REGISTER PAGE                J      307.2K    21.43     0.11     0.05
DELETE NAME                  K    24191.00     1.69     0.01     0.00
ASYNCH GBP REQUESTS          L     7532.5K   525.28     2.70     1.18

EXPLICIT X-INVALIDATIONS     M        0.00     0.00     0.00     0.00
CASTOUT CLASS THRESHOLD      N       78.00     0.01     0.00     0.00
GROUP BP CASTOUT THRESHOLD   O       24.00     0.00     0.00     0.00
GBP CHECKPOINTS TRIGGERED    P      100.00     0.01     0.00     0.00
WRITE FAILED-NO STORAGE      Q        0.00     0.00     0.00     0.00

WRITE TO SEC-GBP FAILED               0.00     0.00     0.00     0.00
DELETE NAME LIST SEC-GBP            115.6K     8.06     0.04     0.02
DELETE NAME FROM SEC-GBP           5718.00     0.40     0.00     0.00
UNLOCK CASTOUT STATS SEC-GBP R        0.00     0.00     0.00     0.00
ASYNCH SEC-GBP REQUESTS             599.00     0.04     0.00     0.00

WRITE AND REGISTER                 4220.8K   294.34     1.51     0.66
WRITE AND REGISTER MULT            1350.3K    94.17     0.48     0.21
CHANGED PGS SYNC.WRTN        S     7901.0K   550.98     2.83     1.24
CHANGED PGS ASYNC.WRTN       T     2752.1K   191.92     0.99     0.43
PAGES WRITE & REG MULT             6432.4K   448.57     2.31     1.01
READ FOR CASTOUT                  29352.00     2.05     0.01     0.00
READ FOR CASTOUT MULT               407.8K    28.44     0.15     0.06

PAGE P-LOCK LOCK REQ               2604.3K   181.61     0.93     0.41
 SPACE MAP PAGES                    844.2K    58.87     0.30     0.13
 DATA PAGES                        1760.1K   122.74     0.63     0.28
 INDEX LEAF PAGES                     0.00     0.00     0.00     0.00

PAGE P-LOCK UNLOCK REQ             2478.9K   172.87     0.89     0.39

PAGE P-LOCK LOCK SUSP             98270.00     6.85     0.04     0.02
 SPACE MAP PAGES                  48419.00     3.38     0.02     0.01
 DATA PAGES                       49851.00     3.48     0.02     0.01
 INDEX LEAF PAGES                     0.00     0.00     0.00     0.00

PAGE P-LOCK LOCK NEG              63760.00     4.45     0.02     0.01
 SPACE MAP PAGES              U   45651.00     3.18     0.02     0.01
 DATA PAGES                   V   18109.00     1.26     0.01     0.00
 INDEX LEAF PAGES             W       0.00     0.00     0.00     0.00
```

Figure 52. Portion of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics detail report showing GBP activity

Related concepts

Changing the size of the group buffer pool
You can use two methods to change the size of the group buffer pool.

Problem: storage shortage in the group buffer pool

In a data sharing environment, you might encounter a problem that is caused by a storage shortage in the group buffer pool.

Too few data entries
If a group buffer pool does not have enough data entries, castout to disk occurs more frequently.

Too few directory entries
When existing directory entries are being reclaimed to handle new work, cross-invalidation must occur for all of the members that have the particular data pages in their buffer pools, even when the data has not actually changed.

Monitoring storage of the group buffer pool

By periodically monitoring the storage use of the group buffer pool, you can avoid data outages that are caused by a lack of storage in the group buffer pool.

Procedure

Issue the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option.

The GDETAIL statistics show a snapshot value of the number of changed pages in the group buffer pool. Ensure that this snapshot value (C) does not rise significantly above the group buffer pool castout threshold (B × A). The following figure highlights the key fields from the report.

```
DSNB756I -DB1A   CLASS CASTOUT THRESHOLD                    = 5,0
              A  GROUP BUFFER POOL CASTOUT THRESHOLD        = 30%
                 GROUP BUFFER POOL CHECKPOINT INTERVAL      = 4 MINUTES
                 RECOVERY STATUS                            = NORMAL
                 AUTOMATIC RECOVERY                         = Y
DSNB759I -DB1A   NUMBER OF DIRECTORY ENTRIES                = 61394
              B  NUMBER OF DATA PAGES                       = 11370
                 NUMBER OF CONNECTIONS                      = 3
DSNB783I -DB1A CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 Mar 17, 2002
DSNB784I -DB1A GROUP DETAIL STATISTICS
⋮
DSNB786I -DB1A   WRITES
                   CHANGED PAGES                      = 1576
                   CLEAN PAGES                        = 0
                   FAILED DUE TO LACK OF STORAGE      = 0
                C  CHANGED PAGES SNAPSHOT VALUE       = 311
⋮
```

Figure 53. Partial output of DISPLAY GROUPBUFFERPOOL command

GUPI

Related reference

-DISPLAY GROUPBUFFERPOOL (Db2) (Db2 Commands)

Too few directory entries

When existing directory entries are being reclaimed to handle new work, cross-invalidation must occur for all of the members that have the particular data pages in their buffer pools, even when the data has not actually changed.

GUPI

For example, in the following figure, F indicates that there have been 18,281 directory reclaims. Figure 51 on page 181

```
DSNB783I -DB1A CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 May 17, 2002
DSNB784I -DB1A GROUP DETAIL STATISTICS
                READS
                  DATA RETURNED A                  = 3845
DSNB785I -DB1A    DATA NOT RETURNED
                    DIRECTORY ENTRY EXISTED B      = 27
                    DIRECTORY ENTRY CREATED C      = 28336
                    DIRECTORY ENTRY NOT CREATED D  = 332, 0

DSNB786I -DB1A  WRITES
                  CHANGED PAGES                    = 20909
                  CLEAN PAGES                      = 0
                  FAILED DUE TO LACK OF STORAGE E  = 8
                CHANGED PAGES SNAPSHOT VALUE       = 974
DSNB787I -DB1A  RECLAIMS
                  FOR DIRECTORY ENTRIES F          = 18281
                  FOR DATA ENTRIES                 = 47
                CASTOUTS                           = 16073
DSNB788I -DB1A  CROSS INVALIDATIONS
                  DUE TO DIRECTORY RECLAIMS G      = 4489
                  DUE TO WRITES                    = 3624
                  EXPLICIT                         = 0
DSNB762I -DB1A  DUPLEXING STATISTICS FOR GBP0-SEC
                  WRITES
                    CHANGED PAGES                  = 20909
                    FAILED DUE TO LACK OF STORAGE  = 8
                  CHANGED PAGES SNAPSHOT VALUE     = 974
DSNB790I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1A DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

Figure 54. Example output of group detail statistics

G indicates that, because of those reclaims, 4489 cross-invalidations occurred. The pages in those members' buffer pools need to be refreshed when next needed, probably from disk, which can degrade system performance.

If there is a high value in **G**, check the group buffer pool hit percentage to see if the lack of directory entries might be causing an excessive number of reads from disk.

You can also check the "SYNCHRONOUS READS DUE TO BUFFER INVALIDATION" counters shown in the member detail report.

You can increase the number of directory entries in the group buffer pool in one of two ways:

- Increase the total size of the group buffer pool.
- Use the ALTER GROUPBUFFERPOOL command to adjust the ratio in favor of directory entries.

