WHITE PAPER

W I N T E R   C O R P O R A T I O N

IBM DB2 9.7 AND SMART ANALYTICS SYSTEMS

Architecture and Key Capabilities for Data Warehousing

The Large Scale Data Management Experts

WinterCorp

SPONSORED RESEARCH PROGRAM

W I N T E R   C O R P O R A T I O N

IBM DB2 9.7 AND SMART ANALYTICS SYSTEMS

Architecture and Key Capabilities for Data Warehousing

September 2010

WinterCorp

245 FIRST STREET, SUITE 1800
CAMBRIDGE MA 02145
617-695-1800

visit us at www.wintercorp.com

Executive Summary

THIS WHITE PAPER reviews the product architecture of IBM's DB2 9.7 and IBM's Smart Analytics Systems. In both cases, the paper focuses on data warehousing.

DB2 9.7 for Linux, UNIX, and Windows (DB2 9.7 for LUW) provides IBM's database engine for the InfoSphere 9.7 data warehouse software product family.

DB2 is a mature software product for data warehousing with comprehensive capabilities for performance and scalability. DB2 9.7 is built on a product foundation featuring:

- A shared nothing parallel database architecture providing performance, scalability and high data availability;
- A long term focus on system efficiency, based on extensive capabilities for physical database design, I/O performance, efficient cross partition network utilization, and cost based query optimization;
- Mixed workload management, providing the capability to satisfy multiple service level objectives on a single system managing a single logical copy of the data;
- Data compression; and,
- An engine that is widely used for both OLTP and data warehouse workloads

along with many other features.

In recent years, IBM has continued to invest in DB2 to address emerging customer requirements and keep pace with the extraordinary increases in scale characteristic of challenging data warehouse applications. In DB2 9.7, the latest release, IBM has expanded the functionality of DB2 to address additional needs in mixed workload management; in resource optimization; in system performance management; in schema evolution; and, in other areas described in this paper. These new capabilities in DB2 9.7 provide the customer with increased manageability; better performance; enhanced cost performance; and, agility.

IBM's Smart Analytics System integrates DB2 with a full stack of system software; with business intelligence and data integration software; and, with hardware—to provide customers with a complete data warehouse appliance deliverable in standardized, pre-installed, pre-tested configurations. Thus, the IBM Smart Analytics Systems greatly increase the simplicity of acquiring and deploying IBM's data warehouse capabilities. Overall, DB2 9.7 and IBM's Smart Analytics Systems are among the most capable data warehouse products available on the market today.

In the opinion of WinterCorp, organizations that need a platform for data warehousing, data analytics, or business intelligence, will want to consider the IBM platforms described in this paper.

Table of Contents

List of Illustrations

1 Introduction

The business interests and analytical strategies of today's leaner and more agile enterprises are moving the practice of data warehousing in new directions.

Data warehouses originally developed primarily to tell enterprise strategists what had happened. How many units of ice cream were sold in Memphis this past weekend? What did we spend last week on consumer electronics merchandise? On labor in our New England distribution centers?

But, today, executives are looking for much more: how will the forecasted weather affect demand for frozen desserts across the Southeast? What can we do about it? How much will that cost and what impact will it have?

Understanding *what happened* is no longer enough. Executives want actionable recommendations: what is going to happen and what to do about it. And, they need *that* information in time to act.

Today's executives also want transparent and traceable analytics to back up both the diagnosis of the business problem and its solution. Because businesses need more precisely defined options, these analytics are often more intensive and more complex than the simple statistics that served as business analytics ten years ago. And, executives increasingly seek forecasts of the impact of implementing various options ("predictive analytics"). In addition, executives are looking for data warehouses to support front line operational business processes, where—for example—customer-facing employees are enabled in their day-to-day decision making as they process orders; answer questions; and, service requests.

These shifts in the business expectations surrounding data warehousing—along with technology trends discussed below—have resulted in a profound change in the requirements for building and operating the data warehouse. This White Paper describes these trends and relates them to the architecture and key capabilities of DB2 9.7 for Linux, UNIX and Windows (DB2 9.7 for LUW). DB2 9.7 for LUW is delivered via IBM's leading data warehouse platforms:

- InfoSphere Warehouse 9.7, a software product; and,
- IBM Smart Analytic System, a data warehouse appliance.

IBM Smart Analytic System ships with InfoSphere Warehouse and Cognos software—a suite of business intelligence tools for query, reporting and analysis—pre-installed on an IBM hardware platform.

DB2 9.7 for LUW incorporates a substantial range of other new capabilities, described at length in *What's New for DB2 Version 9.7* (IBM Document #SC27-2463-01, available online at ***publib.boulder.ibm.com/infocenter***).

1.1. HOW DATA WAREHOUSES ARE GROWING AND CHANGING

These evolving business requirements—for predictive analytics, operational business intelligence, faster reaction to events and deeper analysis of options—result in demanding technical requirements. Some of the key requirements for the newest generation of data warehouses are:

- *Frequent update of data*: instead of weekly or nightly batch update, today's data warehouse often requires online update frequently or continuously; stringent data latency requirements are becoming more common (e.g, data must be online and available for query within x hours, minutes or seconds on receipt); high volume ingest requirements are also becoming more common;

- *Mixed workload*: instead of a steady diet of periodic scheduled reporting or analysis, today's data warehouse often must meet multiple concurrent service levels including interactive query; large long running jobs; and, a mix of concurrent query and update;
- *Many concurrent users*: instead of a few users feeding a relatively small complement of management decision makers, data warehouses must now often support large communities of employees, partners or customers concurrently accessing and updating data;
- *High data availability*: as enterprises come to rely on the data warehouse to support operational business processes, data availability requirements rise and often attain mission critical status; thus, the data warehouse often will need to operate on a continuous or near-continuous schedule;
- *Intensive analytics*: as business strategies rise in sophistication, and data volumes continue to increase, there is a rising demand to perform more and more analysis of the data—and to accomplish it in place in the data warehouse. Just as the tolerance for down time decreases, because business needs data access all the time, the tolerance for delay in analysis also decreases. Thus, no one wants to wait while data is exported to another system for analysis. Instead, the requirement is increasingly to analyze the data in place in the data warehouse.

In addition, data warehouses continue to grow in every direction at a rapid pace: most successful data warehouses today increase rapidly in both data volume and usage levels. Thus, even in modest initiatives in smaller enterprises and departments, the data warehouse often grows to present quite stringent requirements in order to deliver and sustain business value over a period of just a few years. As a further implication of the continual rapid increase in scale, solutions must be even more cost effective; deployment and expansion must be rapid and easily accomplished; data warehouse management must be more automated; and, skill requirements per unit of data must continually decrease.

1.2. APPROACH OF DB2 9.7

The foundation of DB2 for LUW has from the beginning addressed many of these issues. With its shared nothing architecture; focus on scalability and performance; wide range of capabilities for database design; and, comprehensive facilities for analytic applications, DB2 for LUW has for years addressed the most fundamental issues in data warehousing.

However, new capabilities in DB2 9.7 address some of the issues of rising importance today.

Resource optimization is more important than ever in today's tough economic environment, and DB2 9.7 addresses this with major new capabilities in data compression, space reclamation and scan sharing. Scan sharing is particularly significant in systems which must support substantial amounts of concurrent query, analysis and reporting: it means that concurrent queries can share the results of a scan, even if they did not start at the same time or at the same point in a table, boosting efficiency for multiple users at once. Data compression reduces the work and time required to search data, while reducing the space required to store it both on disk, and in the database buffer pool(s).

Data warehouses are growing and changing more rapidly than ever before. Business depends on fact based analysis and prediction to a greater extent than at any point in the past. In fact, the ability to quickly adapt the data warehouse to new business requirements is now a critical factor in enterprise competitiveness. DB2 9.7 addresses this challenge with new capabilities to support changing schemas and changing capacity requirements while keeping the system online.

