DB2 10 for z/OS

A Smarter Database for a Smarter Planet

Julian Stuhler
Triton Consulting
October 2010

Contents

Abstract

This paper will provide a high-level overview of the major new features of IBM DB2 10 for z/OS from an IT Executive's perspective, with the emphasis on the underlying business value that the new release can deliver.

About The Author

Julian Stuhler is a Principal Consultant with Triton Consulting, a UK-based company specialising in the provision of DB2 consultancy, education, software and managed services to clients throughout Europe. He has over 20 years relational database experience, working for a number of clients within the insurance, telecommunications, banking, financial services and manufacturing sectors.

Julian has lectured widely on DB2 subjects, both in the UK and Europe, and won the "Best Overall Speaker" award at the 2000 International DB2 User Group European Conference. He is a co-author of an IBM Redbook on Java Stored Procedures, and a frequent contributor to industry publications such as IDUG Solutions Journal and Database Journal.

Julian is an IBM DB2 Gold Consultant, Past-President of the Board of Directors for the International DB2 User Group and an IBM data Champion. He can be contacted at **julian.stuhler@triton.co.uk**.

Executive Summary

DB2 10 for z/OS is the latest release of IBM's flagship database. This paper provides a high-level overview of the major new features from an IT Executive's perspective, with the emphasis on the underlying business value that DB2 10 can deliver.

Business Benefit Summary

DB2 10 delivers a number of significant business benefits, many of which are exploitable "out of the box" with little or no database, application or system changes. These can be summarised as follows:

- **CPU Reductions**. DB2 includes a raft of enhancements aimed at improving application performance and reducing CPU usage. Most customers can expect to see net CPU savings of 5-10% in their traditional DB2 workload when compared to DB2 9, without any application changes being required. Significant additional savings are possible for other specific workloads, and with some application changes.

- **Scalability Improvements**. DB2 10 delivers a spectacular increase in the number of threads that can be supported by a single subsystem – most customers will be able to achieve 5-10 times the number of concurrent connections compared to DB2 9. This will allow many customers to reduce the number of DB2 members needed to support their workloads, resulting in net CPU and memory savings and improving application performance.

- **Productivity Enhancements**. New features such as temporal tables, automated statistics and improved dynamic schema change reduce the effort required by developers and support staff to deliver robust DB2 applications.

DB2 10 delivers significant "out of the box" benefits that many customers will be able to exploit with little or no additional effort. These include the most aggressive performance and CPU improvements of any DB2 release in the last 20 years, scalability enhancements to support ever-increasing workloads and productivity improvements to allow DB2 developers and support staff to do respond more rapidly to the demands of the business.

Conclusion

Even in the most favourable economic climate, businesses need to control costs and increase efficiency in order to improve their bottom line. In today's more challenging business environment this has become a key factor for the survival and success of enterprises of all sizes.

DB2 10 delivers significant "out of the box" benefits that many customers will be able to exploit with little or no additional effort. These include the most aggressive performance and CPU improvements of any DB2 release in the last 20 years, scalability enhancements to support ever-increasing workloads and productivity improvements to allow DB2 developers and support staff to do respond more rapidly to the demands of the business.

Collectively, these features deliver real and quantifiable business benefit, and many customers will be considering upgrading to DB2 10 much more quickly than they may have done for previous releases.

Introduction

You don't have to be an IT professional to see that the world around us is getting smarter. Everywhere you look our environment is getting more connected and "instrumented", and clever technologies are being adopted to use the resulting real-time data to make things safer, quicker and greener. While this explosion in machine-generated data is happening, human beings themselves are also generating vastly more content than ever before. Today, people and machines together create new data at an astounding rate: more data will be created over the next four years than in the entire history of the planet.

From banking to transportation to healthcare, DB2 for z/OS sits at the heart of many of the IT systems needed to drive a Smarter Planet, and has an important role to play in supporting the transformation.

Building a smarter planet is going to need smarter IT systems, which in turn will depend upon the availability of a robust, efficient and secure way of storing, retrieving and analysing this vast amount of data. From banking to transportation to healthcare, DB2 for z/OS sits at the heart of many of the IT systems needed to drive a Smarter Planet[1], and has an important role to play in supporting the transformation.

In the meantime, the global economic climate remains challenging and DB2 for z/OS customers around the world are still trying to gain competitive advantage by doing more with less: more business insight, more performance, more operational efficiency, more functionality, more productivity with less cost, quicker time to market and a lower TCO.

This paper will provide a high-level overview of the major new features of DB2 10 from an IT Executive's perspective, with the emphasis on the underlying business value that DB2 10 can deliver.

DB2 10 for z/OS is the latest release of IBM's flagship database, and seeks to address these and other challenges. A wealth of material exists on the technical changes within DB2 10, but finding descriptions of how those new features will improve your business results can be a challenge. This paper will provide a high-level overview of the major new features from an IT Executive's perspective, with the emphasis on the underlying business value that DB2 10 can deliver.

In the meantime, many customers are still running DB2 for z/OS Version 8 (or earlier releases) and need to understand how DB2 9 can help their organisation. A brief summary of the business benefits offered by DB2 9 is provided in Appendix A – DB2 9 Review on page 45 of this document.

[1] See http://www.ibm.com/smarterplanet for more details of IBM's Smarter Planet initiative.

DB2 10 – A Smarter Database

In this section, we'll take a detailed look at the major features of DB2 10, and how many of IBM's most innovative enterprise customers are intending to use them to deliver an enhanced IT service to the business.

Many DB2 10 enhancements can be used "out of the box" with little or no effort required to begin exploiting them, reducing the time to value for an upgrade.

Many of these enhancements can be used "out of the box" with little or no effort required to begin exploiting them, reducing the time to value for a DB2 10 upgrade. Please refer to Appendix B – DB2 10 New Features by Implementation Effort on page 50 for a breakdown of the effort required to exploit each new feature.

This section is organised around the key DB2 10 themes:

- **Efficiency** – reducing cost and improving productivity
- **Resilience** – improving availability and data security
- **Growth** – supporting new and expanding workloads
- **Business Analytics** – enhanced query and reporting

> ***"DB2 10 enhances our ability to support our rapidly growing workloads while delivering some very valuable new function with immediate business benefits."***
>
> **Paulo Sahadi, Senior Production Manager, Information Management Division, Banco do Brasil**

Efficiency

Even in the most favourable economic climate, businesses need to control costs and increase efficiency in order to improve their bottom line. In today's more challenging business environment this has become a key factor for the survival and success of enterprises of all sizes.

This section examines the major DB2 10 enhancements that are aimed at improving the efficiency of the IT systems that rely on DB2: a key design objective for the new release. These features can help to reduce ongoing operational costs, improve developer and DBA productivity and enhance the customer's experience by increasing performance and delivering a more responsive application.

> ***Continuous availability, reduced performance cost and future growth with constraints are of paramount importance to our business. We are really excited about the potential of DB2 10 for z/OS to help us achieve our goals in each of these areas. Our high expectation is the reason why Danske Bank will invest a lot of effort in the Beta program."***
>
> **Jan Michael Christensen,**
> **Vice President, Danske Bank**

CPU Reductions

Most DB2 for z/OS customers operate on a CPU usage-based charging model, so increases or decreases in the amount of CPU required to run DB2 applications can have a direct and significant impact on overall operational costs. Traditionally, IBM has tried to limit the additional CPU cost of adding new functionality into each release, keeping the net CPU impact below 5%.

The move to a 64-bit computing platform in DB2 for z/OS Version 8 was an exception to this rule, and introduced some significant processing overheads which resulted in many customers experiencing net CPU increases of 5-10% following the upgrade. DB2 9[2] helped to redress the balance somewhat by delivering modest CPU improvements for many large customers, but IBM was determined to deliver more significant cost reductions in DB2 10.

One of the fundamental design objectives of DB2 10 was to deliver a 5-10% CPU reduction "out of the box", with little or no change

[2] Please see Appendix A – DB2 9 Review on page 13 for a summary of the DB2 9 business benefits.

being required to applications and further savings being possible with some database and/or application changes. Figure 1 below shows a pictorial representation of the typical CPU decrease seen in each release since V3.

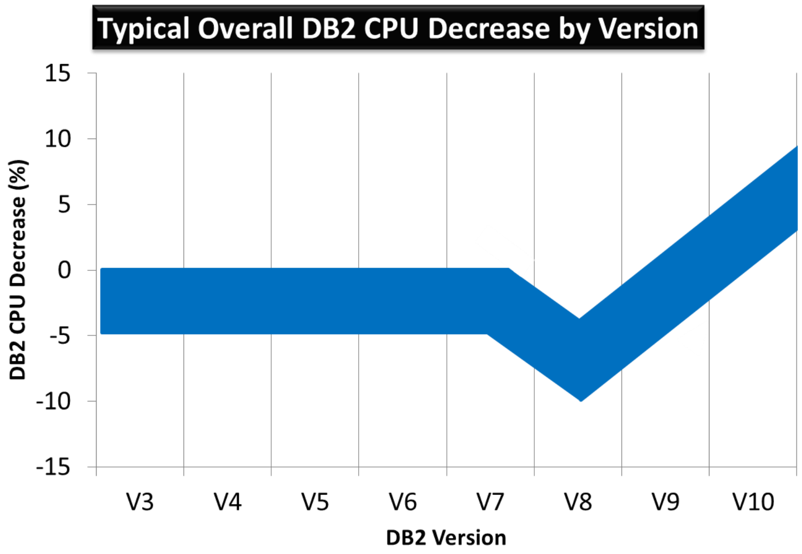


Figure 1 – Typical Overall CPU Decrease by Version

Based on IBM labs tests and some early beta customer experiences, IBM has delivered the most aggressive performance improvements of any DB2 release in the last 20 years.

Most customers can expect to see CPU savings of 5-10% in their traditional DB2 workload without any application changes being required.

Based on IBM labs tests and some early beta customer experiences, IBM has exceeded this objective and delivered the most aggressive performance improvements of any DB2 release in the last 20 years. As many of these improvements are down to internal DB2 code optimisation and exploitation of the latest System z hardware instructions, most customers can expect to see CPU savings of 5-10% in their traditional DB2 workload without any application changes[3] being required.

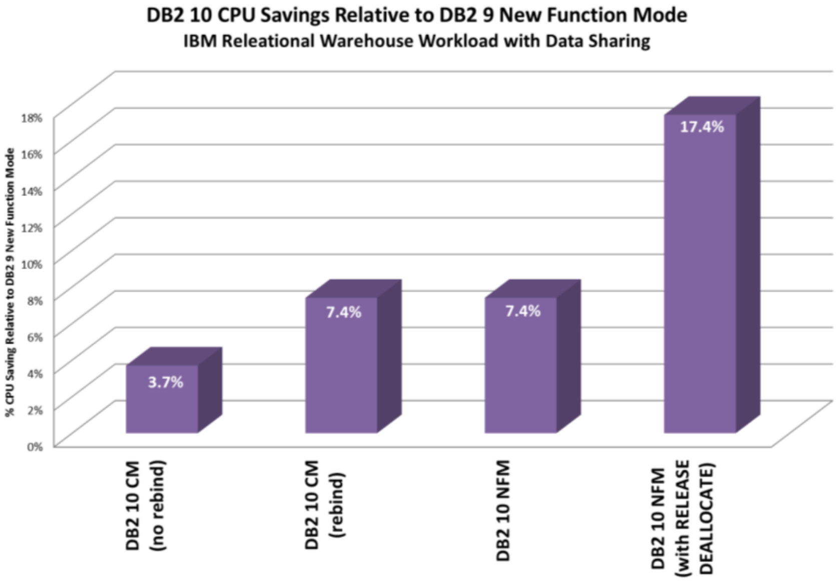


Figure 2 – Sample CPU Savings using IRWW

[3] In order to benefit from improvements in DB2's ability to select the most efficient access path, a "rebind" will usually be required to allow DB2 to re-create the access path structures for an application. This does not require any changes to the application.

Figure 2 above shows the savings relative to DB2 9 that were achieved in internal IBM testing using the standard IBM Relational Warehouse Workload (IRWW). The first column shows a 3.7% CPU saving immediately following migration in DB2 10 Compatibility Mode (CM). The net saving increased to 7.4% following a REBIND of the affected packages with the same access path, and this remained the same when the system was placed in New Function Mode ((NFM). Finally, a net saving of 17.4% was measured once the packages had been rebound to use the new RELEASE protocols described in the section on Other Efficiency Enhancements on page 22.

Savings of 5-40% have been observed in the labs running workloads with heavy concurrent insert activity.

Customers running the following types of workload can expect even bigger CPU savings:

- Workloads previously constrained due to a lack of virtual storage in DB2 Version 8 or Version 9.
- Distributed applications connecting to DB2 via the DRDA protocol (e.g. SAP).
- Workloads using native SQL stored procedures. Efficiency enhancements with commonly-used functions[4] have shown CPU reductions of up to 20% during initial IBM labs testing.
- Workloads with heavy concurrent insert activity, especially sequential insert where rows are inserted sequentially where savings of 5-40% have been observed in the labs.
- Complex query workloads, where up to 20% CPU reduction has been observed with no change to the access path. Greater savings are possible where a more efficient access path is selected.

Up to 20% CPU reduction has been observed for complex query workloads, where with no change to the access path. Greater savings are possible where a more efficient access path is selected.