GUPI

Related concepts

Changing the size of the group buffer pool
You can use two methods to change the size of the group buffer pool.

Group buffer pool size is too small
Pages in the group buffer pool need to be refreshed from disk more often because they are not in the group buffer pool.

Related tasks

Changing the ratio of directory to data entries
You can use the ALTER GROUPBUFFERPOOL command to change the ratio of directory to data entries.

Too few data entries

If a group buffer pool does not have enough data entries, castout to disk occurs more frequently.

GUPI You can see the number of pages cast out by using the GDETAIL option of the DISPLAY GROUPBUFFERPOOL command.

A more serious data entry shortage is indicated by field **E** in the DISPLAY GROUPBUFFERPOOL GDETAIL report shown in the following figure. Figure 51 on page 181

```
DSNB783I -DB1A CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 May 17, 2002
DSNB784I -DB1A GROUP DETAIL STATISTICS
                READS
                  DATA RETURNED A                  = 3845
DSNB785I -DB1A    DATA NOT RETURNED
                    DIRECTORY ENTRY EXISTED B      = 27
                    DIRECTORY ENTRY CREATED C      = 28336
                    DIRECTORY ENTRY NOT CREATED D  = 332, 0

DSNB786I -DB1A  WRITES
                  CHANGED PAGES                    = 20909
                  CLEAN PAGES                      = 0
                  FAILED DUE TO LACK OF STORAGE E  = 8
                CHANGED PAGES SNAPSHOT VALUE       = 974
DSNB787I -DB1A  RECLAIMS
                  FOR DIRECTORY ENTRIES F          = 18281
                  FOR DATA ENTRIES                 = 47
                CASTOUTS                           = 16073
DSNB788I -DB1A  CROSS INVALIDATIONS
                  DUE TO DIRECTORY RECLAIMS G      = 4489
                  DUE TO WRITES                    = 3624
                  EXPLICIT                         = 0
DSNB762I -DB1A  DUPLEXING STATISTICS FOR GBP0-SEC
                  WRITES
                    CHANGED PAGES                  = 20909
                    FAILED DUE TO LACK OF STORAGE  = 8
                  CHANGED PAGES SNAPSHOT VALUE     = 974
DSNB790I -DB1A DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1A DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

Figure 55. Example output of group detail statistics

A value in this field indicates that the data page resources of the coupling facility are being consumed faster than the Db2 castout processes can free them.

You can increase the number of data entries in the group buffer pool in one of two ways:

- Increase the total size of the group buffer pool.
- Use the ALTER GROUPBUFFERPOOL command to adjust the ratio in favor of data entries.

GUPI

Related concepts

Changing the size of the group buffer pool
You can use two methods to change the size of the group buffer pool.

Related tasks

Changing the ratio of directory to data entries
You can use the ALTER GROUPBUFFERPOOL command to change the ratio of directory to data entries.

Auto Alter capabilities

You can avoid cross invalidations due to directory reclaims and writes failed due to lack of storage by using a z/OS capability called Auto Alter.

This capability is initiated by the z/OS structure full monitoring that occurs regularly for each structure. Structure full monitoring adds support for the monitoring of objects within a coupling facility structure. This type of monitoring determines the level of usage for objects within a coupling facility, and issues a warning message if a structure full condition is imminent. Doing this allows tuning actions to avoid a structure full condition. Structure full monitoring occurs every few seconds.

Auto Alter has algorithms that can request the coupling facility to dynamically increase or decrease the number of entries and/or data page elements to avoid structure full conditions. It can increase or decrease the size of the structure if necessary.

Note: The design point for Auto Alter is for gradual growth, not to handle spikes in the workload.

Auto Alter applies to all coupling facility structures. For the lock structure, it can only dynamically increase the RLE portion of the structure. It can increase the size of the SCA. Its main value in Db2; however, is for the group buffer pools. Guessing at a proper directory to data ratio is almost impossible. A

group buffer pool usually needs more than a 5:1 ratio. Static values might or might not be accurate and, over time, they could change as the workload changes. Auto Alter can pinpoint precisely the ratio needed at any given time and change the ratios dynamically. By comparison, if the ratio is changed by operator command, the structure must be manually rebuilt, which causes a slight disruption.

When either the directory entries or data pages exceed the FULLTHRESHOLD, XES will increase the size of the structure and will increase the component in short supply while decreasing the other one. For Db2 group buffer pools, the shortages usually occur for the data elements, not directory entries.

Note that if there is general storage stress on the coupling facility (less than 10% free storage), XES can decrease those structures with ALLOWAUTOALTER(YES). XES will never decrease them below MINSIZE (defaulted to 75% of INITSIZE).

Even though XES is directing the coupling facility to make the changes, the Db2 -DISPLAY GROUPBUFFERPOOL command reflects the current directory to data ratios. The total numbers of directory entries and data elements are found in the coupling facility usage summary section of the RMF coupling facility activity report.

These are the main Auto Alter capabilities:

- Auto Alter supports autonomic tuning, because you set the sizes and Auto Alter does the rest.
- Auto Alter alters ratios in the coupling facility without rebuilding the structure (versus the Db2 -ALTER GROUPBUFFERPOOL command that requires a rebuild).
- Auto Alter builds a better directory to data ratio than manual tuning (up to 40:1). Reclaim avoidance adjusts dynamically to changes with workload.
- Auto Alter can avoid the error: `Writes failed due to lack of storage`.
- Auto Alter allows database administrators to change local buffer pool sizes or add a member without needing a CFRM policy change. It adjusts to gradual growth in the workload.

The way you should have sized your group buffer pools is for proactive tuning, where there is enough storage in each structure to allow for growth, either through increases in local buffer pools or by adding another Db2 member. The goal is to allow group buffer pool tuning to be ignored for a certain amount of time, perhaps six months to a year. This approach is safe when there is sufficient storage available in the coupling facilities.

Important: Your z/OS systems teams must allow enough white space in each coupling facility to allocate all of the structures in both coupling facilities. In the rare event that a coupling facility fails, for whatever reason, the structures should rebuild quickly in the remaining coupling facility. This is specified in the STRUCTURE statement via the REBUILDPERCENT value of 1, and the PREF keyword, where at least one other coupling facility is referenced.

Once you have sized your structures, it is easy to start using Auto Alter. You can enter the parameter ALLOWAUTOALT(YES) with each STRUCTURE statement for which you want to allow structure alteration. This example shows the GBP0 definition:

```
STRUCTURE NAME=groupname_GBP0
INITSIZE=16000
SIZE=32000
ALLOWAUTOALT(YES)
FULLTHRESHOLD=80
MINSIZE=16000
PREFLIST=(CF2,CF1)
DUPLEX(ENABLED)
REBUILDPERCENT(1)
```

You do not want the group buffer pool to fall below the INITSIZE you have specified, so you can code MINSIZE with the same value as INITSIZE. You also set the FULLTHRESHOLD to 80% (also the default).

Related tasks

Planning a coupling facility policy - Requesting structure size (z/OS MVS Setting Up a Sysplex)

What to look for in a OMEGAMON statistics report

You can use Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics reports to understand group buffer pool activity.

Refer to the Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics detail report shown in the following figure. The fields from that report are used to explain some of the activity that takes place for cross-invalidation and refresh of buffers. Some of the information is the same as that described by the DISPLAY GROUPBUFFERPOOL output, which was covered earlier in this section.