The uses of data are also multiplying. It is no longer acceptable in many cases for a data warehouse to support just one use or one type of activity. On a single copy of the data, users often expect to support two or more of: interactive query, predictive analysis, data mining, reporting, operational business intelligence and various types of loads and updates. This means that single systems must satisfy multiple, often competing, service level objectives. Users that need fast response to simple queries often run on the same system that must run large, complex long running jobs. DB2 9.7 addresses this area with advances in its workload management. In addition, such systems are more complex to monitor and administer. DB2 addresses this area with new capabilities in system administration and monitoring.

Finally, many users are looking for simpler, faster ways to configure, acquire and deploy economical and performant data warehouse capacity. These issues are addressed with IBM's Smart Analytic Systems—the new platforms for running DB2 9.7 for data warehousing/business intelligence. The IBM Smart Analytic Systems are pre-integrated, pre-tested, pre-configured appliances that deliver DB29.7 on AIX, Linux and Windows—and with either Power7 or Intel Nehalem processors. In addition, these systems are available with disk storage and optional solid state storage, which is standard on the Power7-based IBM 7700 They are also available in a wider range of prices and configurations than previously offered.

1.3 ORGANIZATION AND METHODOLOGY OF THIS WHITE PAPER

The remaining sections of this White Paper review key areas of the DB2 product foundation and discuss each of the major new data warehouse in more detail.

Section 2 reviews the architectural foundation of DB2.

Section 3 discusses recent developments in resource optimization, including new data compression features, space reclamation and scan sharing.

Section 4 covers new features to support data warehouse flexibility and growth, including capabilities for schema evolution and system expansion.

Section 5 reviews workload management and monitoring.

Section 6 describes IBM's data warehouse appliance; the Smart Analytics System.

Section 7 summarizes findings and conclusions from this review.

2 *Architectural Foundation*

The architectural foundation of DB2 for LUW, in place well before the recent enhancements, features a highly parallel, shared nothing database engine with facilities for:

- Cost based query optimization;
- Highly parallel execution of all database operations and featuring data parallelism, query parallelism and pipelining;
- Modular scaling;
- Sophisticated physical database storage structures and access methods featuring partitioning, multi-dimensional clustering, indexing and materialized views;
- A comprehensive implementation of both native relational tables and XML objects;
- Mixed workload management with dynamic throttling; and,
- Advanced input/output for storage, network and interprocess communication.

And, DB2 for LUW has something else that only widespread implementation can provide: product maturity. DB2 has been used on many operational data warehouses in virtually every industry and on every type of data warehouse application—for more than ten years. IBM has been extending and refining its parallel, shared nothing architecture since 1995, making it far more mature, more generalized and more hardened than most other products on the market.

2.1 SHARED NOTHING PARALLELISM

Parallel Database Architecture. One of the key goals of a parallel database architecture is to allow large tasks to be performed quickly. For example, in a clinical research database it is often important to quickly identify all the patients who presented a certain set of symptoms; received a certain treatment; and, then did not improve. This is a typical database query, which has its analogues in every field of work. Such a database query can involve a large and complex search of several related sets of data—in this case, there would usually be separately stored data on patients, symptoms, treatments and tests—in a database.

In the serial architectures of the past, such a search would involve examining each relevant record in a substantial database in sequence. Even with modern hardware, a serial approach would require much more time than would be acceptable in most enterprises.

Parallel architectures shorten the time to perform such tasks by operating on many database records at the same time. In a parallel database architecture, the basic idea is that the query is broken into pieces; the pieces are farmed out to many independently operating units, often called database worker processes; and, because all the workers are operating at the same time, the task is completed more rapidly. Just as 100 human workers can dig a 1000 yard ditch much faster than one worker, so 100 database worker processes can search a database faster than one database worker process.

But, not all parallel architectures are created equal. In particular, as databases increase in scale and complexity—and as the intensity of data analysis increases—some parallel architectures perform better than others.

Shared Nothing Parallel Architecture. In relation to data warehousing and analytical requirements, one of the fundamental strengths of DB2 is its shared nothing parallel

Purpose and Methodology for this Report

This WinterCorp White Paper reviews the architecture of DB 9.7 and its key capabilities for data warehousing.

This White Paper was sponsored by IBM.

In developing this paper, WinterCorp operated as an independent industry expert, interviewing IBM employees; reviewing product documentation; and critically reviewing product design, measurements and evidence in order to arrive at the descriptions and conclusions presented here. IBM was provided an opportunity to comment on the paper with respect to facts. WinterCorp has final editorial control over the content of this publication.

WinterCorp is an independent consultant based in Cambridge, MA, and focused exclusively on large scale data management since its founding in 1992.

architecture. In such an architecture, there are many database worker processes which operate on the data independently and simultaneously. The database is divided into pieces—called partitions—and the data in each partition is accessed by one—and only one—database worker process. The name "shared nothing" comes from the idea that worker processes are not sharing either data or resources. Thus, each can do most of its work with no little or no requirement to coordinate with the others.

This fundamental relationship—which means that database worker processes do not contend with one another for access to data or resources—contributes to scalability for the most common database operations and takes on rising significance as data volumes and workloads continue to grow. The more data you have—and the more work you have to perform on the database—the more important it becomes to have as much independence as possible among the database worker tasks. Now, in all parallel database architectures, there is some coordination of the database worker processes. Even the simplest query must be planned; divided up into units of work called subqueries; and, parceled out among the database worker processes. Execution of the many separate pieces of work must be coordinated. Results of the subqueries must be recombined to deliver a result to the user. All this happens automatically and—to the user—invisibly.

In DB2, this coordination is the responsibility of a coordinator process. DB2 has been designed so that as much work as possible is done in parallel by the worker processes, thus reducing the work for the coordinator process(es). In addition, the DB2 design aims to make coordination and data movement among the worker processes as efficient as possible. This contributes to performance and to the ability to grow the system to handle larger databases, user communities and workloads.

Modular Scaling. In a strong shared nothing architecture, there is no inherent limit on the system capacity. Systems may be configured as clusters of servers and the capacity of the system is expanded incrementally by adding servers. Thus, a data warehouse can start small—with just a few servers—and grow in data volume, usage or other dimensions as requirements grow. A shared nothing architecture generally enhances scalability by minimizing the coordination required among the servers running the database worker processes.

2.2 ADVANCED I/O SUBSYSTEM

From its earliest days, DB2 for LUW has been designed with an advanced I/O architecture, providing high efficiency, high throughput and low latency in communication between main memory and data storage. Some key features of this system are:

- *Asynchronous operation*, so that processing can be overlapped with I/O;
- *Parallel I/O*, so that multiple data records can be read and/or written concurrently by a single process;
- *Multi-block I/O*, also termed "big block I/O" by IBM (that is, many disk I/Os read or write multiple large blocks in a single operation), so that large volumes of data can be read or written in a single operation; and,
- *Read ahead*, so that query processing does not typically wait for input, particularly when large sets of data are required for a single operation.

Recently, IBM has begun supporting Solid State Disk (SSD) in its Smart Analytic Systems, which run DB2. I/O to the SSDs takes advantage of these same techniques.

Similar advanced principles are applied to network I/O, which is used to transfer data and requests among the servers in a cluster. The network I/O processes also operate asynchronously; transferring data in the same large blocks used for storage I/O. In addition, DB2 features sophisticated flow control mechanisms to insure that data flows smoothly among the nodes in performance of complex parallel operations.

2.3 COST BASED OPTIMIZER WITH QUERY REWRITE

In parallel database systems, query optimization is one of the most important issues. Each query must be divided into parallel and serial steps. The ideal query plan minimizes serial operations; balances work among the servers in the cluster; performs the steps of the query plan in the optimal order; and, minimizes interaction and data transfer between the servers.