All of these figures assume an upgrade from DB2 9 to DB2 10. For those customers considering a move directly from DB2 Version 8 to DB2 10[5], the net impact could be even bigger.

The potential CPU savings made possible by DB2 10 are likely to be the single biggest factor in driving customers to upgrade to the new release – especially as many of the savings can be realised very quickly after the upgrade, and with little or no application changes

Most of the performance measurements available at the time of writing are based in internal IBM lab workloads but early indications from beta customers also show CPU savings in line with the lab tests. The potential CPU savings made possible by DB2 10 are likely to be the single biggest factor in driving customers to upgrade to the new release – especially as many of the savings can be realised very quickly after the upgrade, and with little or no application changes.

Temporal Tables

Many IT systems need to keep some form of historical information in addition to the current status for a given business object. For example, a financial institution may need to retain the previous addresses of a customer as well as the one they are currently living at, and know what address applied at any given time. Equally, an

[4] Optimised functions include IF and SET statement processing and access to the SYSDUMMY1 table. Exploiting these enhancements requires the stored procedures to be regenerated (or dropped and recreated) but no application change.

[5] When upgrading to DB2 10, IBM supports customers moving either from DB2 9 or DB2 Version 8 via a "skip migration" process. Please see Upgrading to DB2 10 on page 39 for further details.

insurance company may need to know what level of cover was in place two months ago when a claim was made. Previously, these kinds of requirements would have required the DBA and application developers to spend valuable time creating and testing the code and associated database design to support the historical perspective, while minimising any performance impact.

Many IT systems need to keep some form of historical information in addition to the current status for a given business object. The new temporal data support in DB2 10 provides this functionality as part of the core database engine.

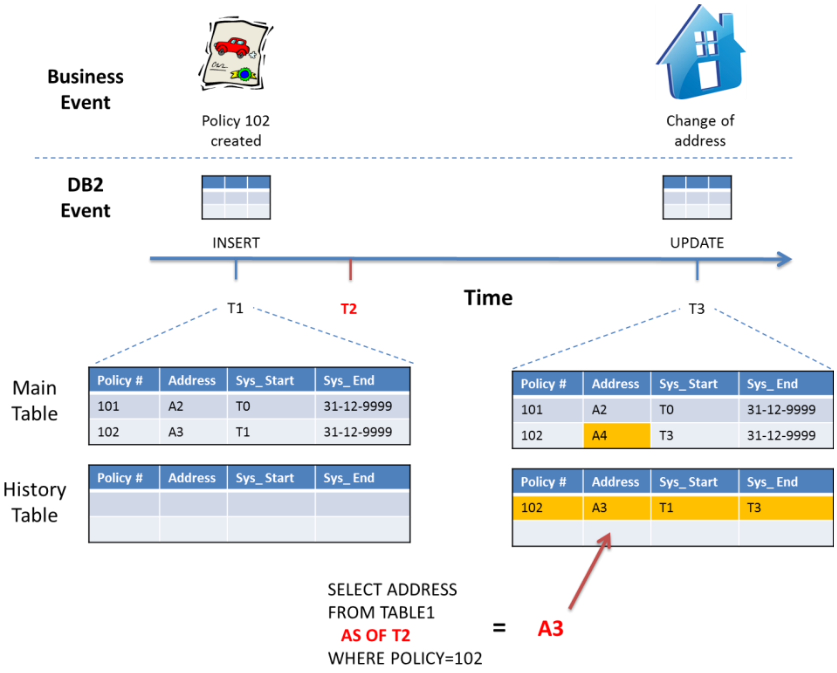
The new temporal data support in DB2 10 provides this functionality as part of the core database engine. The DBA indicates which tables/columns require temporal support when they are created, and DB2 will automatically maintain the history whenever an update is made to the data. Elegant SQL support allows the developer to query the database with an “as of” date, which will return the information that was current at the specified time.

> ***“The new temporal functionality in DB2 10 for z/OS will allow us to drastically simplify our date-related queries. In addition, we’ll be able to reduce our storage costs by using cheaper storage for inactive rows and reduce our processing cost by having DB2 handle data movement more efficiently than the custom code we’ve written to do the same work in the past.”***
>
> ***DB2 10 Beta Customer***

As shown in Figure 3 below, DB2 maintains a separate “history table” for updated rows in a temporal table. This is completely transparent to the developer, who codes SQL against the main table as usual. When a row is updated (as shown at time T3 in the diagram) DB2 will store a version of the old row in the history table before updating the current row in the main table. Similarly, when a row is deleted it is first copied to the history table before being removed from the main table. DB2 maintains system timestamps (the SYS_START and SYS_END columns shown) to record the period during which a given version of the row was current.

> ***"We are really thrilled about the Temporal Data feature – this has the potential to significantly reduce overheads. We have estimated that 80% of our existing applications could have used the V10 temporal features instead of application code - this will drastically save developer time, testing time – and even more importantly make applications easier to understand so improve business efficiency and effectiveness."***
>
> ***Frank Petersen, DB2 Systems Programmer, Bankdata***

Finally, the new “AS OF” clause in SQL SELECT statements allow the developer to see the data as it was at a given point of time. In the example, the policy information at time T2 is required, which will return the original address (A3) instead of the current address (A4).



Main Table (T1):

Policy #	Address	Sys_ Start	Sys_ End
101	A2	T0	31-12-9999
102	A3	T1	31-12-9999

History Table (T1):

Policy #	Address	Sys_ Start	Sys_ End

Main Table (T3):

Policy #	Address	Sys_ Start	Sys_ End
101	A2	T0	31-12-9999
102	A4	T3	31-12-9999

History Table (T3):

Policy #	Address	Sys_ Start	Sys_ End
102	A3	T1	T3

```
SELECT ADDRESS
FROM TABLE1
  AS OF T2
WHERE POLICY=102
```

Figure 3 – DB2 Temporal Data Concepts

With so many IT systems needing to accommodate a historical perspective and maintain audit logs of changes made to sensitive data, DB2’s new temporal support promises to save many hundreds of hours of design, coding and testing that would otherwise be required to build this function manually for each application. While

With so many IT systems needing to accommodate a historical perspective and maintain audit logs of changes made to sensitive data, DB2’s new temporal support promises to save many hundreds of hours of design, coding and testing that would otherwise be required to build this function manually for each application.

the benefit for existing applications is limited, this feature promises to deliver major productivity savings for new developments.

Improved Scalability

The valuable scalability enhancements within DB2 10 are described in the section on Increased Scalability Though Full 64-bit Exploitation on page 30. In addition to supporting workload growth and providing more flexibility, these enhancements can also deliver some significant performance benefits, as follows:

- **Reduction in data sharing overhead**. The virtual storage constraints within previous releases of DB2 imposed a practical limit of 400-500 concurrent active threads[6] within a single DB2 subsystem. As a result, many DB2 data sharing[7] customers were forced to use more DB2 members than otherwise necessary in order to support their workloads. Although DB2's industry-leading data sharing architecture minimises the processing overheads, each additional member will impact overall performance and resource usage.

  Figure 4 below shows a typical scenario for a SAP environment. In this example, a data sharing group consisting of four DB2 members is used in order to support 1,600 concurrent threads from four SAP application servers.

> ***"With the scalability improvements in DB2 10, we expect to be able to quickly reduce our production data sharing group from 20 members to 15. We will also save some CPU and storage from removing the five DB2 systems, and we will have to spend a lot less time monitoring our virtual storage."***
>
> **Paulo Sahadi, Senior Production Manager, Information Management Division, Banco do Brasil**

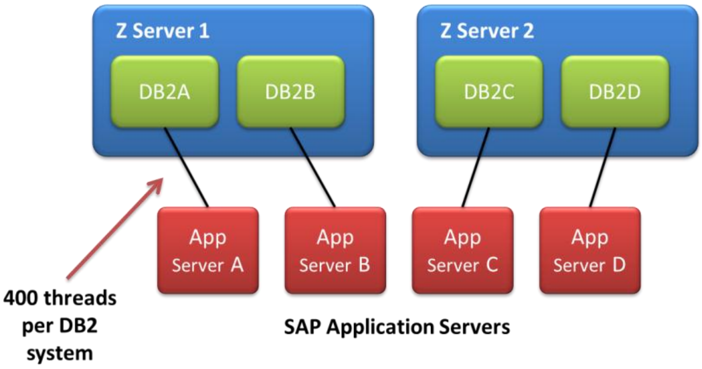


Figure 4 – Typical SAP Data Sharing Configuration

DB2 10 introduces some dramatic scalability improvements (fully described on page 30) that allow each system to handle 5 to 10 times the current number threads. This will allow many customers to reduce the number of DB2 members needed to support their workloads, resulting in net CPU and memory savings and improving application performance.

This is illustrated by Figure 5 below, which shows that the same 1,600 thread workload can be handled by just two DB2 subsystems, with significant scope for additional workload growth (initial SAP benchmarks show 2,500 threads per DB2 system is sustainable). Productivity

[6] An "active thread" is a connection to DB2 that is actively working on behalf of an application requestor. The maximum number of threads supported by a single DB2 system is dependent on the nature of the workload.

[7] Data sharing is an optional DB2 facility that allows multiple DB2 systems (known as "members") to access a single shared copy of the data. It is usually implemented to improve availability, as the loss of any single DB2 system allows processing to continue on the others.

savings are also possible due to the reduced requirement to closely monitor available storage.

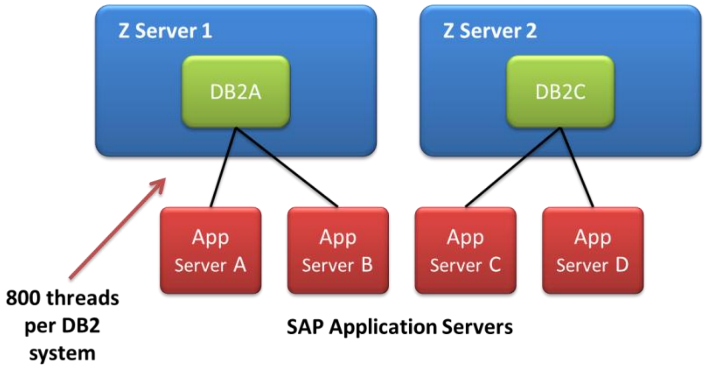


Figure 5 – Potential DB2 10 SAP Data Sharing Configuration

- **Improved dynamic statement caching**. With the growing popularity of running Java and ERP workloads such as SAP on DB2 for z/OS, dynamic SQL[8] is becoming more and more prevalent. DB2 allows dynamic SQL statements to be cached in order to avoid most of the overheads usually associated with executing dynamic SQL, but the size of this cache was limited in previous releases due to the same virtual storage constraints described above. This in turn limited the effectiveness of the cache.

  The virtual storage constraint relief delivered in DB2 10 will allow most customers to dramatically increase the size of the dynamic statement cache, thereby allowing a greater proportion of their dynamic SQL to be cached and reducing CPU and elapsed times for these queries. Please refer to page 15 for some other important enhancements that will improve dynamic statement caching in DB2 10.

The virtual storage constraint relief delivered in DB2 10 will allow most customers to dramatically increase the size of the dynamic statement cache, thereby allowing a greater proportion of their dynamic SQL to be cached and reducing CPU and elapsed times for these queries.

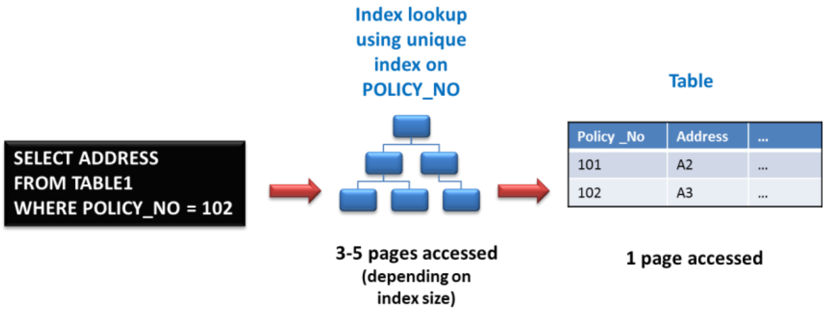
Together, these scalability enhancements provide DB2 customers with more flexibility in the way they distribute their workload across the available System z servers, while reducing DB2 CPU usage and improving the performance of key application processes.

Together, these scalability enhancements provide DB2 customers with more flexibility in the way they distribute their workload across the available System z servers, while reducing DB2 CPU usage and improving the performance of key application processes.

New Hash Access Method

Many high-volume OLTP applications need to efficiently access a single row via a fully qualified primary key, but most of the access paths available to DB2 today are optimised for accessing sets of rows. Previously, the most efficient access path for a single row fetch would have been via a unique index on the table, as shown in Figure 6 below.

[8] Traditional DB2 applications usually use "static SQL" which is hard coded into the application and can therefore be checked and analysed when the program is prepared, saving valuable elapsed and CPU time when the program is run. Dynamic SQL is not hard coded, and cannot therefore be prepared for execution in advance. It is more flexible than static SQL, but often costs more to execute because checking and access path selection can only be done at run time.