```
GROUP BP30                        QUANTITY  /SECOND  /THREAD  /COMMIT
-------------------------------   --------  -------  -------  -------
GROUP BP R/W RATIO (%)              27.26      N/A      N/A      N/A
GBP-DEPENDENT GETPAGES              651.4M    45.4K   233.60   102.02
SYN.READ(XI)-DATA RETURNED   A     1091.6K    76.12     0.39     0.17
SYN.READ(XI)-NO DATA RETURN  B      133.0K     9.28     0.05     0.02
SYN.READ(NF)-DATA RETURNED   C     1062.1K    74.07     0.38     0.17
SYN.READ(NF)-NO DATA RETURN        14515.1K  1012.21     5.21     2.27
UNREGISTER PAGE              D     4323.7K   301.52     1.55     0.68

CLEAN PAGES SYNC.WRITTEN              0.00     0.00     0.00     0.00
CLEAN PAGES ASYNC.WRTN                0.00     0.00     0.00     0.00
REG.PAGE LIST (RPL) REQUEST         4025.4K   280.71     1.44     0.63
NUMBER OF PAGES RETR.FROM GBP        750.0K    52.30     0.27     0.12
PAGES CASTOUT                E     3033.5K   211.54     1.09     0.48
UNLOCK CASTOUT               F      115.7K     8.07     0.04     0.02

READ CASTOUT CLASS           G      105.4K     7.35     0.04     0.02
READ DIRECTORY INFO          H    69375.00     4.84     0.02     0.01
READ STORAGE STATISTICS      I     9237.00     0.64     0.00     0.00
REGISTER PAGE                J      307.2K    21.43     0.11     0.05
DELETE NAME                  K    24191.00     1.69     0.01     0.00
ASYNCH GBP REQUESTS          L     7532.5K   525.28     2.70     1.18

EXPLICIT X-INVALIDATIONS     M        0.00     0.00     0.00     0.00
CASTOUT CLASS THRESHOLD      N       78.00     0.01     0.00     0.00
GROUP BP CASTOUT THRESHOLD   O       24.00     0.00     0.00     0.00
GBP CHECKPOINTS TRIGGERED    P      100.00     0.01     0.00     0.00
WRITE FAILED-NO STORAGE      Q        0.00     0.00     0.00     0.00

WRITE TO SEC-GBP FAILED               0.00     0.00     0.00     0.00
DELETE NAME LIST SEC-GBP            115.6K     8.06     0.04     0.02
DELETE NAME FROM SEC-GBP           5718.00     0.40     0.00     0.00
UNLOCK CASTOUT STATS SEC-GBP R        0.00     0.00     0.00     0.00
ASYNCH SEC-GBP REQUESTS             599.00     0.04     0.00     0.00

WRITE AND REGISTER                 4220.8K   294.34     1.51     0.66
WRITE AND REGISTER MULT            1350.3K    94.17     0.48     0.21
CHANGED PGS SYNC.WRTN        S     7901.0K   550.98     2.83     1.24
CHANGED PGS ASYNC.WRTN       T     2752.1K   191.92     0.99     0.43
PAGES WRITE & REG MULT             6432.4K   448.57     2.31     1.01
READ FOR CASTOUT                  29352.00     2.05     0.01     0.00
READ FOR CASTOUT MULT               407.8K    28.44     0.15     0.06

PAGE P-LOCK LOCK REQ               2604.3K   181.61     0.93     0.41
 SPACE MAP PAGES                    844.2K    58.87     0.30     0.13
 DATA PAGES                        1760.1K   122.74     0.63     0.28
 INDEX LEAF PAGES                     0.00     0.00     0.00     0.00

PAGE P-LOCK UNLOCK REQ             2478.9K   172.87     0.89     0.39

PAGE P-LOCK LOCK SUSP             98270.00     6.85     0.04     0.02
 SPACE MAP PAGES                  48419.00     3.38     0.02     0.01
 DATA PAGES                       49851.00     3.48     0.02     0.01
 INDEX LEAF PAGES                     0.00     0.00     0.00     0.00

PAGE P-LOCK LOCK NEG              63760.00     4.45     0.02     0.01
 SPACE MAP PAGES              U   45651.00     3.18     0.02     0.01
 DATA PAGES                   V   18109.00     1.26     0.01     0.00
 INDEX LEAF PAGES             W       0.00     0.00     0.00     0.00
```

Figure 56. Portion of Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS statistics detail report showing GBP activity

Explanation of fields

A

The number of requests made to read a page from the group buffer pool because the page was invalidated in the member's buffer pool. The member found the required page in the group buffer pool.

B

The number of requests to read a page from the group buffer pool that were required because the page was invalidated in the member's buffer pool. The member did not find the data in the group buffer pool and had to retrieve the page from DASD. If this number is high, use the following formula to calculate the percentage: B /(A + B). If the result is greater than 10%, consider increasing the size of the group buffer pool.

C

The number of requests made to read a page from the group buffer pool because the page was not in the member's buffer pool. The member found the page in the group buffer pool.

D

The number of times Db2 unregistered interest for a single page. This happens when Db2 steals pages from the member's buffer pool that belong to GBP-dependent page sets or partitions.

E

The number of data pages that were cast out of the member's group buffer pool. Castout to a page set or partition is done by the castout owner of the page set or partition. This is normally the Db2 subsystem that had the first update intent on the page set or partition.

F

The number of times Db2 issued an unlock request to the coupling facility for completed castout I/Os. When pages are cast out to DASD, they are locked for castout in the coupling facility. This castout lock is not an IRLM lock; its function is to ensure that only one system can cast out a given page at a time.

G

The number of requests issued by the group buffer pool to determine which pages, from a particular page set or partition, must be cast out because they are cached as changed pages. This request is issued either by the page set or partition castout owner, or, when the group buffer pool castout threshold is reached by the group buffer pool structure owner.

H

The number of requests issued by the group buffer pool structure owner to read the directory entries of all changed pages in the group buffer pool. This request is issued at group buffer pool checkpoints to record the oldest recovery log record sequence number (LRSN). It is used as a basis for recovery if the group buffer pool fails. Such requests might have to be issued several times for each group buffer pool checkpoint to read the directory entries for all changed pages.

I

The number of times Db2 requested statistics information from the group buffer pool. It is issued by the group buffer pool structure owner at timed intervals to determine whether the group buffer pool castout threshold (GBPOOLT) has been reached.

J

The number of times Db2 registered interest in a single page. These are "register-only" requests, which means that Db2 is not requesting any data back from the request. This request is made only to create a directory entry for the page to be used for cross-invalidation when the page set or partition P-Lock is downgraded from S to IS mode, or from SIX to IX mode.

K

The number of requests made by Db2 to delete directory and data entries associated with a particular page set or partition from the group buffer pool. Db2 issues this request when it changes a page set or partition from GBP-dependent to non GBP-dependent. Db2 also issues this request for objects that are defined with GBPCACHE ALL when those objects are first opened.

L

The number of IXLCACHE invocations for the primary group buffer pool.

M

The number of times an explicit coupling facility cross-invalidation request was issued.

N

The number of times group buffer pool castout was initiated because the group buffer pool class castout threshold was detected.

O

The number of times a group buffer pool castout was initiated because the group buffer pool castout threshold was detected.

P

The number of group buffer pool checkpoints triggered by this member.

Q

The number of coupling facility write requests that could not complete due to a lack of coupling facility storage resources.A high number indicates a shortage of group buffer pool data elements because the size of the group buffer pool is too small, or because castout is not keeping up with the rate that changed pages are written. If the pages were not written to LPL, then the write retry attempt was successful.

R

The number of coupling facility requests to read the castout statistics for the secondary group buffer pool. These requests are issued by the group buffer pool structure owner to check for orphaned data entries in the secondary group buffer pool.

S

The number of changed pages written synchronously to the group buffer pool. Pages are written with Write and Register (WAR) requests or Write and Register Multiple (WARM) requests. At commit time changed pages are forced from the member's virtual buffer pool to the coupling facility.

T

The number of changed pages written asynchronously to the group buffer pool. Pages are written in response to Write and Register (WAR) and Write and Register Multiple (WARM) requests. Changed pages can be written from the member's virtual buffer pool to the group coupling facility before the application commits. This happens when, for example, a local buffer pool threshold is reached, or when P-Lock negotiation forces the pages on the vertical deferred write queue to be written to the group buffer pool.

U

The number of P-lock negotiations between data sharing members for space map pages. If this number is high, identify the objects by using IFCID 0259 performance trace. If the high number is caused by insert-intensive applications, consider using the MEMBER CLUSTER or TRACKMOD NO options, or both.

V

The number of P-lock negotiations between data sharing members for data pages. If this number is high, identify the objects by using IFCID 0259 performance trace. The typical cause is tables spaces that use row-level locking. For small tables, consider using MAXROWS=1, and VDWQT=0 to free page P-locks as soon as possible.

W

The number of p-lock negotiations between members for index leaf pages. If this number is high, identify the objects by using IFCID 0259 performance trace. The typical cause is applications that do ascending or descending updates of index keys. Creating indexes with randomized keys might help this situation. However, doing so might incur other performance degradation because more get-page operations, I/O, and lock requests are needed.

Changing group buffer pools

You can change group buffer pool attributes such as the castout threshold values, checkpoint frequency, and size.

Related tasks

Shutting down the coupling facility

Create a plan for those cases in which it is necessary to shut down a coupling facility to apply maintenance or perform another type of reconfiguration.

Starting duplexing for a structure
When duplexing starts, a period exists in which activity to the structure is quiesced. For this reason, you should start duplexing during a period of low activity in the system.

Stopping duplexing for a structure
If you need to stop duplexing, you must first decide which instance of the structure is to remain as the surviving simplexed structure.

Changing the castout threshold values
Use the ALTER GROUPBUFFERPOOL command to change the group buffer pool castout thresholds.

The CLASST option can be expressed as a percentage of the number of data entries or an absolute number of pages. The GBPOOLT option is expressed as a percentage of the number of data entries.

Examples

The following command, for example, changes the class castout threshold to 15% and the group buffer pool threshold to 55%:

GUPI