The space of all possible database queries is immense and present day practice in data warehousing and analytical applications can result in queries of high complexity. When complex queries are performed on large volumes of data, the potential for performance problems due to insufficient query planning is considerable.

IBM has invested in query optimization steadily for decades and has improved the query optimizer in DB2 in every release of the product. First released in 1995, DB2 for LUW has now been used in the field for 15 years in its shared nothing parallel form by customers running increasing complex and demanding data warehouse applications. The query optimizer in DB2 9.7 is thus one of the strongest on the market.

The DB2 LUW query optimizer is cost based and also benefits from query rewrite rules which are used to transform queries prior to optimization, thereby facilitating the rapid creation of better plans than would otherwise result.

In contrast to many less mature products, the DB2 optimizer takes full account of a strong set of physical database structures (described in *Section 2.4*), including indexes, materialized query tables, sophisticated partitioning and a unique DB2 feature named multi-dimensional clusters—all of which can be used as powerful aids to data warehouse query performance. In addition, DB2 has a powerful engine for creating and maintaining statistics on the distribution of data values in the database; on the correlations among data values in sets of columns used together; and, on the selectivity of indexes.

Thus, as well as using classic optimizer techniques, for example, to choose the best order in which to perform a series of table joins, DB2 will frequently decide correctly—based on its statistics and its cost estimating techniques—which of many possible plans will complete a query with good performance. Because of the astronomical number and variety of possible queries—and because of the large number of possible plans for more complex queries—no real optimizer is ever perfect. However, DB2 for LUW has one of the best optimizers on the market—and one that is continuously improving as a result of a large ongoing investment

2.4 ADVANCED PHYSICAL DATA MANAGEMENT

Parallelism is a powerful in data warehousing and is essential today because of large and growing data volumes; high complexity; and, large workloads required to support modern enterprises. Good parallelism and query optimization make it possible for database systems to complete queries, loads and updates in a fraction of the time that would otherwise be required.

Parallel execution gets queries done faster by applying more system resources. But parallelism alone isn't enough for many situations, because it doesn't reduce the amount of work to be performed. If you search one hundred million records—and you do it one hundred ways in parallel—you'll get the answer quickly, but you will still search one hundred million records. And, if you use a specialized hardware architecture to filter the data that is read from disk—say, to reduce the work that must be done in the processor—you will still read one hundred million records from disk.

Because data warehouses continue to grow rapidly, something is needed in addition to parallelism to reduce the work involved in database query. For decades, IBM has invested in research to develop data structures and access techniques that will reduce work and time required to execute common queries—and for the last 15 years, IBM has been building them into DB2 for LUW. The design philosophy has been to look for ways to eliminate I/O—while effectively integrating such measures with parallel operation and advanced optimization. Examples of these features are table partitioning, indexing and materialized query tables. Each of these, and others, are described below.

These features appear to the DBA as physical database design options. When employed, each can be used to facilitate query performance. However, the user need not be aware of their presence. For example, the user writes the query the same way, regardless of whether specific partitioning schemes, indexing or other features described below are in use. It is the optimizer that decides when to use an index or exploit a characteristic of the database partitioning. Such database choices can therefore be changed as the database requirements change without affecting the validity of user queries or applications.

These techniques include:

- *Range and hash partitioning*: Hash partitioning allows data to be spread over the system at the top level to balance work among the units of parallel operation. Range partitioning separates data below the top level by time, date, geography and other natural units of data organization. Query processing then exploits partition elimination to reduce work and time to complete queries. This is a good approach, for example, in databases in which the most recently received data is the focus of a large percentage of queries; in such a situation, only a small portion of the database gets searched for such queries. Range partitioning also increases manageability, facilitating such common needs as the "rolling window" whereby—for example—each week, a new week of data is added to the data warehouse and the oldest week is dropped.
- *Multi-dimensional clustering (MDC)*: a unique feature of DB2 which enables equally efficient retrieval of data along each of several dimensions of equal importance; for example, location oriented applications often index data by both latitude and longitude; these two dimensions are equally important in data retrieval and most clustering or partitioning schemes require that the database designer give greater significance to one of them. Multi-dimensional clustering provides for a design in which the latitude and longitude are equally good for retrieval when used separately; and where their selective power is combined when they are used together.
- *Materialized query tables (MQT)*: a feature whereby selected columns and rows, which may be drawn from multiple related tables and may be aggregated, are automatically replicated by the system to facilitate especially efficient processing of common queries. So, if customer data and sales data are stored in separate tables; and, it is common to ask for summaries of

sales by customer by month, then an MQT can be defined to DB2 for this purpose. DB2 will then automatically maintain summaries by month of purchases by customer. When queries ask for such data, the DB2 optimizer will determine whether time and work can be saved by accessing the MQT rather than the base data. If a savings can be so realized, then DB2 will use the MQT. The existence of the MQT need not be known to, or referenced by, the user query. The MQT is automatically kept up to date by the system as the data changes. DB2 is uniquely capable of maintaining MQTs either via immediate update or deferred update, and of maintaining the integrity of data and the correctness of queries under either option. MQTs can be partitioned differently than the tables on which they are based. They are essentially pre-computed answers to common queries and/or common elements of queries. They can produce a dramatic performance advantage by eliminating the need to repeat common joins and aggregations when queries are performed. MQTs, also known as materialized views in the industry, are present in some form in several data warehouse products; they are completely absent in others. They are an important aid to performance in data warehousing and must almost always be created manually by database designers when they are not an automatic feature of the system. DB2 for LUW has one of the best implementations of this feature in the industry.

- *Indexes*: DB2 supports the most widely used form of database index: the b-tree. DB2 indexes can be defined on any column or combination of columns in a table. When planning a query, the optimizer estimates the selectivity of any available indexes and decides which, if any, to use in performing each query. The optimizer is capable of combining multiple indexesto reduce the I/O required to perform a query. DB2 uses bloom filters and dynamic bitmaps to make such index combinations efficient. DB2 also employs block indexes for multi-dimensional clusters (MDC). Because block indexes are much more compact than row level indexes (by factors of 10, 100 or more), they can be retrieved, written and manipulated with much higher efficiency. The DB2 optimizer is able to exploit this efficiency advantage in queries that include one or more MDC dimensions in selecting data.

These are highlights of the DB2 for LUW database design facilities available to eliminate I/O, reduce work and enhance query performance. Each such technique involves tradeoffs. For example, indexes increase the use of space and increase the time and/or resources required to load or update data. But they accelerate some class of queries. Design choices must be made consciously and bad choices can result in problems. However, IBM provides tools to help with these choices, including intelligent software advisors for selecting indexes and MQTs.

In the opinion of WinterCorp, such physical design options are an important, often critical, capability in a wide range of data warehouse applications and uses.

Some vendors and practitioners argue that such capabilities are no longer important and that users can be successful relying on parallelism, good optimization and powerful hardware only. Such an approach is available to DB2 users, as they are never required to employ any of the advanced physical design features described here. However, with DB2, such facilities are present, and there is a mature optimizer capable of exploiting them automatically. These data structures and access techniques can often spell the difference between success and failure in a data warehouse project because each feature can provide such a large performance benefit in common situations.

These advanced physical data management facilities—integrated with parallelism, optimization and I/O architecture to leverage them—are the consequence of a long term design philosophy and investment direction at IBM. They provide DB2 for LUW with distinctive performance advantages and capabilities. They are enhanced by new features in DB2 9.7, those described in *Section 3*, which represent the continuation of this direction.

2.5 MIXED WORKLOAD MANAGEMENT WITH DYNAMIC THROTTLING

Early data warehouses—many of which were single purpose data marts—typically supported one particular class of work. The most common variations were: (a) a data mart for scheduled batch reporting fed by a weekly batch update; and, (b) a data mart for online query, similarly fed by periodic batch update on the weekend or at night.