Index lookup using unique index on POLICY_NO
Table
3-5 pages accessed (depending on index size)
1 page accessed

```
SELECT ADDRESS
FROM TABLE1
WHERE POLICY_NO = 102
```

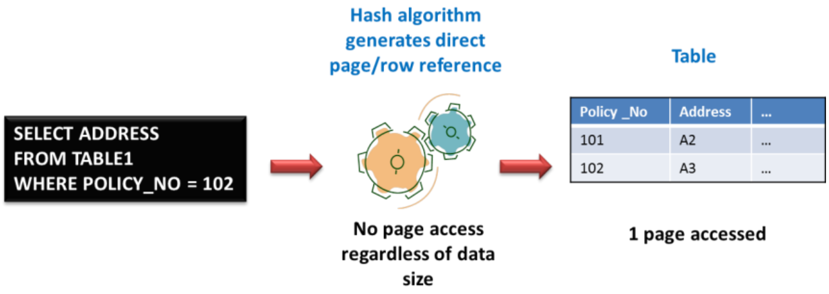
Policy _No	Address	...
101	A2	...
102	A3	...

Figure 6 – Single Row Access via Unique Index

> ***"We will definitely be using the hashing feature in our DB2 applications. In our beta testing hash access saved up to 50% CPU time when compared to traditional table access, and for that kind of improvement we're happy to accept the compromises that hash tables bring."***
>
> **Frank Petersen, DB2 Systems Programmer, BankData**

While this access path can be highly efficient if multiple rows need to be accessed sequentially, the overhead of navigating the index structure can be expensive for single-row access. Depending on the size of the data, the example above would typically require DB2 to access a total of 4 – 6 pages in the index and table, some of which might also require a physical I/O operation to pull the page into the buffer pool if it isn't already resident.

DB2 10 introduces a completely new access method, known as hash access. Where a table has been enabled for hash access, the vast majority of requests for a single row using the unique key will be satisfied with a single page access [9], as DB2 will use the key as input to a hashing algorithm which will produce the page number and row offset needed to directly access the given row. This is illustrated in Figure 7 below.


Hash algorithm generates direct page/row reference
Table
No page access regardless of data size
1 page accessed

```
SELECT ADDRESS
FROM TABLE1
WHERE POLICY_NO = 102
```

Policy _No	Address	...
101	A2	...
102	A3	...

Figure 7 – Single Row Access via a Hash

> ***Hash access offers the potential of significantly reduced CPU and potentially lower I/O and elapsed times. The best candidates for hash access are random single row access into a table with a fixed, known size, with many rows per page, and with small variation in row sizes.***

Hash access tables are not without their disadvantages: they will require 20%-100% more disk space than traditional types and could be more expensive to access for multiple-row access. However, for many high performance applications that predominantly use single-row access these limitations could be an acceptable trade-off for significantly reduced CPU (due to fewer pages accessed) and potentially lower I/O and elapsed times (if physical I/O operations are avoided for index page access).

Automated Statistics

One of the most important factors in DB2 query performance is the access path chosen by DB2, and that is heavily influenced by the table and index statistics gathered by the RUNSTATS utility. The old adage of "garbage in, garbage out" is very relevant to access path

[9] If a row cannot fit on the correct page due to space limitations, it may be relocated in an overflow area and therefore two page accesses may sometimes be required. This should not normally occur if the volumes are specified correctly when the table is defined.

selection, so an important part of any DBA's job is to ensure that accurate, up-to-date statistics are available for critical tables.

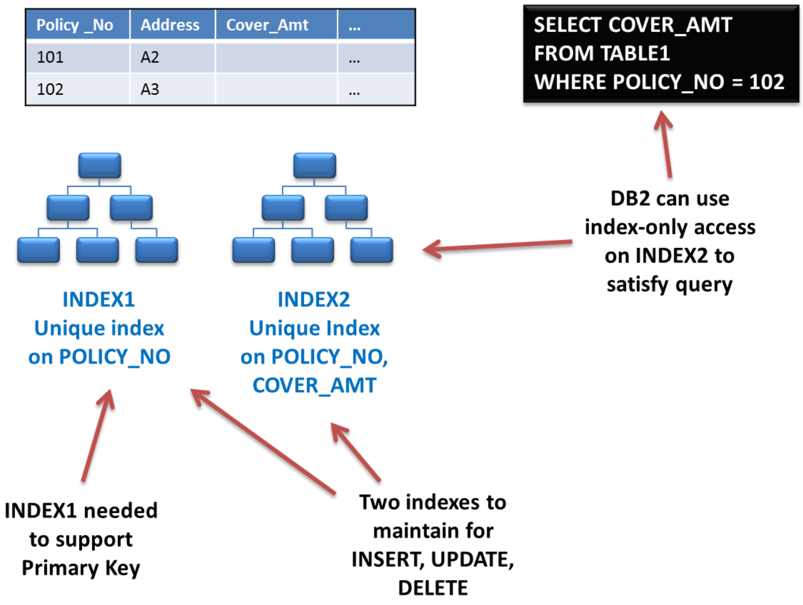
DB2 10 introduces a new automated statistics feature. This frees the DBA to focus on more demanding activities, improving productivity and potentially reducing CPU requirements due to improved access paths and/or elimination of unnecessary RUNSTATS jobs

RUNSTATS can be scheduled to run at fixed times, but this doesn't allow for ad-hoc processes that can significantly change the table characteristics. A simple scheduled approach can result in statistics not being gathered often enough (leading to poor access paths and increased CPU/elapsed time) or too often (wasting the CPU used by the RUNSTATS utility).

DB2 10 introduces a new automated statistics feature to allow DB2 to dynamically monitor the currency of table and index statistics, and automatically schedule the necessary RUNSTATS job when required. This frees the DBA to focus on more demanding activities, improving productivity and potentially reducing CPU requirements due to improved access paths and/or elimination of unnecessary RUNSTATS jobs.

Include Additional Columns in Unique Index

When a primary key is formally defined on a table, DB2 requires a unique index to be defined. In previous versions of DB2, that index could only contain the primary key columns. If additional columns were required to support specific SQL statements (such as the SELECT statement shown in Figure 8 below), it was necessary to create a separate additional index.



Policy _No	Address	Cover_Amt	...
101	A2		...
102	A3		...

```
SELECT COVER_AMT
FROM TABLE1
WHERE POLICY_NO = 102
```

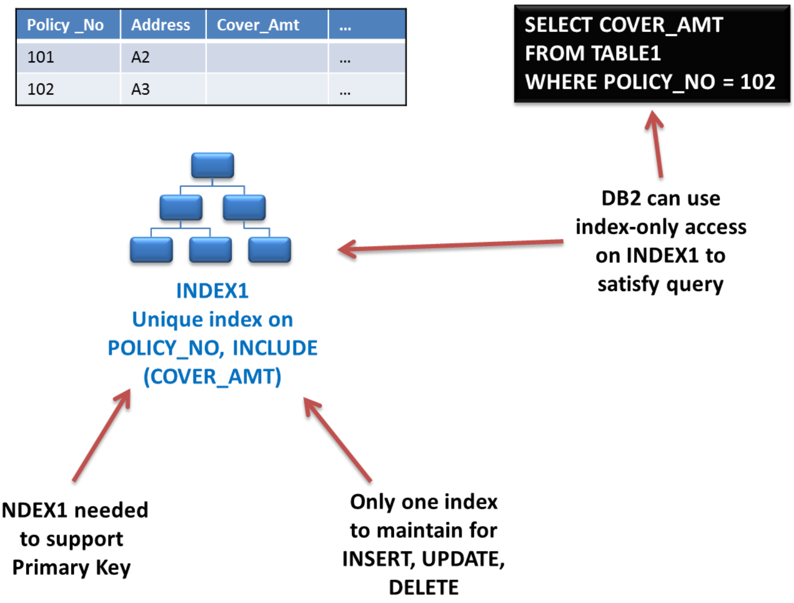
DB2 can use index-only access on INDEX2 to satisfy query
INDEX1 Unique index on POLICY_NO
INDEX2 Unique Index on POLICY_NO, COVER_AMT
INDEX1 needed to support Primary Key
Two indexes to maintain for INSERT, UPDATE, DELETE

Figure 8 – Multiple Indexes to Support Index-Only Access

DB2 10 allows columns other than the unique key to be specified in the index definition. This allows some indexes to be dropped, removing the DASD, CPU and I/O overheads associated that that index while continuing to support the efficient access path needed by the SELECT statement

This approach allowed the SQL statement to be executed efficiently, but added unnecessary overheads to INSERT, UPDATE and DELETE operations as two indexes needed to be maintained rather than one.

DB2 10 allows columns other than the unique key to be specified in the index definition. As shown in Figure 9 below, this allows the second index to be dropped, removing the DASD, CPU and I/O overheads associated that that index while continuing to support the efficient access path needed by the SELECT statement.



Policy _No	Address	Cover_Amt	…
101	A2		…
102	A3		…

```
SELECT COVER_AMT
FROM TABLE1
WHERE POLICY_NO = 102
```

DB2 can use index-only access on INDEX1 to satisfy query
INDEX1
Unique index on POLICY_NO, INCLUDE (COVER_AMT)
INDEX1 needed to support Primary Key
Only one index to maintain for INSERT, UPDATE, DELETE

Figure 9 – Single Index with Non-Unique Columns Included

Where additional indexes have had to be created specifically to support this kind of access, removal of the redundant index will significantly reduce the cost of any update operations against the underlying table. Initial lab tests have shown up to 30% CPU reduction in INSERT with identical query performance in one example where two indexes were replaced with single one using INCLUDE columns.

Buffer Pool Enhancements

As processor speeds continue to increase at a faster rate than disk subsystems, the relative cost of performing random I/O operations is increasing and minimising I/O has become a major objective in improving performance. DB2's buffer pools cache frequently accessed data in memory, avoiding physical I/O activity and significantly improving performance.

DB2 10 introduces a number of new enhancements for buffer pools which should yield significant performance benefits:

- **Large page support**. With the move to a 64-bit computing platform in DB2 Version 8, it became possible to define dramatically bigger buffer pools, up to 1TB. However the size of each hardware "page" within the pool remained at 4KB, so large pools can have many millions of pages leading to increased z/OS overheads.

> ***"We use very large buffer pools – some of them up to 3.2GB in size. We rely on efficient access to buffered data and any saving in the cost of accessing that data will be very beneficial."***
>
> **Philipp Nowak, BMW DB2 Product Manager**

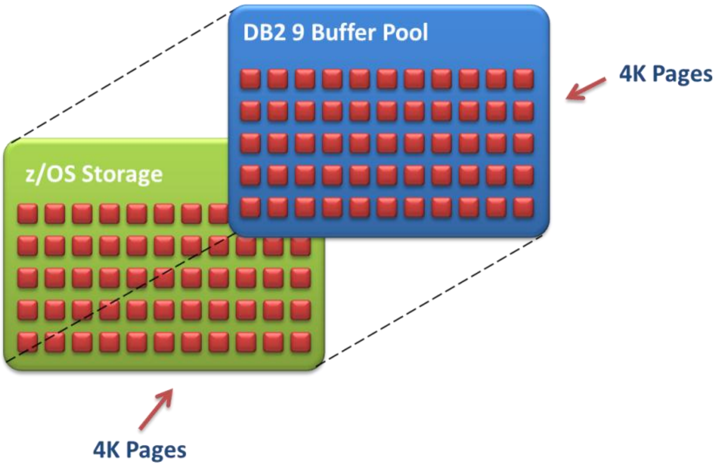


Figure 10 – Buffer Pool using 4K z/OS Page Size

IBM's z10 and newer z196 servers are able to support 1MB page sizes within the hardware, which will result in fewer pages and more efficient access to data within the DB2 buffer pools. Internal IBM testing has shown CPU reductions of 1-4% with this feature enabled.

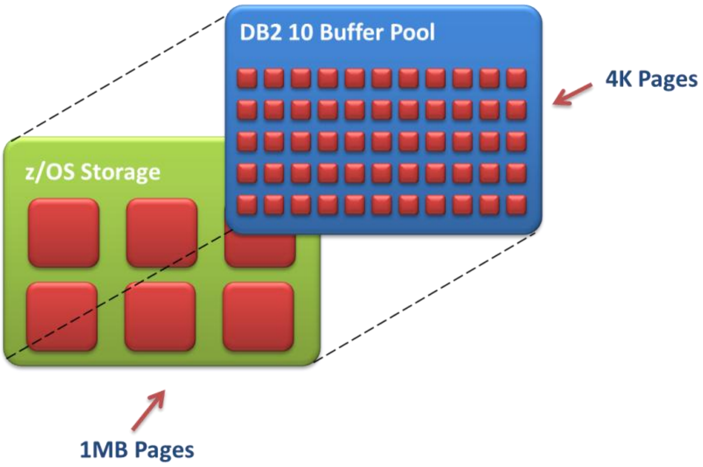


Figure 11 – Buffer Pool using 1MB z/OS Page Size

> ***A new attribute introduced in DB2 10 allows a buffer pool to be marked as "in memory". This enhancement could improve access times to performance critical tables.***

- **In-memory pagesets**. Many DB2 applications make extensive use of "code tables": small, frequently referenced lookup tables. Such tables are often performance critical, and are placed in separate buffer pools that have been sized to ensure that all data remains in storage to avoid any I/O delays. A new attribute introduced in DB2 10 allows a given buffer pool to be marked as "in memory". DB2 will automatically read all data into this buffer pool at start-up (avoiding I/O delays the first time a given data page is accessed) and the optimiser will assume a zero I/O cost when assessing the cost of accessing the table. This enhancement could further improve access times to these performance critical tables.