```
-DB1A ALTER GROUPBUFFERPOOL(GBP1) CLASST(15,0) GBPOOLT(55)
```

GUPI

The following command, for example, changes the class castout threshold to 2500 pages and the group buffer pool threshold to 40%:

GUPI

```
-DB1A ALTER GROUPBUFFERPOOL(GBP1) CLASST(0,2500) GBPOOLT(40)
```

GUPI

These changes take effect immediately.

Related reference
-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)

Changing the checkpoint frequency
Use the ALTER GROUPBUFFERPOOL command to change the checkpoint frequency.

For example, to indicate that you want group buffer pool checkpoints to occur every three minutes, enter the following command:

GUPI

```
-DB1A ALTER GROUPBUFFERPOOL(GBP1) GBPCHKPT(3)
```

GUPI

This change takes effect immediately.

Changing the size of the group buffer pool
You can use two methods to change the size of the group buffer pool.

The method you choose depends on what level of coupling facility the group buffer pool is allocated and whether the group buffer pool is already allocated at the maximum size. For a duplexed group buffer pool, you are changing the size of both the primary and secondary structure with a single command.

Dynamic method

Check whether both of the following conditions are true.

- The group buffer pool is allocated in a coupling facility with CFLEVEL greater than zero.
- The currently allocated size of the structure is less than the maximum size as defined in the SIZE parameter of the CFRM policy.

If they are, you can enter the following command (this example assumes the group name is DSNDB0A):

```
SETXCF START,ALTER,STRNAME=DSNDB0A_GBPn,SIZE=newsize
```

This example assumes that *newsize* is less than or equal to the maximum size that is defined by the CFRM policy for the group buffer pool.

If the maximum size (SIZE in the CFRM policy) is still not big enough, you must use the method described below in Static Method.

Assume a DISPLAY GROUPBUFFERPOOL command produced the following output:

GUPI

```
⋮
DSNB757I -DB1A MVS CFRM POLICY STATUS FOR DSNDB0A_GBP1 = NORMAL
             MAX SIZE INDICATED IN POLICY              = 4096 KB
             ALLOCATED                                 = YES
DSNB758I -DB1A ALLOCATED SIZE                          = 1024 KB
               VOLATILITY STATUS                       = VOLATILE
               REBUILD STATUS                          = NONE
               DUPLEXING STATUS                        = SIMPLEXED
               CFNAME                                  = LF01
               OPERATIONAL CFLEVEL                     = 5
               ACTUAL CFLEVEL                          = 7
DSNB759I -DB1A NUMBER OF DIRECTORY ENTRIES             = 924
               NUMBER OF DATA PAGES                    = 180
               NUMBER OF CONNECTIONS                   = 2
⋮
```

GUPI

Then you enter the following z/OS command to increase the size:

```
SETXCF START,ALTER,STRNM=DSNDB0A_GBP1,SIZE=1536
```

The DISPLAY GROUPBUFFERPOOL command output might look similar to the following after you alter the size:

GUPI

```
⋮
DSNB757I -DB1A MVS CFRM POLICY STATUS FOR DSNDB0A_GBP1 = NORMAL
             MAX SIZE INDICATED IN POLICY              = 4096 KB
             ALLOCATED                                 = YES
DSNB758I -DB1A ALLOCATED SIZE                          = 1536 KB
               VOLATILITY STATUS                       = VOLATILE
               REBUILD STATUS                          = NONE
               DUPLEXING STATUS                        = SIMPLEXED
               CFNAME                                  = LF01
               OPERATIONAL CFLEVEL                     = 5
               ACTUAL CFLEVEL                          = 7
DSNB759I -DB1A NUMBER OF DIRECTORY ENTRIES             = 1426
               NUMBER OF DATA PAGES                    = 284
               NUMBER OF CONNECTIONS                   = 2
⋮
```

GUPI

Notice that the allocated size, the numbers of directory entries, and the number of data pages has increased. The existing ratio is maintained.

Static method

If any of these conditions are true, you must use the following procedure.

- The group buffer pool is allocated in a coupling facility at CFLEVEL=0.
- The allocated size of the structure is already at the maximum size defined by the SIZE parameter of the CFRM policy.

Because the group buffer pool must be rebuilt, you need to follow this procedure when there is little activity in the group.

1. Increase the storage for the group buffer pool in the CFRM policy, and use DUPLEX(ALLOWED).
2. Run the following z/OS command to start the updated policy:

   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=policyname
   ```