Increasingly, today's data warehouses have little or no downtime; are updated frequently or continuously online; and, support frequent or continual concurrent use involving online query; and, batch reporting and analysis. One of the more challenging problems in such a warehouse operation is to satisfy multiple different service level objectives. So, at the same moment, the data warehouse may be servicing:

- Online updates that must be incorporated into the database within five minutes of receipt;
- Short online queries that must be satisfied within two seconds;
- Medium complexity interactive queries that must be satisfied within ten seconds;
- Higher complexity online queries that must be satisfied within sixty seconds;
- Batch jobs, each of which must be completed by a certain time of day; and,
- Deep analytic tasks, each of which must be completed in reasonable time, but without a specific schedule requirement.

In a busy system required to service thousands—or perhaps, hundreds of thousands, or queries per day—this is a massively complex system management challenge. It is well beyond the capacity of any human being to make the individual scheduling and priority decisions involved.

For a data warehouse to satisfy even two or three different service level objectives presents a major challenge, which most data warehouse platforms cannot handle.

What is required are two key capabilities: first, the system must be able to manage the workload under the direction of the responsible person or team; and, second, the system must provide the tools for human managers to understand status; identify problems; diagnose problems; and, correct them in a timely fashion.

In the case of workload management, the person or team involved (with a 7 by 24 system, there is nearly always a team)needs to be able to define priorities and limits to the system and then leave the moment-to-moment decision making to the system—just as a human pilot needs to set the flight plan and leave the small adjustments to the autopilot.

In the case of monitoring, trouble shooting, etc., the people need good tools. With multiple requests (queries, updates, reports, analyses, data mining) in the system most or all of the time—particularly on a system with hundreds or thousands of processors, disks and other components—it is nearly impossible to diagnose problems unaided.

In this area of mixed workload management and monitoring—and, in the broader area of data warehouse administration and management—IBM has been investing seriously for some years now. Previous versions have included workload management facilities and various monitoring tools.

DB2 9.7 provides significant new capabilities in monitoring, metrics, problem diagnosis and workload management. These are discussed further in *Section 5*.

3 Resource Optimization

Today's data warehouses—which manage large data volumes, workloads and user communities—must manage resources efficiently. Without efficient resource management, systems become unaffordable, unresponsive and unable to deliver on service level requirements.

DB2 9.7 contains several significant new or expanded capabilities for resource management including enhanced data compression; enhanced space reclamation; and, a new feature for scan sharing.

3.1 DATA COMPRESSION

It can take a long time to scan a large table. Worse, scan times are lengthening every year, because data volumes are growing much faster than disk I/O bandwidth.

For example, it is common in a data warehouse today to have data tables containing one terabyte (TB), 10 TB or 100 TB of data. Yet the most widely used disk drives read data at about 85 MB/sec. Thus, reading one TB of data requires about 11,764 drive-seconds—meaning that reading one TB of data in parallel on 100 drives would take about 118 seconds, fully occupying all 100 drives; reading ten TB in parallel on 100 drives will take about 1,176 seconds; and, so on. And, doing all the work to read or write terabytes of data also consumes a lot of processor time.

While clever techniques can often reduce the amount of data that needs to be read or written—and while DB2 9.7 makes good use of such techniques—there are still many times every day when a data warehouse platform has to read or write a large volume of data.

Why Compress? One way to reduce the time and system resources involved is, ironically, to employ data compression techniques. If you have to write a TB of data, you might think it takes more total work to compress the data first, because compressing the data requires operating on every byte. But today's processors are so much faster than today's disks, that you can still come out ahead. Up to a point, the more you compress data before writing it to disk, the faster it gets to both read and write the data. With compression, you can hold more data in the buffer pool at any given moment, which increases the cache hit ratio and improves query performance. When backups are compressed, they also can be read and written faster—and they use less space on the backup device.

Before 9.7. DB2 has had a substantial data compression capability since DB2 9.1, whereby data tables are compressed automatically, if specified in the data definition or if requested by the DBA. In the pre-9.7 compression, the system automatically finds repeating data patterns within each data table; replaces these patterns with 12-bit codes; and, maintains a static dictionary within the table for decoding. The repeating patterns consist of a portion of the data in a column; an entire column; or several columns. Because many tables exhibit such repeating patterns (e.g., Sacramento CA 95825 USA or London SW1A 2AA England), the payoff from this compression technique can be quite high. While the payoff clearly varies with the data, compression of a table by 60% to 80% is common, according to IBM. This base capability is still present in DB2 9.7 and is licensed via the optional Storage Optimization Feature for DB2.

New in DB2 9.7. In DB2 9.7, data compression has been enhanced substantially. When the Storage Optimization Feature for DB2 has been licensed, DB2 9.7 will—in addition to all previously available data compression—compress indexes, temporary tables, XML objects and LOBs (large binary objects). In addition, in replication architectures, the Storage Optimization Feature for DB2 will also preserve compression of user data across the replication process.

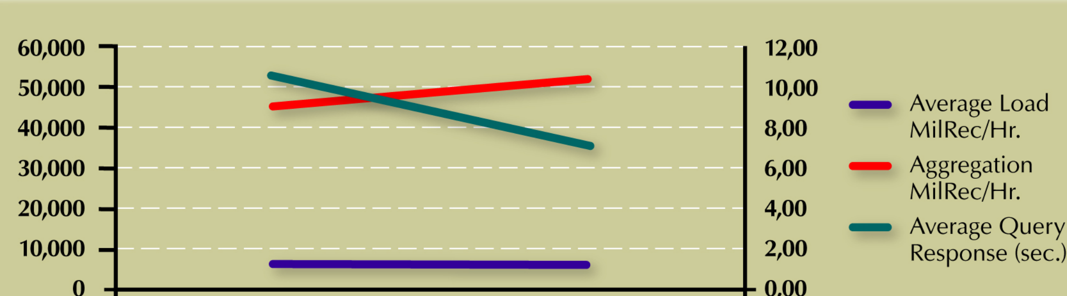
The Storage Optimization Feature for DB2 compresses both of the major components of an index: key prefixes and RID lists. Since these two items account for most of the space in most indexes, the compression effects for indexes can be large. Index compression affects performance in two major ways: it takes less time to read and write an index; and, because the index becomes smaller, it is more often present in the buffer pool—and thus often need not be read from storage at all.

Results and Significance. Temporary tables play a large role in the processing of more complex queries. In many complex queries, disk I/O on temporary tables—which are used to hold intermediate results—is the largest single use of time and system resources. Thus, significant compression of temporary tables delivers a large payoff in efficiency and system performance. Further, the DB2 optimizer determines in each case how much will be saved by compressing the temporary table and employs compression only when the payoff is significant.

To the knowledge of WinterCorp, DB2's automatic compression of temporary tables is, as of the publication date of this paper, unique in the industry. Further, DB2's automatic use of three different compression algorithms on indexes is also unique.

To provide one idea of the impact of index compression, IBM reports that SAP and SAP BW users have experienced index compression ratios of 49% to 73% with DB2 9.7 using the Stoage Optimization Feature for DB2. In one example, an SAP/DB2 customer with a 725GB database measured a total database savings of 68% as a result of compressing both data and indexes. In another case, shown in *Figure 1* above, a user's substantial data warehouse operation measured an overall throughput increase of 15% in execution of a periodic aggregate build step (a key portion of the database update process); and, a decrease of query response time of 23%—both coming as a result of implementing the Storage Optimization Feature for DB2. In this case, the performance benefits came as a result of using all the compression facilities: data compression, index compression and temporary table compression.

Figure 1. Customer Proof-of-Concept Result with the Storage Optimization Feature for DB2 (available with DB2 9.7)



Without Compression	With Compression	
39,13	46,25	Aggregation MilRec/Hr.
5,91	5,65	Average Load MilRec/Hr.
10,58	7,12	Average Query Response (sec.)

(Source: IBM)

3.2 SPACE RECLAMATION

Data warehouse tables that are frequently updated will often come to have unused space within them. DB2 has long had a utility which will reorganize such tables upon DBA request both to increase query efficiency and to reclaim unused space.