- **Memory allocation on demand**. In previous versions of DB2, the full amount of storage was allocated as soon as the first table or index belonging to a given buffer pool was allocated. Over-sized pools could therefore reserve storage that was never used. DB2 10 allocates storage as it is required, allowing more efficient use of available storage within the System z server.

Dynamic Statement Cache Enhancements

As mentioned elsewhere in this document, dynamic SQL is becoming more and more prevalent, and DB2 allows dynamic SQL statements to be cached in order to avoid most of the overheads usually associated with executing SQL in this way. However, the dynamic statement cache previously relied on SQL statements being identical in order to be able to re-use the cached statement.

In the example shown in Figure 12 below, the two SQL statements are different (they are selecting a different policy number) and therefore they will be separately cached even though the access path taken is likely to be identical for each one. This uses up valuable space in the dynamic statement cache, and forces DB2 to fully re-prepare each statement causing significant CPU and performance overheads for high volume transactions.

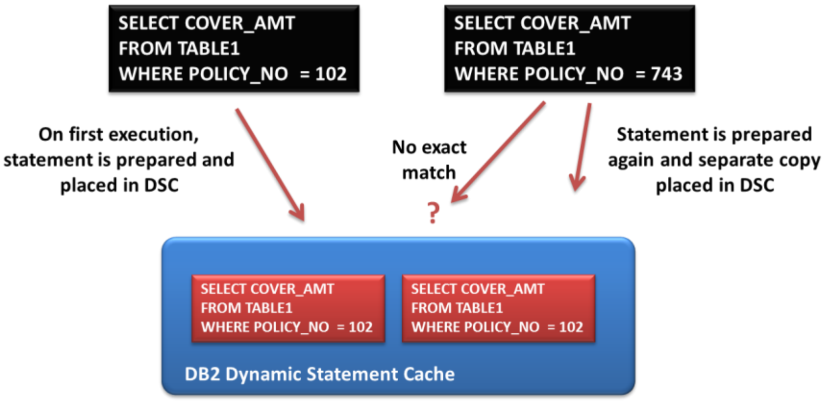


Figure 12 – DSC with No Parameter Markers – Previous Releases

DB2 10 delivers an important enhancement that allows DB2 to recognise that an incoming dynamic SQL statement is fundamentally the same as a previously cached version, even when parameter markers have not been coded by the developer

Although it is possible to address this issue in previous releases through the use of parameter markers[10], many dynamic SQL applications do not use them due to the effort involved in changing the code. Each SQL statement and the surrounding code must be changed, which could add up to many hundreds of lines of code for some applications.

DB2 10 delivers an important enhancement that allows DB2 to recognise that an incoming dynamic SQL statement is fundamentally the same as a previously cached version, even when parameter markers have not been coded by the developer. This is shown in Figure 13 below.

[10] Parameter markers allow literals within SQL statements to be replaced with a “?” to represent a host variable in the application.

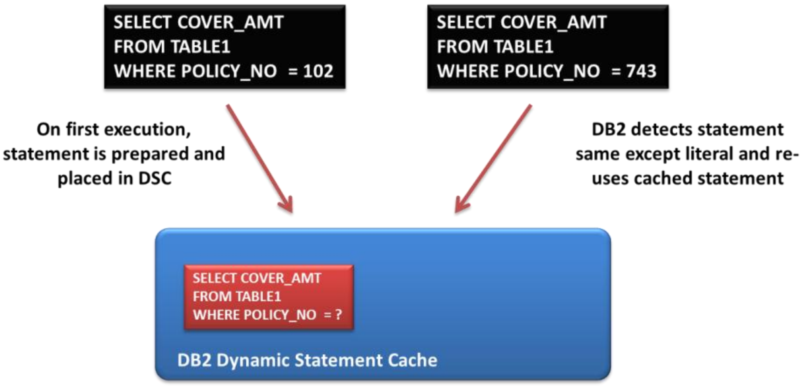

SELECT COVER_AMT
FROM TABLE1
WHERE POLICY_NO = 102
SELECT COVER_AMT
FROM TABLE1
WHERE POLICY_NO = 743
On first execution, statement is prepared and placed in DSC
DB2 detects statement same except literal and re-uses cached statement
SELECT COVER_AMT
FROM TABLE1
WHERE POLICY_NO = ?
DB2 Dynamic Statement Cache

Figure 13 – DSC with No Parameter Markers – DB2 10

In order to enable this feature, a single statement property has to be set within the code. Although this still requires the application code to be changed, the scope and magnitude of the change is much less than that required to implement parameter markers.

This feature will significantly decrease the effort required to enable dynamic SQL statement re-use, which in turn could increase the proportion of dynamic SQL applications able to benefit from the dynamic statement cache, driving down CPU costs and allowing better use to be made of the storage devoted to the cache.

This feature will significantly decrease the effort required to enable dynamic SQL statement re-use, which in turn could increase the proportion of dynamic SQL applications able to benefit from the dynamic statement cache, driving down CPU costs and allowing better use to be made of the storage devoted to the cache

Enhanced Dynamic Schema Change

DB2 Version 8 introduced some major enhancements to allow database structures to be altered dynamically, which were further enhanced in DB2 9. IBM continues to extend this capability to include the most commonly used changes, and DB2 10 adds the following schema changes to those that can be made online:

- Altering the table space type, to allow older table spaces to be converted to the newer "universal table space" format introduced in DB2 9[11]. Universal table spaces can greatly simplify space management, improving both DBA productivity and data availability
- Altering the MEMBER CLUSTER attribute. This is an important performance optimisation in a data sharing environment, and requires significant DBA effort and data outage to implement prior to DB2 10.
- Altering table space page size, dataset size and segment size, and index page size. These parameters can have a significant impact on I/O performance, and DB2 9 introduced some new options which this enhancement will make easier and quicker to implement.

Dynamic schema change significantly improves data availability, but also reduces the possibility of human error and improves DBA productivity as complex scripts to drop and recreate database objects can be replaced by a single SQL statement.

Dynamic schema change significantly improves data availability, but also reduces the possibility of human error and improves DBA productivity as complex scripts to drop and recreate database objects can be replaced by a single SQL statement.

[11] Please see Appendix A – DB2 9 Review on page 16 for more details of this feature.

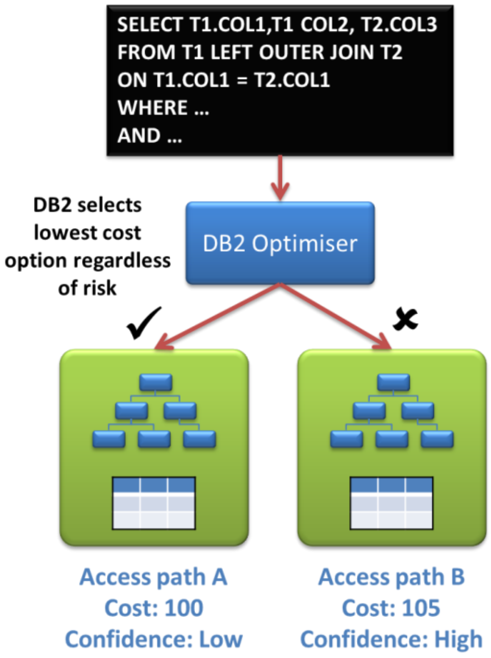
Optimiser Enhancements

DB2 10 includes a number of enhancements to DB2's industry-leading optimiser – the key component that allows DB2 to pick the most efficient access path for a given query. These include:

> ***"With our DB2 10 testing so far, we have had quite a few surprises but all of them have been good. Every single SQL statement we have tested has been better or the same as our current optimal paths – we have yet to see any significant access path regression. We had to spend a lot of time tuning SQL with DB2 9, but we expect that to disappear when we upgrade to DB2 10."***
>
> **Philipp Nowak, BMW DB2 Product Manager**

- **Safe query optimization.** The DB2 optimiser often has to make educated guesses on the amount of data that will be filtered by a given predicate in an SQL statement. Enhancements to the optimiser in DB2 10 allow it to take into account the degree of confidence it has with these estimates, allowing it to choose a slightly more expensive access path if it has significantly lower risk associated with it.

  In the example shown in Figure 14 below, a previous version of DB2 will select the access path with the lowest estimated cost, regardless of the degree of uncertainty associated with any filtering predicates.



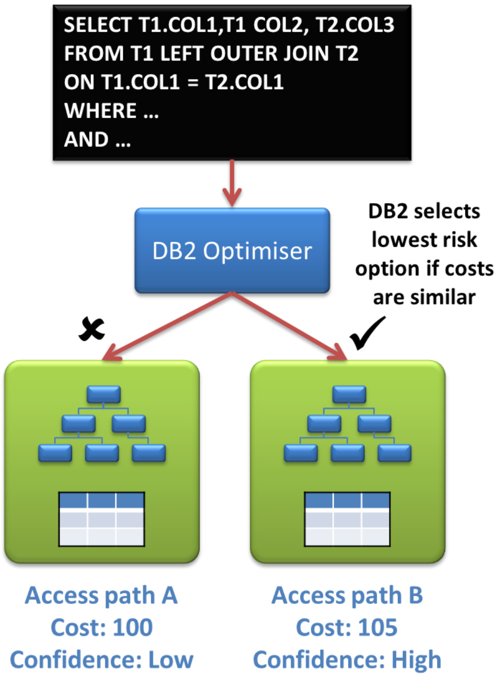
```
SELECT T1.COL1,T1 COL2, T2.COL3
FROM T1 LEFT OUTER JOIN T2
ON T1.COL1 = T2.COL1
WHERE …
AND …
```

Figure 14 – Optimisation based on Cost

> ***Safe query optimization increases the consistency of the access path decisions made by DB2, allowing for more predictable performance across the many different environments it has to support in today's enterprise***

In this second example shown in Figure 15, the DB2 10 optimiser takes into account the fact that Access Path B has only a marginally higher estimated cost but the degree of confidence in the estimate is much higher, and selects this access path instead.

This feature increases the consistency of the access path decisions made by DB2, allowing for more predictable performance across the many different environments it has to support in today's enterprise.



```
SELECT T1.COL1,T1 COL2, T2.COL3
FROM T1 LEFT OUTER JOIN T2
ON T1.COL1 = T2.COL1
WHERE …
AND …
```

Figure 15 – Safe Optimisation

DB2 10 removes a number of restrictions when using this CPU parallelism, allowing it to be used in more situations than previous editions. As much as 80% of the CPU time for large parallel SQL queries can be redirected to zIIP processors, so this also has the potential to reduce overall CPU costs for the DB2 workload

- **Parallelism enhancements.** DB2 has long supported the ability to reduce the elapsed time for key processes by splitting them into multiple tasks which can execute concurrently. DB2 10 removes a number of restrictions when using this CPU parallelism, allowing it to be used in more situations than previous editions. As much as 80% of the CPU time for large parallel SQL queries can be redirected to zIIP processors[12], so this also has the potential to reduce overall CPU costs for the DB2 workload.

- **Improved Predicate Filtering.** When DB2 executes a query, two stages of processing can be involved. Stage 1 processing is responsible for retrieving the data pages from the buffer and initial filtering by applying simpler predicates. If necessary, Stage 2 processing then applies any remaining predicates before the data is passed back to the application, as shown in Figure 16 below.

  Predicates applied during Stage 1 processing are more efficient, as they allow more data to be filtered earlier in the process. DB2 10 includes an enhancement that allows some predicates (such as scalar functions) to be evaluated at Stage 1 rather than Stage 2 as was previously the case. This can significantly decrease the overall cost of a query, reducing the amount of CPU and elapsed time necessary to execute it.

[12] One of the ways in which IBM is reducing the overall cost of mainframe workloads is to offer customers the option of installing additional “speciality processors” within their System z machines. These processors are capable of running only specific types of work, but in so doing they can reduce the load on the general-purpose CP processors and therefore the amount chargeable CPU consumed. The zIIP is a speciality processor designed to offload specific types of data and transaction processing workloads such as remote SQL statements, some DB2 utility processing, and network encryption

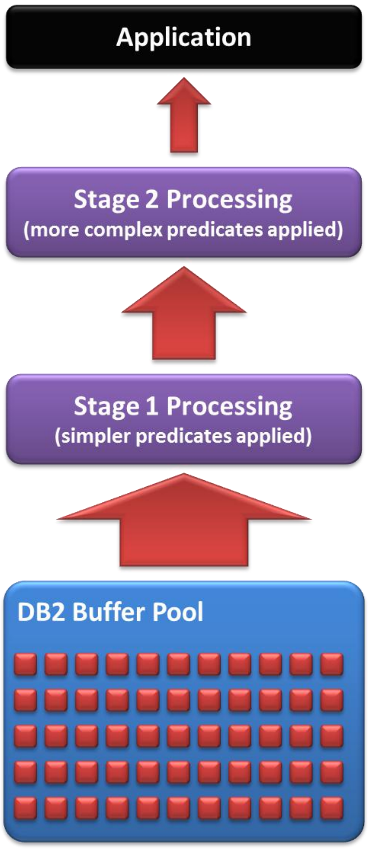
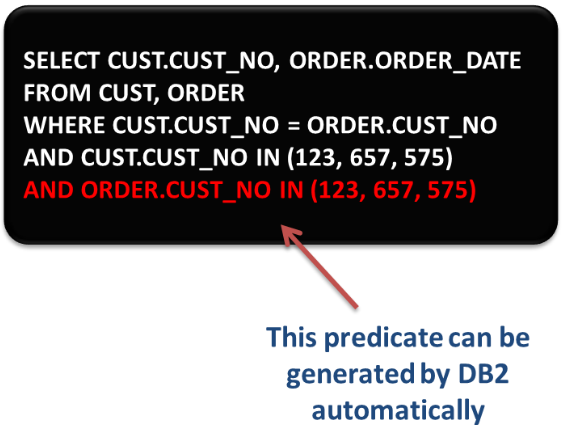


Figure 16 – Stage 1 and Stage 2 Query Processing.