3. Run the following command to stop duplexing:

   ```
   SETXCF STOP,REBUILD,DUPLEX,STRNAME=strname,KEEP=OLD
   ```

4. Run the following command to rebuild the group buffer pool:

   ```
   SETXCF START,REBUILD,STRNAME=strname
   ```

5. Run the following command to start duplexing:

   ```
   SETXCF START,REBUILD,DUPLEX,STRNAME=strname
   ```

6. Change the CFRM policy DUPLEX(ALLOWED) to DUPLEX(ENABLED).
7. Run the following z/OS command to start the updated policy:

   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=policyname
   ```

8. Run the DISPLAY GROUPBUFFERPOOL command to verify that DUPLEX(ENABLED) is in effect.

   If the policy change is still pending, you do not need to rebuild again. Because the current duplexing mode is already DUPLEX, the pending change takes effect on the next rebuild.

Changing the ratio of directory to data entries

You can use the ALTER GROUPBUFFERPOOL command to change the ratio of directory to data entries.

About this task

Procedure

To change the ratio of directory to data entries:

1. For duplexed group buffer pools, stop duplexing to revert the group buffer pool to simplex mode.
2. Issue the ALTER GROUPBUFFERPOOL command with the RATIO keyword to change the ratio.
3. Issue the following command command to rebuild the group buffer pool and have the change take effect.

   ```
   SETXCF START,REBUILD,STRNAME=strname
   ```

4. For duplexed group buffer pools, start duplexing.

Example

GUPI For example, if the current ratio is 10 (that is, there are 10 directory entries to every 1 data page), you can use the following ALTER GROUPBUFFERPOOL command to increase the ratio to 14:

```
-DB1A ALTER GROUPBUFFERPOOL (GBP0) RATIO (14)
```

GUPI

Related tasks

Stopping duplexing for a structure
If you need to stop duplexing, you must first decide which instance of the structure is to remain as the surviving simplexed structure.

Starting duplexing for a structure
When duplexing starts, a period exists in which activity to the structure is quiesced. For this reason, you should start duplexing during a period of low activity in the system.

Related reference

-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)
z/OS SETXCF START command (MVS System Commands)

Changing the GBPCACHE attribute

The GBPCACHE option determines whether the group buffer pool is to be used for both caching and cross-invalidation.

About this task

The GBPCACHE option can be set to YES or NO. A value of YES indicates that the group buffer pool is to be used for both caching and cross-invalidation. A value of NO indicates that the group buffer pool is to be used only for cross-invalidation.

Procedure

To change the GBPCACHE attribute:

1. Submit the ALTER GROUPBUFFERPOOL command with the GBPCACHE option.
2. Rebuild the group buffer pool by using the SETXCF START,REBUILD,STRNAME=*strname* command to make the change take effect.

   A group buffer pool with a GBPCACHE(NO) attribute can be duplexed. However, if the group buffer pool attribute is set to GBPCACHE(YES) and the NO attribute is pending, Db2 ignores the pending GBPCACHE(NO) attribute if you start a duplexing rebuild.

Access path selection in a data sharing group

The way that a data sharing member is configured affects access path selection. Use the EXPLAIN statement to obtain information about access path selection.

Effect of member configuration on access path selection

Because plans and packages are bound on individual members in the group, the way a member is configured influences the access path chosen for statements in that plan or package.

For example, it is possible to have different buffer pool sizes and different RID (record identifier) pool sizes on each member. It is also possible that members are on different CPC models.

When you bind your application from one of the members, Db2 chooses the best access path, given the catalog statistics, CPC model and buffer pool sizes, among other things. Suppose, though, that the selected access path is optimal for the one member, but is a relatively poor choice for a different member in the same group. Because the group shares the catalog and directory, the same plan (and hence the same access paths) are used regardless of member, after the application is bound.

Where to bind in a mixed data sharing configuration: If your data sharing group consists of mixed CPC models, be aware that the speed of a central processor might change your access path. This effect is more likely to occur with long-running queries than with fast-running transactions.

Automatic rebind: The access path can change if automatic rebind occurs while the application is executing on a different member than the one on which the original bind occurred.

How EXPLAIN works in a data sharing group

EXPLAIN informs you about access paths that Db2 chooses.

PSPI

Because EXPLAIN can be run on one member and a plan can be bound and executed on other members in a data sharing group, it is important to know which member performed the EXPLAIN. The PLAN_TABLE's GROUP_MEMBER column contains the name of the member that performed the EXPLAIN. The column is blank if the member was not in a data sharing environment when the EXPLAIN was performed.

PSPI

How Db2 maintains in-memory statistics in data sharing

In data sharing environments, each member subsystem sets it own interval for writing in-memory statistics to real-time statistics tables. However, certain utilities and SQL statements in one member might invalidate the in-memory statistics of other members.

Certain utilities and SQL statements can invalidate the in-memory statistics of other members of a data sharing group. For example, such utilities and statements include:

- The RUNSTATS utility
- The REORG utility
- The LOAD utility
- The BACKUP SYSTEM utility
- Any utility or SQL statement that resets page sets to empty, such as mass delete operations and drop table statements

Therefore, all members must externalize their statistics to the real-time statistics tables and reset their in-memory statistics, at the beginning of any such utility job.

Each member updates their statistics in a serial process, by completing the following actions:

1. Acquires a row lock on the statistics table.
2. Reads the target row from the statistics table.
3. Aggregates the in-memory statistics.
4. Updates the statistics table with the new total values.

Similarly, when a page set is reset to empty, the member that resets a page set notifies the other members. The in-memory statistics for the page set are invalidated in all members.

However, the utilities do not fail if all statistics cannot be externalized, or the empty-page notification process fails.

The NULL value indicates that the statistics are unknown.

In data sharing environments, the values in SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS can be negative for short periods of time. These negative values can occur, because individual subsystems externalize the statistics at different intervals. For example, consider the following situation:

1. You have an empty table space.
2. You insert 1000 rows on member A.
3. You delete these 1000 rows on member B.
4. If member B externalizes the in-memory statistics before member A does, the value of the TOTALROWS column in SYSIBM.SYSTABLESPACESTATS is -1000.
5. After member A has externalized the in-memory statistics, the value of the TOTALROWS column in SYSIBM.SYSTABLESPACESTATS is 0.

Related concepts

When Db2 externalizes real-time statistics (Db2 Performance)

Related tasks

Setting up your system for real-time statistics (Db2 Performance)

Related reference

How utilities affect the real-time statistics (Db2 Performance)

REORG TABLESPACE (Db2 Utilities)

REORG INDEX (Db2 Utilities)

RUNSTATS (Db2 Utilities)

COPY (Db2 Utilities)

SYSTABLESPACESTATS catalog table (Db2 SQL)

Information resources for Db2 11 for z/OS and related products

Information about Db2 11 for z/OS and products that you might use in conjunction with Db2 11 is available online in IBM Knowledge Center.

You can find the complete set of product documentation for Db2 11 for z/OS in IBM Knowledge Center.

You can also download other PDF format manuals for Db2 11 for z/OS from IBM Knowledge Center in PDF format manuals for Db2 11 for z/OS (Db2 Home).

© Copyright IBM Corp. 1994, 2020

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785 US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

© Copyright IBM Corp. 1994, 2020

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below:

© (*your company name*) (*year*).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (*enter the year or years*).

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This information is intended to help you to plan for and administer Db2 11 for z/OS. This information also documents General-use Programming Interface and Associated Guidance Information and Product-sensitive Programming Interface and Associated Guidance Information provided by Db2 11 for z/OS.

General-use Programming Interface and Associated Guidance Information

General-use Programming Interfaces allow the customer to write programs that obtain the services of Db2 11 for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

GUPI General-use Programming Interface and Associated Guidance Information... GUPI

Product-sensitive Programming Interface and Associated Guidance Information

Product-sensitive Programming Interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive Programming Interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

PSPI Product-sensitive Programming Interface and Associated Guidance Information... PSPI

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at: http://www.ibm.com/legal/copytrade.shtml.

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

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

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

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

The glossary is available in IBM Knowledge Center.

See the Glossary topic for definitions of Db2 for z/OS terms.

© Copyright IBM Corp. 1994, 2020

Index

A

ACCESS DATABASE MODE
  NGBPDEP 131
  OPEN 132
access methods
  SNA
    descriptions 56
  TCP/IP
    descriptions 28
access methods, SNA
  overview 56
access methods, TCP/IP
  overview 28
access path, effects 193
accessibility
  keyboard viii
  shortcut keys viii
accounting trace
  global lock contention 151
advisory restart-pending 122
affinity
  for online utility jobs 77
affinity, breaking between systems 68
ALIAS option
  of DSNJU003 utility 49
ALTER GROUPBUFFERPOOL command
  change group buffer pools 190
  GBPCACHE option 193
application design recommendations 128
AREST status 122
authorization
  commands 74
  controlling access in a group 11
automatic
  rebuild of coupling facility structures
    description 95
availability
  provided by data sharing 1

B

BACKOUT DURATION option of panel DSNTIPL 121
backout processing, postponing 121
backup database, recommendations 89
batch processing, designing 129
BINDSPECIFIC
  method for configuring TCP/IP group access 37
BSDS
  method for configuring TCP/IP group access 34
buffer pool
  monitoring and tuning for group 171
  relationship to group buffer pool 6

C

castout
castout *(continued)*
  causes 168
  class queue 172
  description 168
  displaying castout owner 169
central processor complex 6
change log inventory (DSNJU003) utility 48
change log inventory utility