With DB2 9.7, the reorganization command has a new option in which the data is not reorganized but unused space within a table—in the form of empty blocks or extents—is reclaimed for new use. With DB2 9.7, a new table space format is provided whereby unused extents can be automatically remapped by the system, saving space. All new table spaces will have this format.

These two enhancements significantly simplify storage space management and increase the efficiency with which storage space is used by the system.

3.3 SCAN SHARING

As noted in *Section 3.1*, scans of much or all of a large table are a fact of life in data warehousing—even in products like DB2 which have significant capabilities for reducing the frequency and duration of large scans. For example, in a scoring or data mining exercise, it might be necessary to operate on most or all of the rows in a large table.

DB2 has long had the capability to perform such scans in a highly parallel fashion—potentially engaging all of the processors and all of the disks concurrently to complete the scan as rapidly as possible.

However, there is sometimes a downside to deploying most or all of the system resources to expedite a single scan of a large volume of data. One important case is when—while the large scan is still running—another large scan of the same data is submitted to the system. This can easily happen in a data warehouse with a large or active user community.

On most data warehouse platforms, one of two things will happen: either (a) the second user will wait (since much or all of the system resources are engaged in the first scan) until the first scan is complete—a wait which could be minutes or hours; or, (b) the two scans will both run at the same time, contending for resources and interfering with one another. In case *b,* both scans are likely to take much longer than either would alone. In case *a*, the second user waits an average of 1.5 times as long for a result as the first user (twice as long if he or she happens to submit the query just as the first scan is beginning).

In DB2 9.7 there is a better way for handling this situation: the second scan will piggyback on the first scan. That is, each data block to be scanned will be read once and fed to both query processes. If the first scan was 10% complete when the second scan began, they will both get all the blocks in the last 90% of the first scan, until the first scan is done. The second scan will then go back and read the first 10% of the table(s) being scanned. The net reduction in storage I/O—compared to executing the queries in sequence—is about 90%. Providing the system has sufficient processor capacity to handle both queries at once, then both users get about the same response time—which is roughly the same as what either user would experience if running on the system alone. As additional scans start, they too can piggyback on any existing scans, so when sufficient resources are available three or more concurrent scans can participate in the I/O sharing process, further increasing the magnitude of I/O reduction.

Scan sharing is supported on a variety of DB2 tables include range and hash partitioned as well as MDC tables.

4 *Flexibility and Growth*

The rapid growth of data warehouses is a widespread phenomenon. This means that data volumes and workloads grow, but it also means that data definitions grow and change. As data finds new uses in the organization, it may be necessary to change the definitions of columns, tables, views, stored procedures and other objects in the database.

And, at the same time this growth and change is occurring in many data warehouses, the dependence of the business operation on the data warehouse is increasing. Increased operational dependence often translates into an increase in the data availability requirement. That is, downtime for the data warehouse becomes less acceptable even as the pace of change is increasing.

In response to this phenomenon, DB2 9.7 has some distinctive new capabilities to support data warehouse growth and schema evolution.

4.1 DATA WAREHOUSE GROWTH

As a data warehouse grows, then—from time to time—capacity must be added to the system. In a parallel, shared nothing architecture, at some point new nodes—servers or modules in the DB2 architecture—must be added to the production data warehouse cluster.

On most data warehouse platforms, adding nodes to the data warehouse cluster requires that: (a) the entire system be stopped; (b) the new nodes connected physically while the system is down; and, then (c) that the system be restarted. On many platforms, any redistribution of the data to the new nodes must be accomplished while the system is down[2]. Each of these steps takes time. During that time, the data in the data warehouse cluster being expanded is not available for use.

With DB2 9.7, the entire data warehouse expansion process now occurs with the system online. One or more servers or modules can be added; with the system online, the new servers or modules can be made visible to the data warehouse; and, the data can be redistributed—a table at a time if desired. Each table is locked as it is redistributed, but the system remains available and able to process queries and updates against any unlocked tables. It is worth noting that clusters running DB2 for LUW can be operated with multiple generations of hardware. Thus, a customer expanding an existing system can add servers or data modules from a new hardware generation while continuing to use servers or data modules from one or more prior generations. Servers or data modules with more capacity manage more data partitions. This is a capability in production use by IBM customers. Unlike many other data warehouse platforms, DB2 enables a strategy in which the customer incorporates new hardware while retiring the old as it becomes most economical to do so This is not a new feature with DB 9.7—it has been in place for years. However, it complements the new features for system expansion described above.

4.2 SCHEMA EVOLUTION

Data warehouses are typically dynamic environments—as dynamic as the business enterprises they support. All the external factors that affect the way a business operates—market conditions,

[2]Some data warehouse platforms have no capability for expanding a system in place, even by taking the system down. In this situation, the customer needs to acquire a complete new system; find physical space for it concurrently with the old system; unload the data from the old system; reload the data on the new, larger system; and, then find a new use for—or resell—the old system.

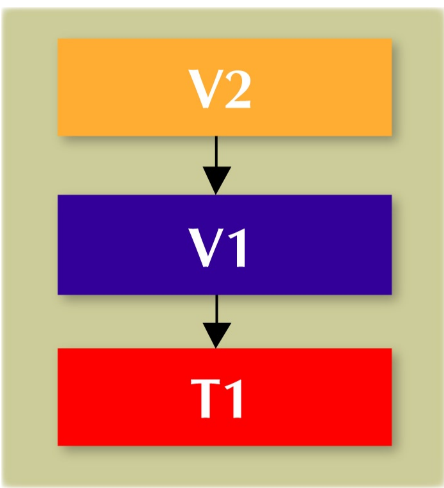
regulations, competition, supply chain dynamics, technology changes—bring about changes to the data in the data warehouse and these changes often affect the schema.

New uses of the data develop weekly or daily, must often be accommodated quickly, and often require change to the schema. New data sources are likewise added frequently, and sometimes provide more reliable or complete data values than existing sources. As the number of database objects (tables, columns, views, etc.) grows into the thousands, schema changes often become a daily activity.

On several data warehouse platforms, most or all schema changes interrupt access to the data. Further, some schema changes can be quite labor intensive. This was the case with DB2 as well, prior to DB2 9.7.

In DB2 9.7, IBM has implemented some key enhancements to decrease both the DBA effort and the disruption associated with schema change. By simplifying the process, these enhancements also reduce the potential for human error in schema change.

Figure 2: V1 and V2 are dependent objects in a data warehouse



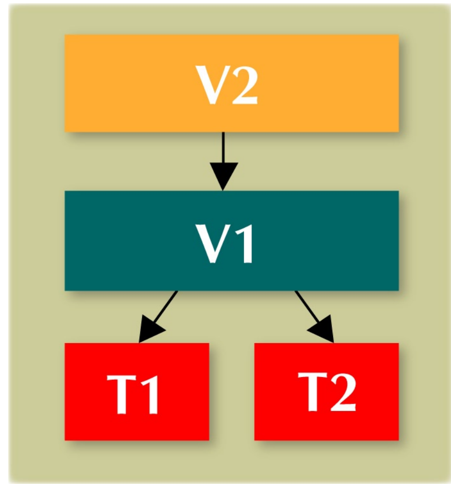
Relaxing the Object Dependency Model. In the typical data warehouse schema, many objects are defined in terms of—and therefore depend on—the definition of other objects. Thus, VIEW definitions depend on either table definitions or other VIEW definitions. This is illustrated in *Figure 2*, in which the VIEW V1 depends on the TABLE T and the VIEW V2 depends on V1.

Now suppose, in response to changing business requirements, it becomes desirable to decompose T1 into two tables: T1 and T2. In the object dependency model of DB2 before DB2 9.7—and on most other data warehouse platforms—changing the definition of T1 would invalidate V1. The DBA would need to delete V1 and redefine it in terms of the new table structure. But the system would not allow the DBA to delete V1 because V2 depends on V1. So, the DBA would have to delete everything that depends on V2; then delete V2; then delete anything depending on V1; and, then, delete V1.