DB2 10 includes several enhancements to improve index access and reduce query cost.

- **Improved Index Access.** DB2 10 includes several enhancements to improve index access. A new access path allows DB2 to scan an index just once where several OR predicates can all use the same index, whereas previous versions would have scanned the index multiple times. A technique known as "predicate transitive closure" allows DB2 to use predicates supplied by the user in a query to derive additional predicates and potentially improve the access path chosen. DB2 has been able to use this technique for some time, but DB2 10 is now able to also apply it to IN-list queries, as shown in the example in Figure 17 below.



```
SELECT CUST.CUST_NO, ORDER.ORDER_DATE
FROM CUST, ORDER
WHERE CUST.CUST_NO = ORDER.CUST_NO
AND CUST.CUST_NO IN (123, 657, 575)
AND ORDER.CUST_NO IN (123, 657, 575)
```

This predicate can be generated by DB2 automatically

Figure 17 – Predicate Transitive Closure for In-Lists.

Together, these optimiser enhancements further consolidate DB2's position as having the most sophisticated and effective optimiser in the industry, able to minimise overall CPU demand through the selection of the most efficient access path.

Together, these optimiser enhancements further consolidate DB2's position as having the most sophisticated and effective optimiser in the industry, able to minimise overall CPU demand through the selection of the most efficient access path. Although several of these will require DB2 packages to be rebound, another enhancement known as plan stability (described on page 24 of this document) removes the main inhibitor to such enhancements being fully exploited.

MEMBER CLUSTER for Universal Table Spaces

The MEMBER CLUSTER table space option can dramatically reduce lock contention for high INSERT activity in a data sharing environment, and is a commonly used tuning tool. DB2 9 introduced the new Universal Table Space (UTS) format which offered many advantages over traditional table spaces[13], but did not support the MEMBER CLUSTER option.

The DBA therefore had to choose between implementing MEMBER CLUSTER (which required a change to non-UTS table spaces and a drop and recreate of the table) or staying with UTS and living without the benefits of MEMBER CLUSTER. This was a particular problem in SAP environments, as recent versions of SAP make extensive use of UTS table spaces.

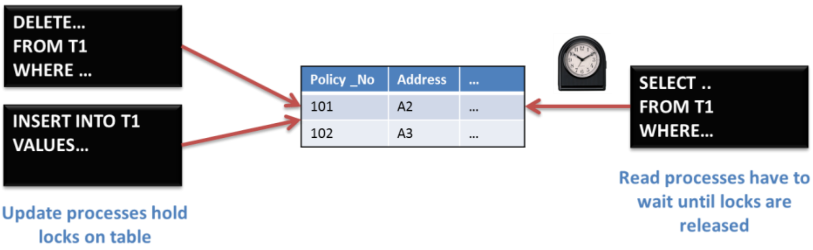
The MEMBER CLUSTER changes give the DBA more flexibility in improving application performance within a data sharing environment, with greatly reduced implementation effort.

DB2 10 removes this restriction and allows MEMBER CLUSTER to be specified for UTS table spaces. As previously discussed, the DB2 10 enhancements to dynamic schema change also allow MEMBER CLUSTER to be implemented via a simple ALTER rather than requiring the table space to be dropped and recreated. These changes give the DBA more flexibility in improving application performance within a data sharing environment, with greatly reduced implementation effort.

Currently Committed

A common problem in high-volume transaction environments involves read-only processes waiting until locks held by updating processes are released. DB2 has introduced many forms of lock avoidance over the years, and a further feature in DB2 10 provides additional flexibility.

The diagram in Figure 18 below provides a typical example, with the SELECT process on the right waiting on the DELETE and INSERT processes on the left to complete and release their locks.


DELETE…
FROM T1
WHERE …
INSERT INTO T1
VALUES…
SELECT ..
FROM T1
WHERE…
Update processes hold locks on table
Read processes have to wait until locks are released

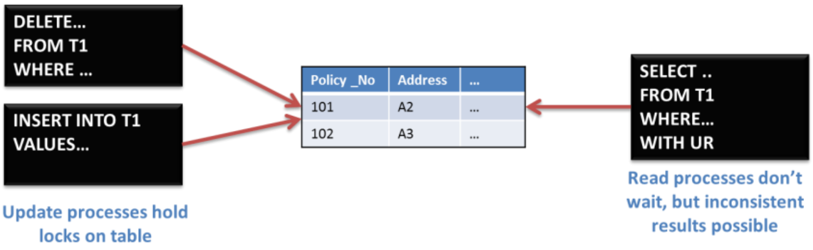
Policy _No	Address	…
101	A2	…
102	A3	…

Figure 18 – Typical Lock Wait Scenario

Several releases ago, DB2 introduced the possibility of using "uncommitted read" semantics for read processes, to effectively

[13] Please see Appendix A – DB2 9 Review on page 16 for more details of this feature.

ignore any locks held by other processes as shown in Figure 19 below. However, by definition such read processes could return inconsistent results: in the example below if the DELETE and INSERT were moving a sum of money between two bank accounts it is possible for the money to be counted twice, or not at all, by the SELECT process. UR is therefore useful only in situations where the absolute consistency of the selected data is not important.

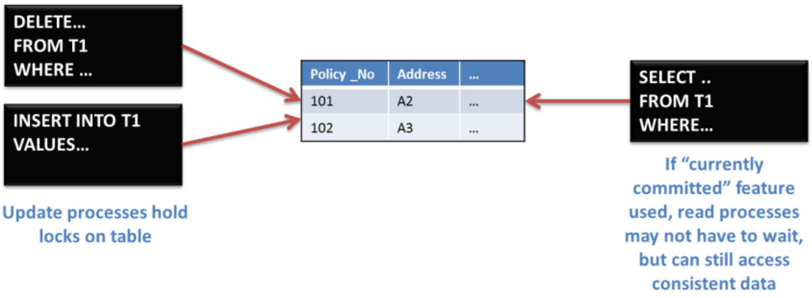

DELETE…
FROM T1
WHERE …
INSERT INTO T1
VALUES…
Update processes hold locks on table
SELECT ..
FROM T1
WHERE…
WITH UR
Read processes don't wait, but inconsistent results possible

Policy _No	Address	…
101	A2	…
102	A3	…

Figure 19 – Use of Uncommitted Read

DB2 10 introduces a new way of specifying concurrency options for read processes that allows them to access consistent data by ignoring an uncommitted DELETE or INSERT activity and reading the last committed version of rows in the table as shown in Figure 20 below. Note that in its current implementation, currently committed applies to delete and insert activity only, and read processes will still have to go through normal locking mechanisms for pending updates.

As scalability limits are removed and DB2 supports ever-higher transaction volumes, improving application concurrency will become increasingly important. This feature is a significant step forward, and brings DB2 in line with similar capabilities offered by several other relational databases.


DELETE…
FROM T1
WHERE …
INSERT INTO T1
VALUES…
Update processes hold locks on table
SELECT ..
FROM T1
WHERE…
If "currently committed" feature used, read processes may not have to wait, but can still access consistent data

Policy _No	Address	…
101	A2	…
102	A3	…

Figure 20 – Use of Currently Committed

The currently committed behaviour can be enabled on an application by application basis via a simple BIND parameter. As scalability limits are removed and DB2 supports ever-higher transaction volumes, improving application concurrency will become increasingly important. This feature is a significant step forward, and brings DB2 in line with similar capabilities offered by several other relational databases.

Backup & Recovery Improvements

The combination of DB2 for z/OS and the underlying System z platform are undisputed in terms of resilience and security. However, situations do occur when database recovery is necessary: either as a result of human error or hardware or software failure. In such circumstances, it is of paramount business importance to reliably recover the affected data in the shortest possible time.

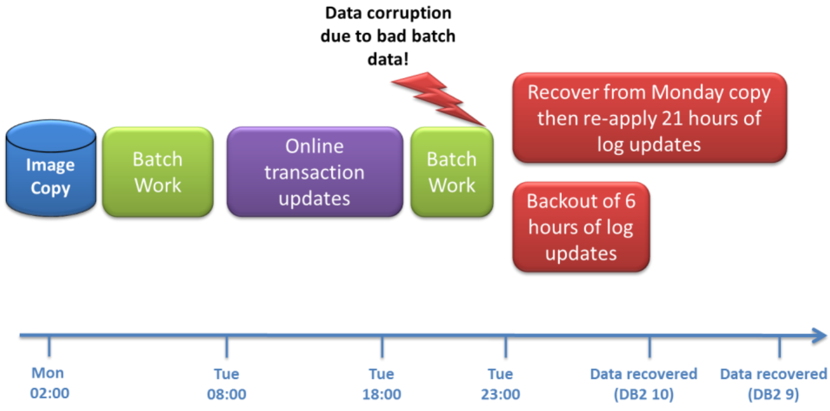

Data corruption due to bad batch data!
Image Copy
Batch Work
Online transaction updates
Batch Work
Recover from Monday copy then re-apply 21 hours of log updates
Backout of 6 hours of log updates
Mon 02:00
Tue 08:00
Tue 18:00
Tue 23:00
Data recovered (DB2 10)
Data recovered (DB2 9)

Figure 21 – Backout v Log Apply Recovery

With DB2 10'snew backout capability, the DBA has new options that could result in a quicker overall recovery and less disruption/cost to the business.

As shown in Figure 21 above, for the first time DB2 provides DBAs with the option of backing out changes in the event of a database recovery[14], in addition to the more traditional recover/roll forward approach. In this example, incorrect input data to a batch process has corrupted the DB2 database. In prior releases, the DBA would be forced to run a full recover to the point in time prior to the batch process, requiring all affected tables to be restored from the previous image copy and 21 hours of batch/online updates to be replayed from the DB2 log. With DB2 10's backout capability, the DBA could instead choose to undo the committed changes made by the offending batch process, resulting in a quicker overall recovery and less disruption/cost to the business.

Other related enhancements in DB2 10 remove some of the restrictions on use of FlashCopy[15] technology, and provide more options for the production of "consistent" backups without impacting application availability.

Other Efficiency Enhancements

A number of other performance and productivity enhancements are delivered in DB2 10, including:

By combining the open, fetch and close sub-tasks into a single request, DB2 10 is able to more efficiently execute the instruction and reduce the amount of CPU and elapsed time consumed by distributed requests.

- **Distributed Access Optimisations**. DB2 is increasingly being called upon to act as a data server for distributed applications such as SAP. DB2 10 delivers a useful enhancement for incoming distributed requests involving a single-row SELECT using the FETCH FIRST ROW ONLY clause. By combining the open, fetch and close sub-tasks into a single request, DB2 10 is able to more efficiently execute the instruction and reduce the amount of CPU and elapsed time consumed.

  Another enhancement in this area allows the DBA to specify when the DB2 resources held by incoming distributed requests are released, via the RELEASE parameter of the package bind. For short OLTP-type distributed requests this ability is expected to save significant CPU and elapsed time by preventing repeated de-allocation and re-allocation of DB2 packages each time a distributed request is received.

Our fastest growing (and most expensive) workload is Dynamic SQL over DDF. V10 has many improvements to reduce the CPU cost of these workloads.

DB2 10 Beta Customer

[14] Note that DB2 has long supported backout during system recoveries or system restarts. DB2 10 provides this option for individual table space recoveries for the first time.

[15] FlashCopy is a function provided by IBM disk storage systems that allows near-instantaneous copies to be made of data. Other vendors provide similar functionality.

Internal IBM testing has demonstrated CPU savings of 10% for this type of workload.

DB2 10 extends SQL procedure support to improve its consistency with DB2 for Linux, UNIX and Windows and other midrange databases, making it easier to port existing applications to DB2 for z/OS.

- **Native SQL Procedure Enhancements**. Stored procedures can be written in most of the programming languages commonly in use on the mainframe, but developers and DBAs also have the option to write stored procedures in native SQL (also known as Native SQL Procedures). Writing procedures in this language provides some significant benefits in application portability and maintenance as well as some cost/performance benefits (see page 5), DB2 10 extends SQL procedure support to include some valuable new capabilities, such as the ability to accept XML objects as parameters and use SQL scalar and table functions. These extensions improve DB2 10's consistency with DB2 for Linux, UNIX and Windows and other midrange databases, making it easier to port existing applications to DB2 for z/OS.

DB2 10 allows more work to be offloaded to a zIIP speciality engine. These changes can directly reduce the operational costs of eligible DB2 workloads.

- **Improved zIIP exploitation**. DB2 10 allows some portions of the RUNSTATS utility, buffer pool prefetch and deferred write processing to be offloaded to a zIIP speciality engine[12]. The amount of work from incoming distributed connections that can be offloaded to zIIP processors has also been increased[16] so that up to 60% is now eligible. These changes can directly reduce the operational costs of eligible DB2 workloads.