  specifying resynchronization port number 48
checkpoint, group buffer pool
  monitoring and tuning 171
child locks 133
cold start 123
commands
  authorizing 74
  MODIFY DDF 43, 49
  prefix
    displaying 78
  scope 73
communications database (CDB)
  identifying remote data sharing groups, SNA 59
  identifying remote data sharing groups, TCP/IP 46
  mapping Db2 location names 27
  updating 47, 60
concurrency
  data sharing
    tuning recommendations 135
  summary of tuning recommendations 135
conditional restart 122, 123
configuration
  possibilities with data sharing 3
connection request
  Db2 location name 27
connections
  displaying, group buffer pool 177
  exit routine, considerations 11
  failed-persistent 75
  maximum number of distributed 27
  monitoring connections to remote systems 85
contention
  calculating global percentages 147
contention, false
  avoiding 137
  detecting 137
  reducing 138
coupling facility
  DiscFailConn event 119
  duplexed 102, 105
  group buffer pool 101, 102
  lock structure 103
  non-duplexed 101, 104
  planning for shutdown 125
  recovery scenarios 101
  reducing page size overhead 155
  SCA structure 103
  structures
    connection disposition 75

coupling facility *(continued)*
structures *(continued)*
dealing with hung 119
displaying size and usage 78
monitoring 127
structure disposition 75
subset of members 104, 105
Coupling Facility Activity Report of RMF 145, 176
CREATE TABLESPACE statement
MEMBER CLUSTER option 162
TRACKMOD option 164
cross-invalidation 7
cross-system extended services (XES) 119
CURRENT MEMBER, special register 84

D

D XCF,STR command of z/OS
output 80
output showing group buffer pool information 175
data entries for group buffer pool
description 180
symptoms of too few 184
data sharing
advantages 1
concurrency 132
deleting members 24
explicit hierarchical locking 133
flexible configurations 3
locks 132
member on which SQL statements run 84
quiescing members 23
removing members
data sets to keep 23
data sharing group
access methods 27
accessing remote groups in SNA networks 65
accessing remote groups in TCP/IP networks 51
configuring as SNA server
for group-generic access 61
specifying generic LU name 61
configuring as TCP/IP server
designating subsets of members 49
specifying DRDA port number 48
specifying generic LU name 50
specifying resynchronization port number 48
definition 1
description 1
identifying generic LU name 63
maintaining 14
member 1
shutting down network connection to requester 68
subsetting aliases 50
data sharing groups
identifying remote groups 46, 59
data sharing members
deactivating 24
destroying 25
database
backup, recommendations 89
recovery 88
Db2 commands
routing 73
scope 73
Db2 Connect requester
configuring to use group access, TCP/IP 51
requirement to enable Sysplex support 51
DB2 GENERIC LUNAME parameter 60–62
Db2 location name
mapped to network element 27
Db2 location names
mapping to network element 27
Db2 release level
displaying 78
Db2 requester
configuring to use group access, TCP/IP 52
configuring to use group-generic access, SNA 67
configuring to use member-routing access, SNA 65
configuring to use member-specific access, TCP/IP 54
DDF
location alias
dynamic 43
updates the LOCATION 69
deactivating
data sharing members
restoring 26
deadlock 141
deferred restart 123
deferring restart processing 123
DELETE NAME 177
deleting
data sharing members 24
destroying
data sharing members 25
DFSMShsm (Data Facility Hierarchical Storage Manager)
archiving logs in data sharing 86
directory entries of group buffer pool
description 180
symptoms of too few 183
disability viii
disaster recovery, in a group 91
DiscFailConn (Disconnected/Failed Connection) event 119
DISPLAY BUFFERPOOL command
LIST option 158
DISPLAY GROUP command
displaying information about the group 78
DISPLAY GROUPBUFFERPOOL command
group detail report example 180
MDETAIL option 165
monitor group buffer pools 177
summary report example 177
displaying
information about
coupling facility structures 80
data sharing group 78
LPL (logical page list) 82
distributed data
connection limit in a data sharing group 27
recommendations for SNA alternatives 56
Distributed Data Facility
of DSNJU003 utility 69
distributed data facility (DDF) 27, 85
DNS network addressing
example configuration 40
domain name server (DNS) 40
dormant 23
DRDA 27
DRDA port

DRDA port *(continued)*
updating using DSNJU003 utility 69
DRDA port number
recommendation 48
reserving 48
specifying 48
DSN3@ATH connection exit routine 11
DSN3@SGN sign-on exit routine 11
DSN7501A 104
DSN7501I 104
DSNB228I 106
DSNB325A 106
DSNB331 104
DSNB744 105
DSNB7445 105
DSNB745 105
DSNDWQ04 mapping macro 162
DSNHDECP load module
group attachment name 76
subgroup attachment name 76
DSNJU003 34
DSNJU003 (change log inventory) utility
ALIAS option 49
designating subsets of members 49
specifying DRDA port number 48
DSNJU003 utility
specifying resynchronization port number 48
DSNR020I 23
DSNR030I 23
DSNTIP5 installation panel
specifying DRDA port number 48
DUPLEX
option of CFRM policy 123
duplexing 7
DVIPA 29
DVIPA network addressing
example of group access 52
example of member-specific access 54
DXR143I 104
dynamic location aliases
defining 42
managing 43
dynamic method 190

E

exit routine, considerations 11
EXPLAIN statement, executing 194
explicit hierarchical locking 133

F

failed-persistent connection
description 75
protects retained locks 118
failure scenario
connectivity failure
lock structure and SCA 97
duplexed group buffer pool 97
false contention
calculating percentages
statistics trace 149
false lock contention, preventing 137
field procedure, considerations 11
function level of IRLM 14

G

GBP-dependent 6
GBPCACHE 12
GBPCACHE ALL 166
GBPCACHE ALL clause
caching pages during a read 165
planning consideration 165
when to use 165
GBPCACHE attribute of group buffer pools
duplexed group buffer pools 193
procedure for altering 193
GBPCACHE CHANGED 166
GBPCACHE clause
effect on guidelines for group buffer pool castout thresholds 173
GBPCACHE NONE 166
GBPCACHE SYSTEM 166
GBPCACHE SYSTEM clause
use for LOB table spaces 165
general-use programming information, described 200
GENERIC column
SYSIBM.LUNAMES table 62
GENERIC column, SYSIBM.LUNAMES table 60
generic LU name
and RACF PassTickets 65
configuring members to use 62
defining for data sharing group 61
identifying for data sharing group 63
needed for RACF PassTickets 50
removing affinity to a partner 85
updating using DSNJU003 utility 69
GENERIC option
of DSNJU003 utility 69
global contention
calculating percentages 147
global lock contention
limits 138
governor (resource limit facility) 130
GRECP (group buffer pool RECOVER-pending) status 82
group access, TCP/IP
configuring Db2 Connect requester 51
configuring Db2 requester 52
description 29
example using dynamic VIPA network addressing 52
group attachment name
displaying 78
submitting applications using 76
group attachment of online utility jobs 77
group buffer pool
assigning a page set 154
castout threshold guidelines 173
changing size 190
changing using ALTER GROUPBUFFERPOOL 190
checkpoint 170
connectivity failure scenarios 97
cross-invalidation 154
data entries 180
default castout threshold 173
dependency 156, 159
description 154

group buffer pool *(continued)*
determining inner-system sharing 159
directory entries 180
duplexing
altering ratio 192
connections 81
GBPCACHE attribute 193
monitoring storage 183
monitoring using DISPLAY GROUPBUFFERPOOL 177
monitoring using RMF report 176
monitoring using z/OS command D XCF,STR 175
read operations 164
recommended castout threshold 173
RECOVER-pending (GRECP) status
removing using START DATABASE command 101
relationship to virtual buffer pools 6
storage monitoring 183
storage shortage recovery scenario 106
structure failure scenarios 97
thresholds 172
too few data entries 184
too few directory entries 183
too small 180
tuning size and ratio 180
write operations 166
group detail report of DISPLAY GROUPBUFFERPOOL command 180
group domain name 28
group DVIPA 28
group IP address
associated with data sharing group 69
group name
displaying 78
group release level, displaying 78
group restart
description 111
group-generic access, SNA
configuring data sharing groups for 61
configuring Db2 requester 67
description 58
example of 67
switching to member-specific access 68
two-phase commit processing 58
GUPI symbols 200

H

hung coupling facility connections 119

I

ICF (Integrated Coupling Facility) 92
IFCID (instrumentation facility component identifier)
identifiers by number
0045 151
0250 106
0251 162
0259 162
indoubt transactions
resolving 110
Integrated Coupling Facility (ICF) 92
IP address
accept incoming connection requests 69
IRLM (internal resource lock manager)
applying maintenance 14
avoiding false contention 137
displaying subsystem name and procedure name 78
failed-persistent connections 118
function level 14
global transaction
locking 132
ISTGENERIC coupling facility structure for VTAM 61

L

light restart 113
LIMIT BACKOUT parameter of installation panel DSNTIPL1 121
links
non-IBM Web sites 201
LOB table spaces
use GBPCACHE SYSTEM clause 165
location alias
dynamic 42
limitation with RACF PassTickets 65
updating using DSNJU003 utility 69
LOCK ENTRY SIZE parameter of IRLMPROC 138
lock structure size
changing dynamically 152
changing through rebuild 152
LOCKINFO field of DISPLAY DATABASE output, page set P-locks 162
locking
explicit hierarchical 133
locking protocol 2
effect on parent L-locks 138
locks
decreasing lock table entry size 140
detecting global contention 145, 146
false contention 137
global deadlock detection 141
global transaction 132
hierarchy
child locks 133
parent locks 133
physical
description 156
instrumentation data 162
page 161
retained state 160
when obtained 156
retained