After replacing T1 with the combination of T1 and T2, the DBA would have to do all the work of recreating V1, V2 and any dependent objects. Frequently, one would find that many objects depended on V1 and/or V2—including access control grants, other views and perhaps other objects. While these changes were being made—often with errors along the way—access to the data they define was interrupted.

In DB2 9.7, the object dependency model is relaxed, reducing the DBA effort involved but still maintaining data warehouse integrity. This is accomplished as follows: when the definition of T1 is changed, the system invalidates V1 and V2, but does not delete them. Upon the next access to the data referenced by such a system-invalidated object, the system revalidates the definition. If, for example, the data needed by V1 is now accessible via access to the join of T1 and T2, as indicated in *Figure 3*, the system will appropriately exploit the revalidated view definition. Thus, data access is interrupted only briefly, if at all, and DBA effort is much reduced. V2 has remained intact and unaffected. V1 has been changed only to redefine how it obtains the data to which it refers.

Figure 3: T1 has been replaced by the join of T1 and T2, without the deletion of dependent views



Support of Online Schema Changes. In addition, DB2 9.7 newly supports additional online schema changes. These include:

- DDL to rename a column without deleting it; dependent objects are automatically updated by the system to refer to the new name;
- Support for CREATE OR REPLACE for view, functions, triggers and other database objects. This DDL statement creates the object if it does not presently exists and replaces the definition of the object if the object does already exist; and,
- Support for data type changes via ALTER COLUMN, which is supported for any pair data types that can be used together in a CAST function at runtime.

These additional online capabilities simplify the process for a wide range of schema changes, thereby reducing DBA effort and increasing online data availability.

5 *Workload Management and Monitoring*

Workload management capabilities integrated with the database engine were introduced in DB2 9.5. In DB2 9.7, these capabilities have been extended and significant new monitoring capabilities have been added.

5.1 ADVANCES IN WORKLOAD MANAGEMENT

Once an autopilot for an airplane has been set for the next destination, it automatically makes adjustments for the variations in wind, atmospheric pressure, temperature and other factors that are encountered in flight. The pilot needs to intervene only in the case of unusual conditions.

Workload management in a data warehouse is a similar idea: the administrator defines a set of rules that determine how the system will handle each type of work. The system then runs automatically and self-adjusts to most variations that occur. However, there are times when the administrator must intervene. With the advances in DB2 9.7, workload management can be set up to handle a wider range of conditions automatically. In addition, when exceptions occur, the system can take a wider range of actions to deal with them or assist in the intervention process.

All work entering DB2 9.7 is subject to workload management and automatically classified into default workloads and service classes as follows:

- Workloads, which map connections to service classes by the attributes of the connection (the client user id is an example of such an attribute);
- Service classes, which define the qualities (e.g., priorities) whereby requests will be serviced; and,
- Thresholds, which define the resource limits that are applied (e.g., up to five seconds of processor time per short query before some action is taken).

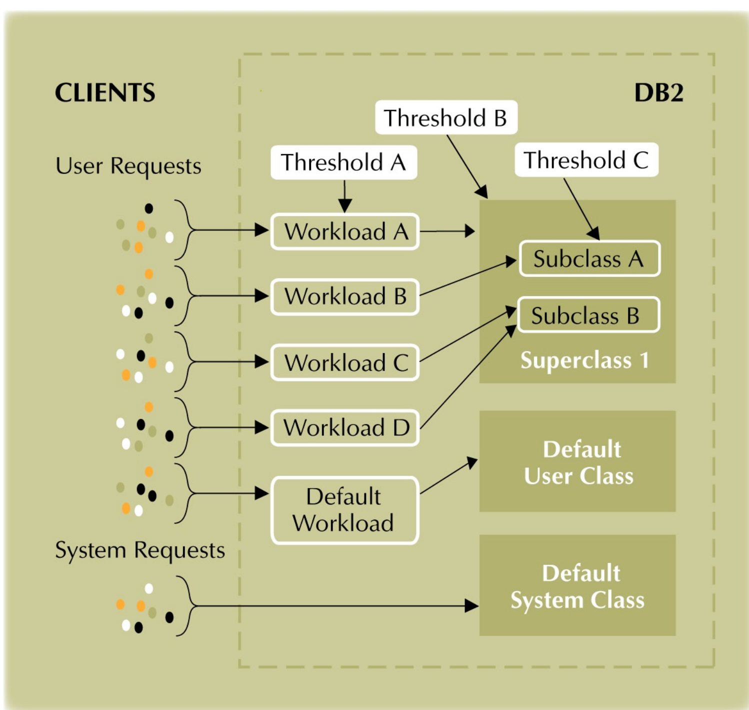
The workload can then be monitored via the default configuration.

Providing the Performance Optimization Feature for DB2 has been licensed (and this feature is standard in some editions of InfoSphere Warehouse 9.7), the administrator can customize the default configuration. Then, using the mechanisms described above, the administrator can define how multiple concurrent workloads—such as database loads, short queries and long queries—can each be serviced by the data warehouse according to different rules. Thus, database loads must get enough resources to stay on schedule (e.g, suppose daily loads must be completed by 4am); short queries must get high priority and be serviced in seconds; and, long queries must get enough resources to be completed in reasonable time, which may range from minutes to hours.

Via the workload management definitions, the administrator can set up appropriate priorities and thresholds so that: system resources are allocated generally in accordance with the user's business needs; and, when all the needs cannot be accommodated at a given moment, some appropriate action is taken. The actions to be taken are defined in connection with the thresholds. For example, a threshold can be set so that if a request submitted as a "short query" consumes more than three seconds of processor time—something that is not expected to happen frequently—then that query has its priority lowered. In addition, a threshold can be defined to record certain information for later analysis so that the reason this "short" query used so much processor time can be understood.

Activity based thresholds, based on such attributes as processor time consumed and rows read, are enhanced in DB2 9.7 and greatly extend the power of workload management. Time-based

Figure 4: Example of a Workload Management Structure



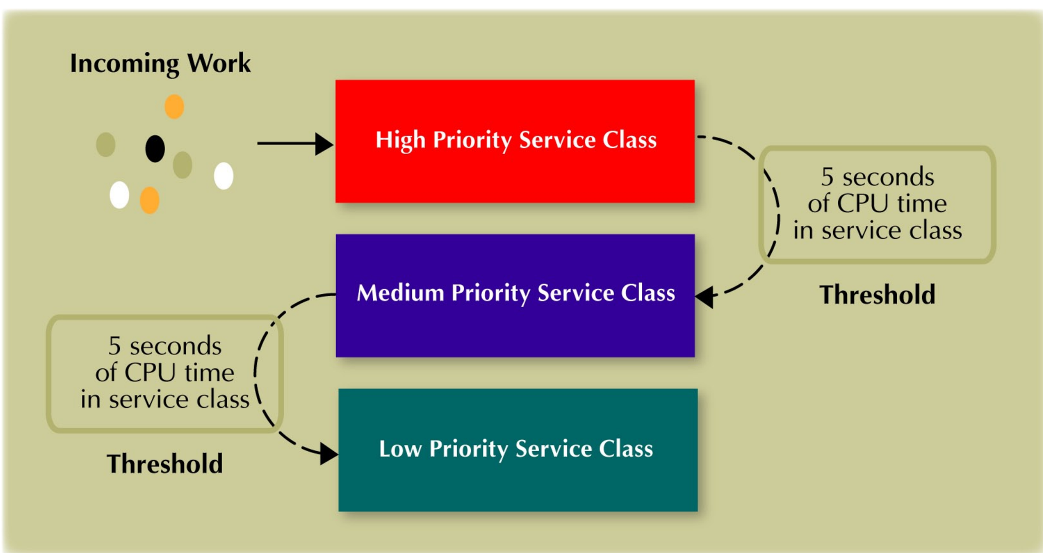
(source: IBM)

thresholds, which were available previously, now support 1-minute check intervals. In addition, the generality with which workloads and service classes can be defined has been extended:

1. The connection attributes that can be used to identify workloads now include IP addresses and wild card characters (in addition to a substantial set of attributes supported previously);
2. Buffer pool I/O priorities can now be associated with service classes;
3. Service classes are now integrated with Linux service classes (integration with AIX service classes was available previously);
4. Thresholds can now be associated with workloads (previously thresholds could be associated with service subclasses and databases).