- **SSD support**. DB2 10 supports the use of Solid State Disk (SSD) drives. Although currently much more expensive per MB than traditional magnetic disk, SSDs are usually less expensive per I/O and provide some significant performance benefits. Placing certain DB2 objects on SSD (sort work files, high performance tables that cannot be fully cached in memory, etc.) can dramatically reduce I/O times and improve performance.

We have measured a 38% reduction in CPU and a 7% reduction in suspend time for some heavy insert workloads in a data sharing environment. That's a significant saving which provides immediate business benefit."

Philipp Nowak, BMW DB2 Product Manager

- **Parallel index update at insert**. Applications generating a high volume of INSERT activity against a given DB2 table may encounter significant I/O delays due to the need to maintain multiple indexes in addition to the base table data. DB2 previously updated indexes sequentially but in DB2 10 it is able to overlap the I/O operations for index updates, reducing the elapsed time for these processes. In common with other parallel activities, reduced elapsed time is usually achieved at the cost of an increase in overall CPU time, but in this case the overhead is very small, and eligible for offload to a zIIP engine if one is available.

Parallel index update will reduce the elapsed time for some of our most critical processes, delivering a more responsive application and allowing us to sustain greater throughput."

Paulo Sahadi, Senior Production Manager, Information Management Division, Banco do Brasil

- **Work File enhancements**. Work files are used to support joins and large sorts, and are used by a large proportion of most typical DB2 workloads. DB2 10 provides a number of enhancements in this area, including the ability to handle larger records (up to 64KB), optimisations for small sorts, and the ability to do more work in memory rather than externally on disk. Collectively these are expected to reduce CPU time and improve scalability.

[16] This functionality is also available to be retrofitted to DB2 Version 8 and DB2 9, via APAR PM12256.

- **Inline Large Objects.** Previously, DB2 Large Objects (LOBs) had to be stored in a separate table space to the conventional table data, requiring both locations to be accessed when LOB information was retrieved. DB2 10 allows part of the LOB to be stored "inline" within the base table, so that smaller LOBs (up to a few KB in size) can be retrieved without accessing the auxiliary table space, thereby improving performance and reducing CPU. One beta customer measured up to 80% CPU savings for SELECTs and 30% improvement for INSERT where the LOB can fit in the base table.

DB2 10 allows part of a LOB to be stored "inline" within the base table, thereby improving performance and reducing CPU. One beta customer measured up to 80% CPU savings for SELECTs and 30% improvement for INSERT where the LOB can fit in the base table

Resilience

The System z platform is rightly famed as one of the most robust and secure computing platforms on the planet. However, business and regulatory requirements in this area continue to get more demanding so this is an important and ongoing focus area for the DB2 development team.

This section groups together some key new features that make DB2 more resilient to possible negative impacts of planned change, as well as increasing the flexibility and scope of the critical access control mechanisms that protect sensitive data from unauthorised access.

Plan Stability

One of the major headaches all DB2 users face when upgrading to a new release is the possibility of access path regression. In order to benefit from any enhancements to the optimiser (such as those covered in the section on CPU Reductions on page 5 and Optimiser Enhancements on page 17), DB2 plans and packages typically need to be rebound under the new release.

The vast majority of the time, this will result in the same or better access path being selected (with a corresponding drop in CPU cost and elapsed time), but occasionally DB2 may select a worse one and performance suffers. Even though the risk of performance regression is usually small, it is enough to act a serious disincentive and many DB2 customers fail to exploit potential CPU reductions in order to avoid this risk. It is not unusual to see plans and packages that have not been rebound in the last 10-15 years at some sites.

This basic approach to access path management is shown in Figure 22 below.

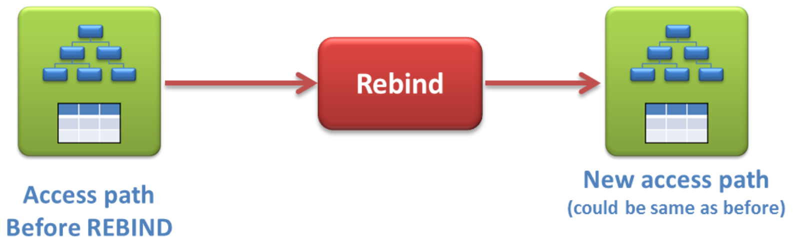


Figure 22 – DB2 with no Plan Stability

In an attempt to rectify this situation, DB2 9 introduced some new functionality in the maintenance stream[17] called Plan Stability. This enhancement provided some welcome new options for REBIND that allowed up to two old versions of a static SQL access path to be stored. If performance regression occurred following a REBIND, the previous access path could be quickly and easily re-established by running another REBIND using the SWITCH parameter. This capability is shown the diagram in Figure 23 below.

Plan stability removes the majority of the risk associated with rebinding static packages following any DB2 upgrade, allowing more customers to exploit more of the performance enhancements delivered in each release.

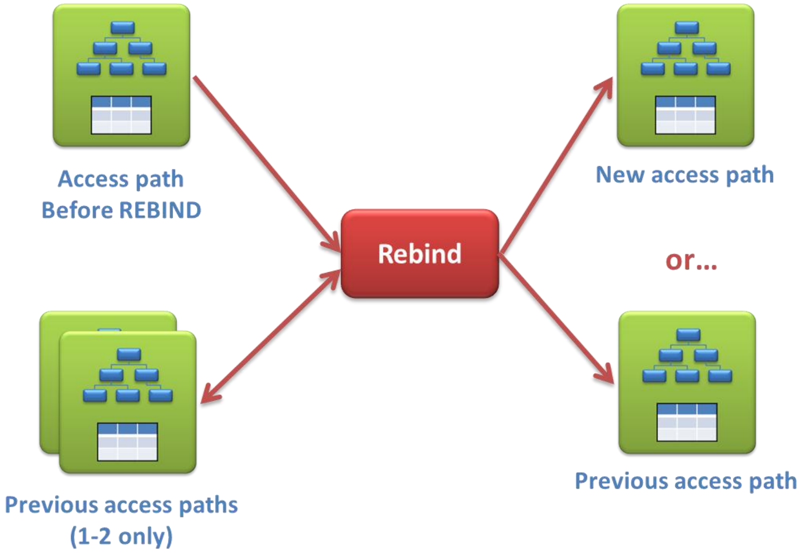


Figure 23 – DB2 9 Plan Stability

Those customers that have not rebound their plans/packages for a considerable time may see very significant benefits, as they are able to exploit several releases worth of optimiser enhancements at once.

DB2 10 incorporates this enhancement into the base DB2 product code, and enhances it to allow any number of previous access paths to be stored.

Plan stability removes the majority of the risk associated with rebinding static packages following any DB2 upgrade, allowing more customers to exploit more of the performance enhancements delivered in each release. Those customers that have not rebound their plans/packages for a considerable time may see very significant benefits, as they are able to exploit several releases worth of optimiser enhancements at once.

Enhanced Audit Capabilities

Many legal compliance and business requirements involve the creation of a detailed audit log for sensitive data held within DB2. Although previous versions of DB2 did have an audit capability, this had a number of serious flaws – the most significant of which only allowed DB2 to audit the first access to any given table within a single transaction, with all subsequent accesses not being recorded on the audit log.

DB2 10 provides a long-awaited solution, in the shape of a formal audit policy. This is flexible enough to allow specific tables to be monitored for specific periods, and includes a record of ALL relevant SQL activity within a given transaction, including read and update activity. Wildcarding is supported to allow a single audit rule to cover multiple tables, and distributed identity support allows the "real"

[17] Via APAR PK52523 – see http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db29.doc.perf/db2z_usepackagecopy2alleviateperfregress.htm for more details.

userid passed by distributed applications such as SAP to be recorded in the audit record.

The new audit capabilities in DB2 10 address some long-overdue shortcomings in the functionality provided by previous DB2 releases, and will provide auditors with the means to track and report on all significant access to sensitive DB2 data without having to resort to external packages or expensive application coding.

The new audit capabilities in DB2 10 address some long-overdue shortcomings in the functionality provided by previous DB2 releases, and will provide auditors with the means to track and report on all significant access to sensitive DB2 data without having to resort to external packages or expensive application coding.

More Flexible Administrative Authorities

Many DB2 customers go to great lengths to ensure that access to sensitive DB2 data is limited to users who have a business requirement to be able to read and update it. However, many of the DB2 system authorities[18] necessary for DB2 systems programmers and DBAs to do their jobs also implicitly gave them read and write access to all of the data in the system. Additional processes were therefore required to audit the activities conducted under these authorities.

DB2 10 introduces a number of new system authorities designed to allow proper separation of administration and data access. These are built around specific roles in a typical DB2 environment, as follows:

- **Security Administrator**. The new SECADM authority allows all security-related administration tasks (granting and revoking data access, setting up roles, etc.) to be conducted by a security administrator without having to provide SYSADM super-user privileges. The traditional security privileges held by SYSADM can be removed when this option is enabled.

- **System Database Administrator**. DBAs can now be given a new SYSTEM DBADM authority (allowing them to make schema and structure changes to all databases) with or without implicitly being given the ability to access the underlying data (or give others access). Previously, DBADM had to be provided individually for each database, and implicitly allowed data access.

- **Data Administrator**. The new DATAAACESS authority provides access to all data within a subsystem without the ability to structurally change any of the DB2 objects.

- **Performance Specialist.** Staff responsible for monitoring and tuning SQL need to be able to investigate access paths via the EXPLAIN facility, manage performance traces and update DB2 statistics. The new SQLADM authority provides this set of abilities, without being able to access any underlying user data access or change any DB2 database structures.

- **Developer.** The new EXPLAIN privilege allows developers to check the access path DB2 will use for critical SQL statements without needing to have access to the underlying data being accessed within the SQL.

More flexible administrative authorities will make it considerably easier to address many of the concerns commonly raised when auditing access to sensitive DB2 data.

"As we operate in a multi-national environment, we have to expend a lot of effort to ensure that we adhere to strict local audit requirements. The new administrative authorities in DB2 10 will greatly reduce that effort, and allow us to work in new territories much more quickly than before."

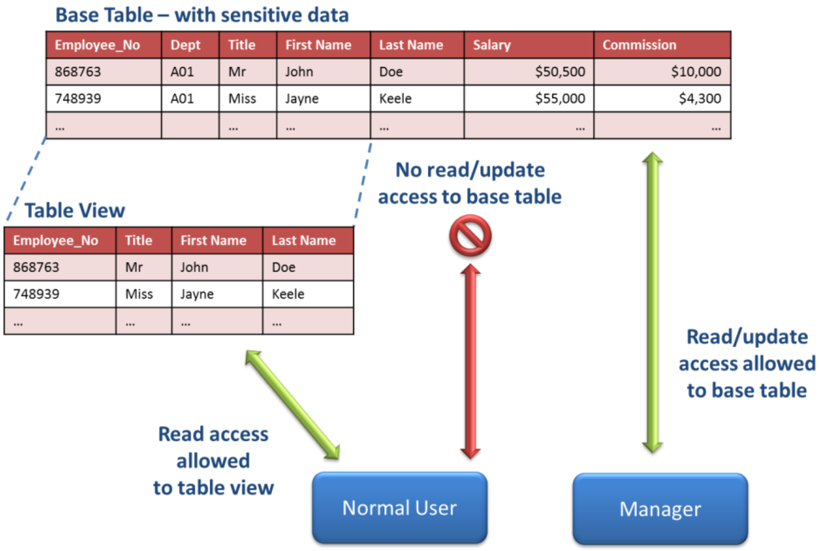
Luca Mussato,
Senior DB2 System Specialist, DB2 Beta Customer

[18] Such as SYSADM and DBADM

Established DB2 sites will have to carefully evaluate their use of existing DB2 system authorities before being able to exploit these new capabilities. However, this increased flexibility will allow many sites to reduce their use if the all-powerful SYSADM authority and make it considerably easier to address many of the concerns commonly raised when auditing access to sensitive DB2 data.

Improved Data Access Control

DB2 access has traditionally been granted at the table level. Where more granular access was required (allowing access to only a subset of the columns or rows in a table) this had to be implemented within each application, or an awkward view mechanism had to be used. This is illustrated in Figure 24 below, where the manager has full read/write access to a table containing sensitive data but a view has had to be defined on the table to limit access for normal users to the non-sensitive columns.