P-locks 160
structure
changing size 151
displaying size and usage 78
monitoring usage 144
symptoms of storage shortage 108
XCF message buffer size effect on resolving global contention 140
log
active 86
archive 86
archiving using DFSMShsm 86
log record
applying 88
header 88

log record sequence number (LRSN) 88
logical page list (LPL)
  description 82
  failed group buffer pool write 167
  recovering pages
    methods 94
  status, description 82
LU name of requester
  specifying 64
LUWID option of DISPLAY THREAD command 85

M

maintenance, applying to a group 14
mapping macro, DSNDWQ04 162
MAX REMOTE ACTIVE option 69
MAX REMOTE CONNECTED thread limit 48
MAXDBAT parameter 69
MAXROWS option of CREATE TABLESPACE statement 161
MAXUSRS
  decreasing parameter of IRLMPROC 140
  effect on lock entry size 138
member
  configuring to use generic LU name 62
  definition 1
  preventing from processing DDF requests 69
  subsets 49
  unregistering from Workload Manager 69
MEMBER CLUSTER 12
MEMBER CLUSTER option of CREATE TABLESPACE 129, 162
member consolidation
  changes for 19
member name
  displaying 78
member on which SQL statements run, determining 84
member release level, displaying 78
member-routing access, SNA
  configuring Db2 requester 65
  description 57
  example of 65
  RACF PassTicket limitation 57
  two-phase commit resynchronization 57
member-specific access, SNA
  switching from group-generic access 68
member-specific access, TCP/IP
  configuring Db2 requester 54
  description 41
  example using DVIPA network addressing 54
members
  consolidate 17
message by identifier
  DSN7501I 103
  DSN7504I 103
  DSN7505I 107
  DSN7512A 107
  DSNB228I 101
  DSNB250E 101
  DSNB301E 106
  DSNB303E 101, 104
  DSNB304I 101
  DSNB305I 101
  DSNB314I 101
  DSNB319A 106
  DSNB320I 101
message by identifier *(continued)*
  DSNB321I 101
  DSNB353I 101
  DSNB354I 101
  DSNB743 102, 105
  DSNB744 102
  DSNB745 102
  DSNI006I 94
  DSNI021I 94
  DSNI022I 94
  DXR136I 103
  DXR142E 108
  DXR143I 103
  DXR170I 108
MODIFY DDF command 43
monitoring
  group buffer pool storage 183

N

naming conventions 10
network protocol
  choosing 60
network protocol, choosing 27, 47
node profile
  requester 50

P

P-locks, page
  reducing contention 161
  space map contention 162
page
  locks, physical 161
  validity test in data sharing 164
page set
  determining if group buffer pool dependent 159
  P-lock, determining retained state 160
page set of partition 132
page size
  reducing coupling facility overhead 155
parallel sysplex 10, 12
Parallel Sysplex
  definition 1
parent locks 133
pending work 122
performance
  designing table spaces 129
  improving 128
  migrating batch applications 129
  recommendation 128
  setting expectations 128
performance trace
  global lock contention 151
phases of restart 115
physical lock
  description 156
  instrumentation data 162
  retained state 160
  when obtained 156
physical open 132
PLAN_TABLE table, GROUP_MEMBER column 194
PORT option

PORT option *(continued)*
of DSNJU003 utility 69
postponing backout 121
prefetch processing
for GBP-dependent page sets and partitions 165
product-sensitive programming information, described 200
programming interface information, described 200
PSPI symbols 200

Q

QUIESCED 23
quiescing data sharing members 23

R

RACF PassTickets
generic LU name 65
generic LU name requirement 50
location alias limitation 57, 65
ratio of directory entries to data entries
changing 192
choosing a value 180
description 180
real-time statistics
in-memory statistics 194
reason codes
X'00C20204' 106
X'00C20205' 101
X'00F70609' 106, 107
rebinding automatically
access path change in data sharing 193
RECOVER TABLESPACE utility, options
TOCOPY 90
TOLOGPOINT 90
TORBA 90
recovering databases 88
recovery
after disaster 91
coupling facility 95
description 88
LPL (logical page list) 94
options 90
point-in-time 90
preparing for fast 89
system-level backup 90
remote data sharing groups
identifying 46, 59
removing
data sharing members 23
reports
group detail report 180
summary report 177
requester
definition 27
setting up Db2 for z/OS 45, 59
shutting down network connection to data sharing group 68
requester LU name
specifying 64
RESET GENERICLU command 68
resource limit facility (governor)
data sharing 130
resource limit specification table (RLST)
command scope 130
unique name 130
resource measurement facility (RMF)
Coupling Facility Activity Report 176
restart
conditional 122
Db2 data sharing group 111
deferring processing 123
group 111
light 113
postponing backout processing 121
restoring
data sharing members 26
resynchronization port
updating using DSNJU003 utility 69
resynchronization port number
specifying 48
retained utility ID lock 112
reverting to simplex mode of structure 124
RMF (resource measurement facility)
Coupling Facility Structure Activity report 145
monitor group buffer pools 176
row locking, page P-lock contention 161

S

SCA (shared communications area)
displaying size and usage 78
increasing storage 107
scope of commands 73
SCOPE parameter of IRLMPROC, NODISCON option 14
secure port
updating using DSNJU003 utility 69
SETXCF STOP, REBUILD command of z/OS 124
shared data, restricting access to 155
SHAREPORT option, TCP/IP PORT configuration profile statement 48
shortcut keys
keyboard viii
sign-on exit routine, considerations 11
simplex mode of structure, reverting 124
single-member access, SNA
description 58
single-member access, TCP/IP
description 45
SNA
access methods
group-generic access, description 58
group-generic, example 67
member-routing, description 57
member-routing, example 65
single-member access, description 58
switching from group-generic to member-specific 68
connecting distributed partners 65
definition 56
SNA server
configuring data sharing group as
for group-generic access 61
specifying generic LU name 61
space map, reducing P-lock contention 162
START DATABASE
restrictions 130

startup, Db2 75
static method 190
statistics trace
  false contention 149
  global locking 146
status
  displaying member status 78
  GRECP 82
  LPL 82
STOP DATABASE
  restrictions 130
STOP DDF command 68
storage
  group buffer pool, monitoring 183
structure
  duplexing
    starting 123
    stopping 124
subgroup attachment name
  submitting applications using 76
subgroup attachment of online utility jobs 77
subsets of members 49
subsystem
  name
    for data sharing group members 78
subsystem parameter
  lock modes
    for UPDATE and DELETE 141
  XLKUPDLT 141
summary report, DISPLAY GROUPBUFFERPOOL command 177
synchronous read I/O problem, resolving 131
SYSIBM.IPLIST table
  for member-routing access, TCP/IP 46
  for single-member access, TCP/IP 46
  for TCP/IP access 47
SYSIBM.IPNAMES table
  for group access, TCP/IP 46
  for member-routing access, TCP/IP 46
  for single-member access, TCP/IP 46
  for TCP/IP access 46
SYSIBM.LOCATIONS table
  for group access, TCP/IP 46
  for group-generic access, SNA 59
  for member-routing access, TCP/IP 46
  for member-specific access, SNA 59
  for single-member access, SNA 59
  for single-member access, TCP/IP 46
  for SNA access 59
  for TCP/IP access 46
SYSIBM.LULIST table
  for member-specific access, SNA 59
  for single-member access, SNA 59
  for SNA access 60
SYSIBM.LUNAMES table
  for group-generic access, SNA 59
  for member-specific access, SNA 59
  for single-member access, SNA 59
  for SNA access 60
  GENERIC column 60, 62
SYSLGRNX table space 88
Sysplex
  definition 1
Sysplex Distributor
Sysplex Distributor *(continued)*
  role in DVIPA network addressing 30
SYSPLEX parameter, Db2 Connect requester 51
Sysplex query parallelism 2
system-level point-in-time recovery 91
Systems Network Architecture (SNA) 11

T

table spaces
  LOB
    use GBPCACHE SYSTEM clause 165
TCP/IP
  access methods
    group access, description 29
    member-specific access, description 41
    single-member access, description 45
  configuring for data sharing
    BINDSPECIFIC method 37
    BSDS (INADDR_ANY method) 34
  connecting distributed partners 51
  definition 28
  DNS network addressing
    example configuration 40
  DVIPA network addressing
    example of group access 52
TCP/IP server
  configuring data sharing group as
    designating subsets of members 49
    specifying DRDA port number 48
    specifying generic LU name 50
    specifying resynchronization port number 48
threads
  monitoring 85
threshold, group buffer pool
  castout 172
  class castout default 172
  defaults 173
  guidelines 173
timeout in a group 141
Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
  accounting report 151
Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS (statistics report 187
trace
  accounting 151
  statistics, global locking activity 146
TRACKMOD 12
TRACKMOD option of CREATE and ALTER TABLESPACE 129, 164
transaction rates 6
Transmission Control Protocol/Internet Protocol (TCP/IP) 11
two-phase commit processing
  using group-generic access, SNA 58
two-phase commit resynchronization
  using member-routing access, SNA 57

U

utilities
  identifier 77
  stand-alone 78
  submitting to a group 77

utilities *(continued)*
work data sets 77
utility ID lock, retained 112

V

validation routine, considerations 11
VARY NET,INACT,ID command 68
VTAM (Virtual Telecommunications Access Method)
breaking affinity between systems 68
generic resources 58
VTAM affinity 85

W

workload balancing 2
Workload Manager (WLM)
role in DNS network addressing 40
role in dynamic VIPA network addressing 30
role in member-specific access 41
unregistering members 69

X

XCF (cross-system coupling facility) component of z/OS
message buffer size effect on resolving global contention 140
XES (cross-system extended services) 119
XLKUPDLT
subsystem parameter 141

Z

z/OS commands
D XCF 80



Product Number: 5615-DB2
5697-P43

SC19-4055-05