Priority aging is a scheme whereby the priority of a database request is progressively reduced as it consumes more resources. Thus, lightweight request are performed at a high priority and are generally finished before they can age. Medium weight requests get started at high priority but end up finishing at a lower priority. Heavy weight requests "age" to a low priority and consume the bulk of the needed resources at low priority. Priority aging creates a bias toward light weight requests while still allocating some resources to heavy weight requests. In many situations, this conforms to users natural expectations. Priority aging is readily built into a workload management scheme in DB2 9.7 by the use of thresholds, as shown in *Figure 5*.

Figure 5: Implementation of Priority Aging via Thresholds in DB2 9.7



(source: IBM)

The overall significance of the workload management advances in DB2 9.7 is that more powerful and flexible definitions can now be established; and, that therefore workload management can be implemented and maintained with less effort—and maintain the desired system behaviors over a wide range of conditions. These advances have occurred as a result of continuing product development focus in this area, but also as a result of customer experience actually managing complex workloads and customer suggestions based on that experience.

5.2 ADVANCES IN DATA WAREHOUSE MONITORING

Data warehouses exist to solve a complex data integration problem and automate the processes of bringing the data together, keeping it up to date and delivering it to those who need to use it.

In a typical data warehouse, at any given moment there are many database objects in use; multiple queries and updates in progress; and, frequent automated system operations.

Like an airplane that looks simple and elegant on the outside but may employ a complex design out of sight of the passengers, DB2 generally aims to manage the complexity automatically, so that the users and administrators typically do not have to see or understand the moving parts within.

However, in the typical data warehouse, things can and do go wrong. Some queries take too long or do not finish; updates may fall behind; equipment failures may occur; software errors show up; incorrect data gets loaded into the system; and, people make mistakes—to give just a few examples. Moreover, such events are a daily occurrence in a sufficiently large and complex data warehouse. There is no way to prevent them from happening: the key is to automate and simplify as much as possible the process of identifying; diagnosing; and, correcting system problems when they occur.

IBM reports that one customer's analysis showed that his DBA's were spending 90% of their time troubleshooting problems in data, query performance and other aspects of system operation. This may be an unusually high percentage, but WinterCorp independent research and consulting experience indicate that a large percentage of the staff resources available to a data warehouse operation *are* spent on day-to-day management of operational and performance problems.

After assessing the experience of its customers with data warehouse operation and maintenance, IBM decided some time ago to devote a large investment to manageability. And, some of the major new features of DB2 9.7 are for online monitoring. They are designed to help the administrator detect, diagnose and solve problems.

These monitoring features are a critical complement to DB2's workload management capability. Online monitoring features are needed to understand what the workload is; understand the behavior of the system; assess the workload management set up; and, understand how to improve it.

Included in the new online monitoring capabilities in Optim Performance Manager, Extended Edition are:

- *End-to-end monitoring* to reveal activity, time and resource consumption in not only the database, but also in the network, operating system and application server;
- *A comprehensive set of default metrics*, automatically maintained and aggregated by the system, and available on demand to administrators; as an example, one set of metrics tracks wait time of all sorts and breaks it down into 13 separately measured categories, such as direct I/O wait; buffer pool wait; network wait; and, others;
- *Aggregation of key metrics* automatically at the request, database object, statement, workload and system levels; and,
- *New and expanded information on query plans ("Explain from Section")*, automatically enriched by the system at run time, for example to show actual vs. estimated cardinality of intermediate query results; such plan information is presented in a graphic display, with drill down and other aids available by means of graphical user interface (GUI).

These capabilities have been developed with a conscious focus on the system resources required to collect and maintain the performance data, and a commitment to keep the overhead of data collection well under 3%.

Based on customer experience, IBM believes that a large percentage of system problems—likely to be 80% or more—will be readily diagnosed via the built in metrics and tools.

In WinterCorp's experience, few things are more important than accurate and comprehensive measurement data, in managing the performance and operation of complex systems. You need the data to understand the problems that crop up day-to-day; work out solutions; and, choose the most effective one. The monitoring capabilities IBM has added to DB2 9.7 address the key problems in understanding system behavior and dealing with problematic SQL.

6 *IBM's Smart Analytics System*

As part of its Smarter Systems for a Smarter Planet initiative, IBM has taken a major step forward in providing a range of data warehouse appliances. These are pre-integrated, modular, workload optimized systems for analytics. Called IBM Smart Analytics System these products are available for IBM Power Systems, IBM System *z* and IBM System *x*. IBM DB2 9.7 for Linux, UNIX, and Window is an integral part of IBM Smart Analytics System.

This paper discusses the IBM Smart Analytics System initiative with respect to IBM's 5600 (Intel based) and 7700 (Power 7 chip based) platforms, as well as two new models: the IBM 1050 and 2050 Smart Analytics System

To the customer with a data warehouse or analytic application requirement, IBM Smart Analytics System has considerable significance with respect to acquisition, implementation, deployment, operation, management, growth and support. In particular, the IBM Smart Analytics System products are:

- Engineered with standardized, modular configurations of hardware and software that comprise a complete system stack for the intended use;
- Installed, integrated, configured, tested and performance tuned in IBM factories and labs, relieving the customer of these chores for typical uses; and,
- Packaged, delivered and priced as a single unit, complete with all hardware, software, documentation; and, support. Support includes integrated support for the entire solution, with a single contact for all the products included and an annual health check by a IBM provided DBA with performance expertise.

This approach reduces time and cost for the customer to in progressing from the new data warehouse requirement to a deployed, operational data warehouse that is harvesting business benefits.

Each system consists of one or more modules, each of which is a complete, integrated unit of hardware and software. Systems are built up from:

- One or more foundation modules, which provides the DB2 coordinator function
- One or more data modules, to add data capacity;
- Zero or more user modules, to add workload capacity;
- Zero or more application modules: to add application software, tools and analytical capacity (e.g., Cognos); and,
- Zero or more failover modules, to provide standby capacity during node failures.

Modules can be added individually or in groups to expand the capacity of the system.

Also, to simplify communication concerning the configurations expected to be most widely used, IBM has also introduced a remarkably simple sizing scheme in which the six most common data warehouse configurations are designated simply with sizes ranging from extra small (XS) to extra-extra-large (XXL).

As an example, for the Smart Analytics 7700, capacities are as shown in *Table 1*.

Table 1: IBM Smart Analytics 7700, Standard Sizes

Size	Number of Data Modules	Spinning Disk (TB)[3]	SSD Capacity (TB) [4]	Estimated User Data Capacity[5]
XS	1	29	.7/4	29
S	2	58	1.4/8	58
M	3	86	2.1/13	86
L	6	173	4.2/25	172
XL	10	280	7/40	288
XXL	20	560	14/84	575

(source: IBM)

Each data module is a Power 740 server with 16 cores, 128GB memory, 96 300GB disk drives. Compared to the prior model, using the Power6 processor, the new data module features 4 times as many cores; 4 times the memory; 3 times the number of disks[6]; and, a 65% reduction in space required. The interconnect increases in capacity from 1Gb to 10Gb Ethernet.