Base Table – with sensitive data

Employee_No	Dept	Title	First Name	Last Name	Salary	Commission
868763	A01	Mr	John	Doe	$50,500	$10,000
748939	A01	Miss	Jayne	Keele	$55,000	$4,300
…		…	…	…	…	…

Table View

Employee_No	Title	First Name	Last Name
868763	Mr	John	Doe
748939	Miss	Jayne	Keele
…	…	…	…

No read/update access to base table
Read/update access allowed to base table
Read access allowed to table view
Normal User
Manager

Figure 24 – Traditional Row/Column Control via Views

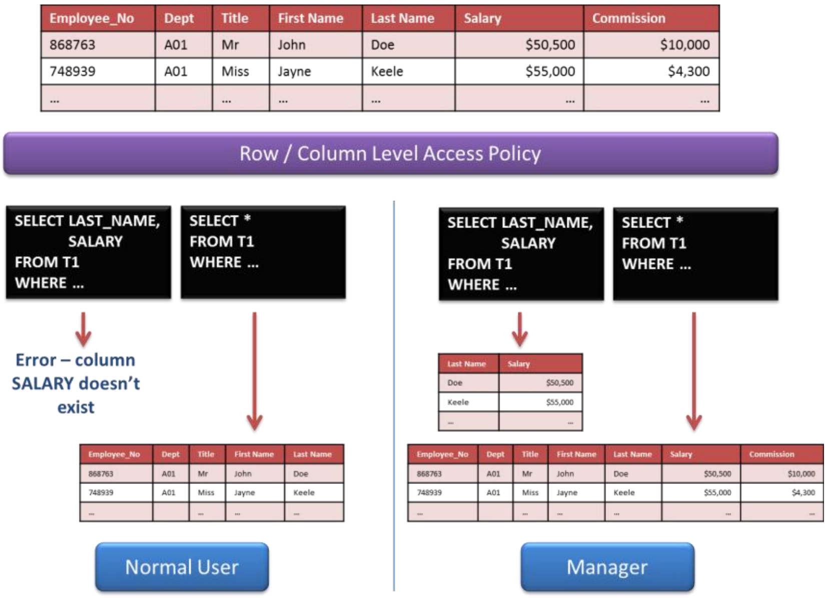
DB2 10 introduces new capabilities for row and column level access control which are fully integrated into the database engine and defined using standard SQL constructs.

Recent versions of DB2 have expanded these choices by providing sophisticated multi-level security mechanisms to meet exacting military standards. However, this approach can be complex to understand and implement, and lacks flexibility.

DB2 10 introduces new capabilities for row and column level access control which are fully integrated into the database engine and defined using standard SQL constructs.

At the row level, a policy can be created to filter rows from the table based on the role/authorities of the requesting user. The row policy applies to INSERT, UPDATE and DELETE statements in addition to SELECT. Similarly, a column policy can be created to mask certain sensitive column values, including subsets of the column. As these policies are integrated at the table level, they are transparent to the all applications accessing the data and universally and consistently enforced.

Figure 25 below shows an example of these new capabilities within a DB2 10 environment. The same table is now protected by a column access policy that only allows managers to see the SALARY and COMMISSION columns in the table. Based on the restrictions defined in the policy, a normal user will not be able to see the sensitive columns (and will in fact receive an error saying that the column doesn't exist if they try to select it explicitly). In comparison, a manager will have full access to all columns.



Employee_No	Dept	Title	First Name	Last Name	Salary	Commission
868763	A01	Mr	John	Doe	$50,500	$10,000
748939	A01	Miss	Jayne	Keele	$55,000	$4,300
...		...	...	...	...	...

Row / Column Level Access Policy

```
SELECT LAST_NAME,
       SALARY
FROM T1
WHERE …
```

```
SELECT *
FROM T1
WHERE …
```

Error – column SALARY doesn't exist

Employee_No	Dept	Title	First Name	Last Name
868763	A01	Mr	John	Doe
748939	A01	Miss	Jayne	Keele
...		...	...	...

Normal User

```
SELECT LAST_NAME,
       SALARY
FROM T1
WHERE …
```

```
SELECT *
FROM T1
WHERE …
```

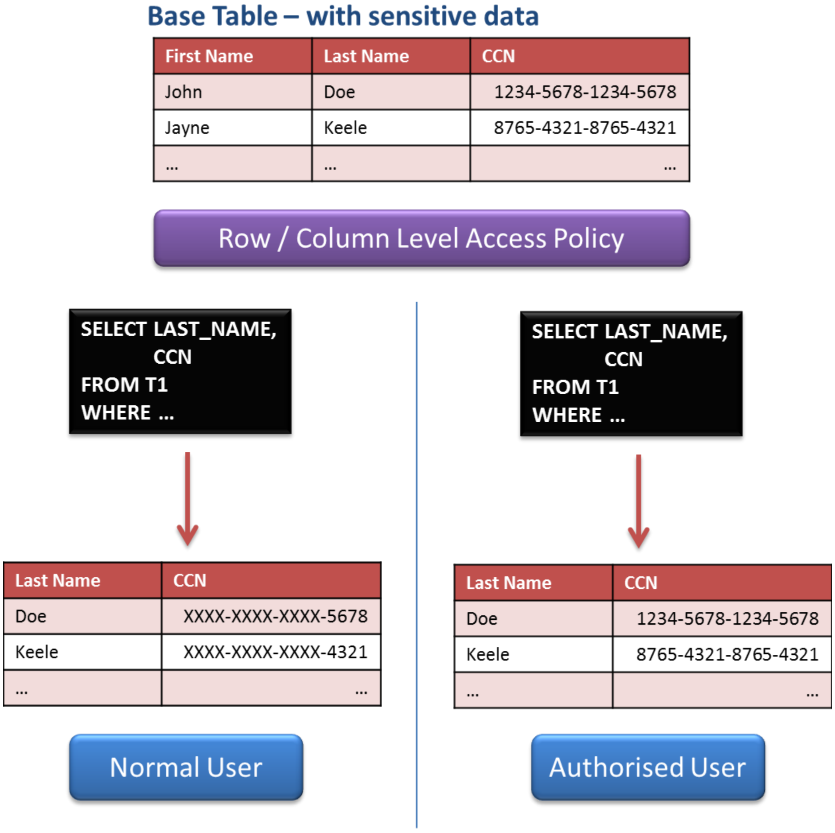
Last Name	Salary
Doe	$50,500
Keele	$55,000
...	...

Employee_No	Dept	Title	First Name	Last Name	Salary	Commission
868763	A01	Mr	John	Doe	$50,500	$10,000
748939	A01	Miss	Jayne	Keele	$55,000	$4,300
...		...	...	...	...	...

Manager

Figure 25 – DB2 10 Column Policy

Extensions to this same approach provide data masking capabilities using the same policies. In the example shown in Figure 26 below, all digits of a credit card number except the last four are masked with an "X" character for normal users, but authorised users are automatically allowed the see the full number.



Base Table – with sensitive data

First Name	Last Name	CCN
John	Doe	1234-5678-1234-5678
Jayne	Keele	8765-4321-8765-4321
...	...	...

Row / Column Level Access Policy

```
SELECT LAST_NAME,
       CCN
FROM T1
WHERE …
```

Last Name	CCN
Doe	XXXX-XXXX-XXXX-5678
Keele	XXXX-XXXX-XXXX-4321
...	...

Normal User

```
SELECT LAST_NAME,
       CCN
FROM T1
WHERE …
```

Last Name	CCN
Doe	1234-5678-1234-5678
Keele	8765-4321-8765-4321
...	...

Authorised User

Figure 26 – DB2 10 Column Masking

These enhancements finally allow security logic to be full separated from application logic, providing significant additional flexibility and ensuring consistent security behaviour regardless of the mechanism used to access the data. Subsequent changes to security can then be implemented independently without having to change any application logic, reducing implementation cost and improving developer productivity.

Improved Dynamic Schema Change

The enhancements to DB2's dynamic schema change capabilities (as described on page 16) are also relevant in terms of resilience. Each of the new capabilities eliminates a requirement to unload, drop, recreate and reload data that would otherwise be required in order to implement the schema change. This in turn increases data availability while also increasing resilience due to the lower likelihood of human error leading to temporary or permanent loss of data.

Temporal Data

As described in the section on Temporal Tables on page 7, this important new facility allows DB2 to maintain a full transaction history for specified tables and allows developers to code elegant SQL queries to determine the state of a given row at any point in the past.

In addition to addressing fundamental line-of-business requirements for maintaining a history of a given business object, temporal tables can also help to address regulatory requirements. The ability to more rapidly respond to such requirements and deliver the necessary audit/history trail, while expending fewer man-hours in coding and testing applications, will be a valuable business benefit for many large enterprises.

Growth

Supporting workload growth, both in terms of increased scalability/throughput and catering for new types of workload, remains a key focus area for the DB2 for z/OS development team.

Despite the recent global economic slowdown, large DB2 customers around the world continue to experience ever-increasing transaction and data volumes with DB2 for z/OS being asked to shoulder much of the load. Each release of DB2 must continue to significantly expand DB2's limits in order to cope with this demand.

At the same time, DB2's role as an enterprise data server means that it is called upon to support an ever more diverse set of workloads. New classes of data such as XML place unique demands on the database engine, while an increasing focus on hosting business intelligence and advanced analytics applications on the System z platform is driving a new generation of hardware and software solutions.

This section addresses the new DB2 10 features aimed at supporting these demands.

> ***"With the scalability improvements in DB2 10, we expect to be able to quickly reduce our production data sharing group from 20 members to 15. We will also save some CPU and storage from removing the five DB2 systems, and we will have to spend a lot less time monitoring our virtual storage."***
>
> **Paulo Sahadi, Senior Production Manager, Information Management Division at Banco do Brasil.**

Increased Scalability Though Full 64-bit Exploitation

In an effort to drive down IT costs and deliver better value, customers everywhere are constantly trying to do more with less. They want fewer machines to manage, fewer databases to care for and they want to support larger and more complex workloads without an associated increase in IT staff. These drivers mean that there is constant pressure to increase the amount of work a DB2 system can handle.

Storage constraints remain the single biggest factor in limiting the scalability of a single DB2 system today. Each process that runs concurrently within that system requires some storage, so the more workload a given system is asked to handle, the higher the storage requirements.

In DB2 Version 8, IBM embarked upon a major project to transform DB2 into a 64-bit RDBMS, removing many of the addressability issues inherent in the previous 31-bit memory model (see Figure 27, below).

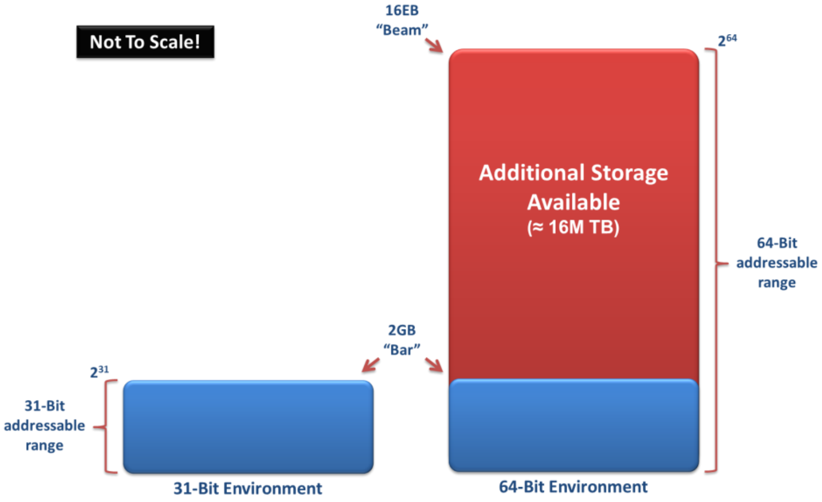


Figure 27 – 64-Bit Memory Model

DB2 10 delivers a spectacular increase in the number of threads that can be supported by a single subsystem – most customers will be able to achieve 5-10 times the number of concurrent connections compared to DB2 9.

DB2 Version 8 also moved several key storage areas above the 2GB "bar" into the newly-addressable memory space, relieving some of the storage constraints and allowing more workload per DB2 subsystem. DB2 9 increased scalability by a further 10-15% by moving another set of storage areas above the bar, but even with those enhancements most customers have been limited to running a maximum of 300-500 concurrent active connections within each DB2 system. As shown by Figure 4 on page 9, this often meant paying a performance and productivity penalty by increasing the number of DB2 systems needed to run a given workload.

In DB2 10, IBM has completed the bulk of the remaining work in the 64-bit migration effort, with 80-90% of the remaining DB2 storage structures moving above the bar. This has enabled a spectacular increase in the number of threads that can be supported by a single subsystem – most customers will be able to achieve 5-10 times the number of concurrent connections compared to DB2 9. In addition to the cost and performance benefits described on page 9, this vastly improves DB2's ability to support very high-volume workloads. However, the ability to run such a large number of threads within a single DB2 system is bound to expose new bottlenecks, and IBM has already begin to address these (see DB2 Catalog Enhancements on page 33 and Other Growth Enhancements on page 33 for examples).