A 7700 data module comes standard with 700 GB of solid state device (SSD) capacity—1.4 TB of SSD for the "small" system—which is used automatically by DB2 9.7 for temporary work space, thus accelerating the performance of most queries substantially and increasing the throughput of the system. In addition each system can optionally be configured with more SSD—up to 8 TB for the small system—for additional performance or I/O throughput. With the standard 7700 module configuration, I/O capacity is now 6.4 GB/sec.

Clearly, some customers with a 58 TB data warehouse requirement may need more or less than the standard "small" configuration. However, IBM's IBM Smart Analytics System scheme provides for this, by enabling the customer to vary the number of each type of module independently. So, for example, a customer with many concurrent users may add more "user modules."

For the Smart Analytics 5600, standard capacities are as shown in *Table 2*.

Each data module is an IBM System x3650 server with 4 cores, 32GB memory, 24 300GB disk drives.

In addition, the Smart Analytics 5600s is available in configurations including solid state disk. For the Smart Analytics 5600s, standard capacities are as shown in *Table 3*.

Each 5600s data module is an IBM x3650 with 8 cores, 64 GB memory, 24 450GB disk drives and 650GB of solid state disk.

[3]Rounded by WinterCorp to the nearest TB.

[4]Standard/Maximum (rounded by WinterCorp to the nearest TB). This is the IBM Blue Darter SSD.

[5]At 55% of available space, with RAID6, estimated data compression of 2.5x, rounded to the nearest TB.

Table 2: IBM Smart Analytics 5600, Standard Sizes

Size	Number of Data Modules	Spinning Disk (TB)[7]	Estimated User Data Capacity[8]
XS	2	14	12
S	4	29	24
M	6	43	36
L	8	58	48

(source: IBM)

Table 3: IBM Smart Analytics 5600s, Standard Sizes

Size	Number of Data Modules	Spinning Disk (TB)[9]	Solid State Disk (TB)[10]	Estimated User Data Capacity[11]
XS	2	22	1.3	18
S	4	43	2.6	36
M	6	65	3.8	54
L	8	86	5.1	72

(source: IBM)

The Smart Analytics System consists of both a hardware platform and the InfoSphere Warehouse software. There are three versions of the software depending on the configuration you choose for your hardware platform, as shown in *Table 4*.

WinterCorp believes that many customers begin a data warehouse effort with considerable uncertainty around the workload requirements. In this situation, it may make sense to order a standard configuration and adjust as more information becomes available. Additional modules can be added; extra modules can be redeployed in another analytical system. Though the scenario will vary with the situation, the standardization and integration provided here will simplify planning, acquisition and deployment in many cases.

IBM has also introduced two single server configurations: the IBM Smart Analytics System 1050 and 2050. These are available with SUSE Linux 11 or Windows Server 2008. Both are available with either InfoSphere Warehouse 9.7 Departmental Base Edition or Departmental Edition (these include DB2 9.7 with somewhat reduced data warehouse capabilities) and optionally include

[6]One might imagine that 4x the number of disks would be needed in the 7700 to maintain balance among the resources. 3x is sufficient because of the incorporation of solid-state disk as a standard feature also adds storage I/O capacity.

[7]Rounded by WinterCorp to the nearest TB

[8]At 55% of available space, with RAID6, estimated data compression of 2.5x, rounded to the nearest TB

[9]Rounded by WinterCorp to the nearest TB

[10]Rounded by WinterCorp to the nearest 100GB (0.1TB)

[11]At 55% of available space, with RAID6, estimated data compression of 2.5x, rounded to the nearest TB

Table 4: Infosphere Warehouse 9.7 Feature Matrix

	Departmental Base	Departmental	Enterprise
Smart Analytics System Model Availability	**1050 / 2050**	**1050 / 2050**	**5600 / 7600 / 7700**
DB2 Server Foundation (DB2 Enterprise Server Edition)	Included	Included	Included
Database Partitioning Scalability	Included	Included	Included
Unlimited Database Size	Included	Included	Included
Modeling and Design (Design Studio)	Included	Included	Included
Embedded Data Movement and Transformation (SQL Warehousing Tool)	Included	Included	Included
Administration and Control (Administration Console)	Included	Included	Included
Online Analytical Processing (Cubing Services)	Included	Included	Included
Federated Data Access (InfoSphere Federation Server Relational Wrappers)	Included	Included	Included
IBM Cognos 8 BI Starter Edition	Included	Included	Included
Data Mining and Visualization (Intelligent Miner, MiningBlox)	—	Included	Included
Text Analytics	—	Included	Included
DB2 Alphablox including BloxBuilder	—	Included	Included
DB2 Alphablox Connectors	—	—	Included
Integrated Workload Management (DB2 Workload Management)	—	Included	Included
Performance Analysis and Tuning (Optim Performance Manager Extended Edition, DB2 Performance Expert)	—	—	Included
Deep Compression (Data, Index, Temp. Table, XML)	—	—	Included

(source: IBM)

Cognos 8 BI Reporting. The 1050 can be configured for 330 GB to 3.3 TB of user data. The 2050 can be configured for 3.3 TB to 13.2 TB of user data.

While IBM has been moving down this path for several years, offering progressively simpler configuration, pricing and packaging options for data warehousing (e.g., with the balanced configuration unit, the balanced warehouse and others), the IBM Smart Analytics System offerings are another significant step forward.

7 *Conclusions*

DB2 is a mature software product for data warehousing with comprehensive capabilities and outstanding performance and scalability. DB2 9.7 is built on a product foundation featuring:

- A shared nothing parallel database architecture providing performance, scalability and data availability;
- A long term focus on system efficiency, based on extensive capabilities for physical database design, I/O performance and cost based query optimization;
- Mixed workload management, providing the capability to satisfy multiple service level objectives on a single sytem managing a single logical copy of the data; and,
- Data compression,

along with many other features.

In recent years, IBM has continued to invest in DB2 to address emerging customer requirements and keep pace with the extraordinary increases in scale and rising complexity of challenging data warehouse applications. In DB2 9.7, the latest release, IBM has expanded the functionality of DB2 to address additional needs in mixed workload management; in resource optimization—including advanced data compression features; in system performance management; in schema evolution; and, other areas described in this paper. These new capabilities in DB2 9.7 provide the customer with increased manageability; better performance; enhanced cost performance; and, agility.

InfoSphere Warehouse 9.7 delivers these DB2 9.7 capabilities in the form of a software only product family for data warehousing.

IBM's Smart Analytics System integrates DB2 with a full stack of system software; with business intelligence and analytics software; and, with hardware—to provide customers with a complete data warehouse appliance deliverable in standardized, pre-installed, pre-tested configurations. IBM has recently announced new versions of its Smart Analytics Systems, which provide significant advances in performance and throughput, while requiring less space and incorporating new hardware features including Solid State Disk. In addition, IBM has introduced two new, single server configurations to provide compact, low price entry points to the product line. Thus, the Smart Analytics Systems greatly increase the simplicity, advance the performance and enhance the economics of acquiring and deploying DB2's data warehouse capabilities.

Overall, DB2 9.7, InfoSphere Warehouse 9.7 and IBM's Smart Analytics Systems are among the most capable data warehouse products available on the market today.

In the opinion of WinterCorp, organizations that need a platform for data warehousing and/or data analytics will want to consider the IBM platforms described in this paper.

WinterCorp is an independent consulting firm that specializes in the performance and scalability of terabyte- and petabyte-scale data management systems throughout their lifecycle.

Since our inception in 1992, we have architected many of the world's largest and most challenging databases in production today. Our consulting services help organizations define business-critical database solutions, select their platforms, engineer their implementations, and manage their growth to optimize business value.

With decades of experience in large-scale database implementations and in-depth knowledge of database products, we deliver unmatched insight into the issues that impede performance and the technologies that enable success.

WinterCorp

245 FIRST STREET, SUITE 1800
CAMBRIDGE MA 02145
617-695-1800

visit us at www.wintercorp.com