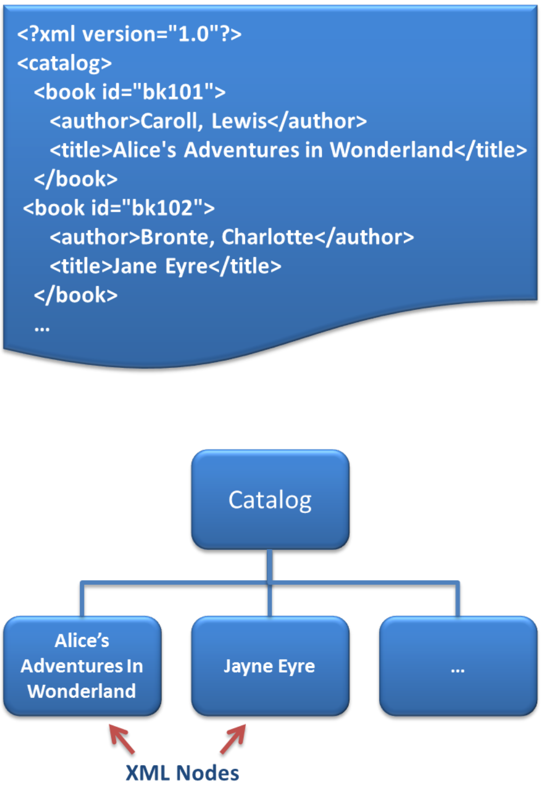
pureXML Enhancements

DB2 9 introduced pureXML to DB2 for z/OS: a major new feature that allowed XML documents to be stored natively within DB2 and easily retrieved using the power of SQL and XPath. Some significant performance improvements have since been delivered via the DB2 9 service stream, but DB2 10 introduces several additional enhancements to pureXML functionality, including:

- **XML Schema Validation**. Although XML is a universal interchange format capable of representing almost any form of data, it is possible to define a schema that limits the structure and content of specific XML documents. The process of ensuing that a given XML document adheres to a given schema is known as schema validation, and DB2 10 includes a new built-in function to support this important task. The function is capable of automatically determining the relevant XML schema for validation, and makes schema validation 100% eligible for offload to a zIIP processor if available. This feature promises to improve developer productivity and reduce CPU costs for this important XML process.

DB2 10 provides support for partial XML document update, allowing individual nodes within a document to be added, changed or removed. This will significantly reduce the amount of log data written for updates to large XML documents, improving performance and reducing CPU and elapsed times.

- **Partial update**. The initial implementation of pureXML required that a complete XML document be replaced if it needed to be updated. No support was provided for updating just a part of the XML document, known as "nodes" – see Figure 28 below.



```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Caroll, Lewis</author>
    <title>Alice's Adventures in Wonderland</title>
  </book>
 <book id="bk102">
    <author>Bronte, Charlotte</author>
    <title>Jane Eyre</title>
  </book>
  …
```

Figure 28 – Sample XML Document

DB2 10 addresses this issue by providing support for partial document update, allowing individual nodes within a document to be added, changed or removed. This capability is illustrated in Figure 29 below.

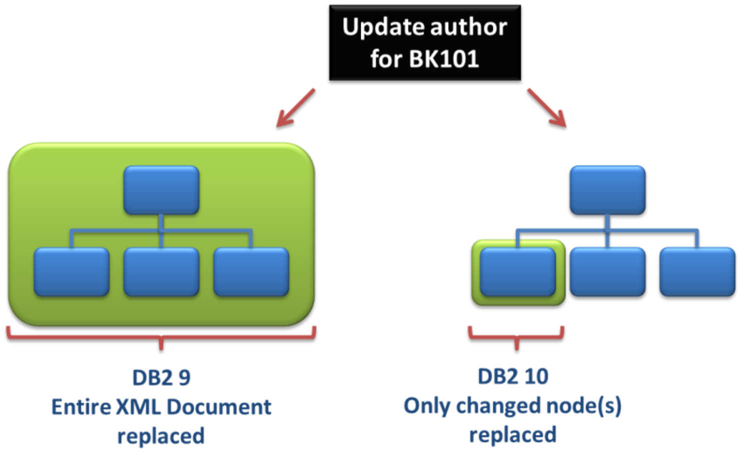


Figure 29 – Sample XML Document

This enhancement will significantly reduce the amount of log data written for updates to large XML documents, improving performance and reducing CPU and elapsed times.

- **Binary XML format.** DB2 10 introduces a new binary XML format that can be used for more efficiently passing XML documents between a client application and DB2. This can dramatically reduce the size of XML documents, and has resulted in CPU and elapsed time reductions ranging from 10% to 30% in internal IBM tests.

- **CHECK DATA support**. The CHECK DATA utility can be used against conventional DB2 data to ensure that the table data is consistent with any associated indexes and related tables. DB2 10 extends support to include XML documents, so that users can ensure that all nodes in an XML document are well formed, internally consistent and valid.

- **Other pureXML enhancements**. Additional pureXML features delivered in DB2 10 include improved indexing support for XML columns and enhanced support for User Defined Functions (UDFs) and triggers and stored procedures.

Collectively, these enhancements address the major functional and operational issues encountered by users of the initial pureXML support delivered in DB2 9, while improving consistency with other members of the DB2 family and improving both developer and DBA productivity.

Collectively, these enhancements address the major functional and operational issues encountered by users of the initial pureXML support delivered in DB2 9, while improving consistency with other members of the DB2 family and improving both developer and DBA productivity.

DB2 Catalog Enhancements

The Catalog[19] is one of the most vital components in any DB2 system. It contains a set of tables and internal structures that represent all of the metadata necessary for the subsystem to operate, and is used extensively by nearly every DB2 process from application programs to DBAs creating a new database.

DB2 10 includes some important enhancements to the Catalog, including:

> ***"The re-structured catalog in DB2 10 will greatly reduce the impact of our routine processes and allow us to run them during the week so we can be more responsive to the requirements of the business."***
>
> **Paulo Sahadi, Senior Production Manager, Information Management Division, Banco do Brasil**

- **Standard UTS format**. In previous releases, the catalog used a non-standard internal storage format and maintained links between the various tables using special internal pointers. DB2 10 converts some key Catalog structures to use the standard Universal Table Space Partition by Growth (UTS PBG) format introduced in DB2 9 (see Appendix A – DB2 9 Review on page 45 for more details). This change allows standard online REORG processes to be used against the catalog, improving performance and availability compared to previous releases.

  Contention between various processes needing to read/update the Catalog can cause significant operational disruption. As part of the same change, the converted Catalog tables have been spread across many more table spaces than before, and row level locking has been implemented to allow DB2 10 to support much higher levels of current access than was previously possible.

- **Maximum Catalog Size**. Many DB2 subsystems have to support hundreds of thousands of database objects. Each of these objects must be recorded in the DB2 Catalog and in some cases the 64GB limit for certain components such as the SPT01 table space can be a scalability limitation.

  DB2 10 addresses the 64GB limitation on the SPT01 table space by moving some large columns into LOB (Large Object) columns. This makes use of the inline LOB support added in DB2 10 (see Other Efficiency Enhancements on page 22), making the Catalog tables more readable and allowing more packages to be supported within a single DB2 system.

Other Growth Enhancements

DB2 10 includes a number of other enhancements designed to improve scalability and support future workload growth, including:

- **Latch and UTSERIAL contention relief**. A latch is an internal DB2 lock on an object, taken to ensure only one process updates a given resource at any one time. As overall workload increases latches can become an increasingly important factor in limiting DB2 throughput. DB2 10 includes some optimisations to reduce latching delays for many routine DB2 processes such as logging and accessing buffer pool pages. The UTSERIAL lock taken by

[19] Technically, this discussion applies to both the DB2 Catalog and the Directory (which contains internal representations of the objects in the Catalog). The term "Catalog" is used to represent both objects for brevity.

DB2 utilities to has also been removed, allowing for greater concurrency for compatible utilities.

- **Currently committed**. This new locking mechanism (described on page 20) should significantly increase the concurrency of some read-only workloads that are currently unable to make use of the older Uncommitted Read (UR) protocol.

- **Extended Address Volumes support**. With today's advanced storage management subsystems, traditional concepts such as physical DASD "volumes" are being transformed into purely logical constructs capable of supporting much higher capacities. z/OS 1.10, 1.11 and 1.12 introduced and subsequently enhanced support for Extended Address Volumes (EAVs), allowing up to 262,668 cylinders per logical volume (approximately 180GB). DB2 10 includes changes for EAV support[20], allowing fewer volumes to be defined and reducing administrative overheads.

- **TIMESTAMP Enhancements**. DB2 10 extends the precision of the TIMESTAMP data type to support fractions of a second from 0 up to 12 digits of precision (previously the precision was fixed at 6 digits). This provides greater flexibility, and the increased precision allows timestamps to be used as unique identifiers for certain tables with a much lower risk of duplicates being generated.

DB2 9 delivered some significant new functionality to support business analytics workloads. This emphasis continued within DB2 10, with a large number of new and enhanced features, both within the DB2 product itself and within the supporting tools and infrastructure.

A new SQL data type called TIMESTAMP WITH TIMEZONE has also been introduced in DB2 10. This data type is able to store time zone information in addition to the standard timestamp data, with DB2 handling the necessary conversion to UTC[21] for timestamp comparisons and arithmetic. . This enhances DB2's ability to support applications used in multiple time zones

Business Analytics

Traditionally, DB2 for z/OS was considered to be primarily an OLTP data server, with the DB2 for Linux, Unix and Windows variant (or other vendor's databases) being a more common choice for analytics and data warehousing duties. This approach is often dictated by cost concerns or historical inertia, but the superior resilience and scalability of the System z platform, combined with the increasing popularity of real-time warehousing, is leading many customers to re-examine this decision.

DB2 10 introduces support for moving sums, averages and aggregates, extending the OLAP SQL functionality previously delivered in DB2 9.

DB2 9 delivered some significant new functionality to support business analytics workloads, including improvements to indexing, query optimisation and SQL extensions. This emphasis continues within DB2 10, with a large number of new and enhanced features, both within the DB2 product itself and within the supporting tools and infrastructure.

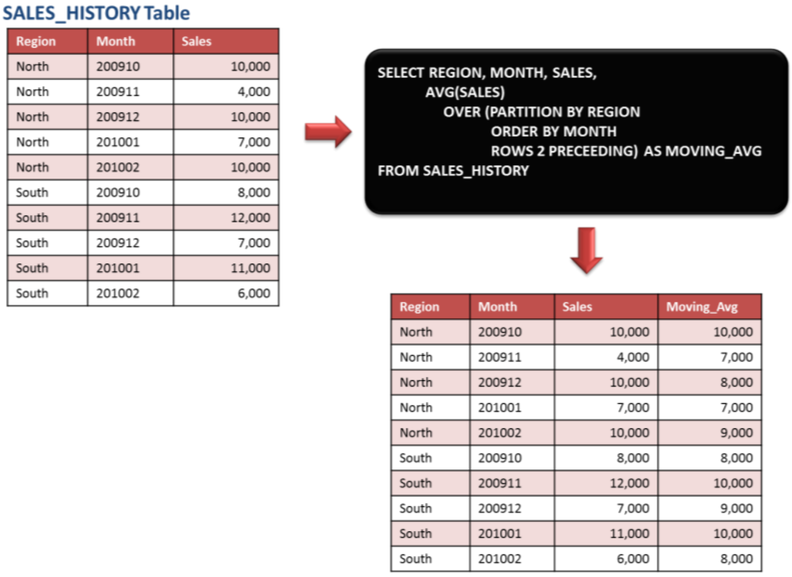
[20] APARs are also available for V8 and 9 to provide EAV support.

[21] UTC – Coordinated Universal Time. This is equivalent to GMT (Greenwich Mean Time) and provides a point of reference from which the time zone offsets for locations around the world can be expressed.

Enhanced OLAP SQL Functionality

Online Analytical Processing (or OLAP) is used to allow analysis of multi-dimensional sets of data to provide new business insights. For example, sales data (typically represented by a cube) might be analysed to determine if a particular branch, product or brand is performing well.

DB2 10 introduces support for moving sums, averages and aggregates, extending the OLAP SQL functionality previously delivered in DB2 9. Figure 30 below shows an example of some SQL using the new moving average functionality to display sales figures for each region in a sales history table.



SALES_HISTORY Table

Region	Month	Sales
North	200910	10,000
North	200911	4,000
North	200912	10,000
North	201001	7,000
North	201002	10,000
South	200910	8,000
South	200911	12,000
South	200912	7,000
South	201001	11,000
South	201002	6,000

```
SELECT REGION, MONTH, SALES,
        AVG(SALES)
            OVER (PARTITION BY REGION
                ORDER BY MONTH
                ROWS 2 PRECEEDING) AS MOVING_AVG
FROM SALES_HISTORY
```

Region	Month	Sales	Moving_Avg
North	200910	10,000	10,000
North	200911	4,000	7,000
North	200912	10,000	8,000
North	201001	7,000	7,000
North	201002	10,000	9,000
South	200910	8,000	8,000
South	200911	12,000	10,000
South	200912	7,000	9,000
South	201001	11,000	10,000
South	201002	6,000	8,000

Figure 30 – Moving Average Example

These new SQL constructs allow DB2 to more efficiently process common OLAP queries within the database engine, reducing the cost and elapsed time associated with extracting large volumes of data for analysis within the OLAP tool itself.

IBM Smart Analytics Optimizer

As DB2 for z/OS becomes more attractive as a host for large-scale analytics and analytics workloads, the requirement to deliver high performance with minimal administration and tuning overheads increases accordingly.

IBM has developed an innovative solution that combines DB2 for z/OS and a dedicated blade server capable of executing the complex queries typically found in analytics/analytics applications with fast and predictable response times.

In recognition of this trend, IBM has developed an innovative solution that combines DB2 for z/OS and a dedicated blade server capable of executing the complex queries typically found in analytics/analytics applications with fast and predictable response times. Known as the IBM Smart Analytics Optimizer[22], this solution is deeply integrated with DB2 10[23] and allows DB2 to offload eligible query components to the locally attached blade as shown in Figure 31 below.

[22] For further details on the Smart Analytics Optimizer, please see the IBM Redbook *Using IBM System z As the Foundation for Your Information Management Architecture* (2)

[23] Note that DB2 9 is also supported, with the necessary maintenance applied

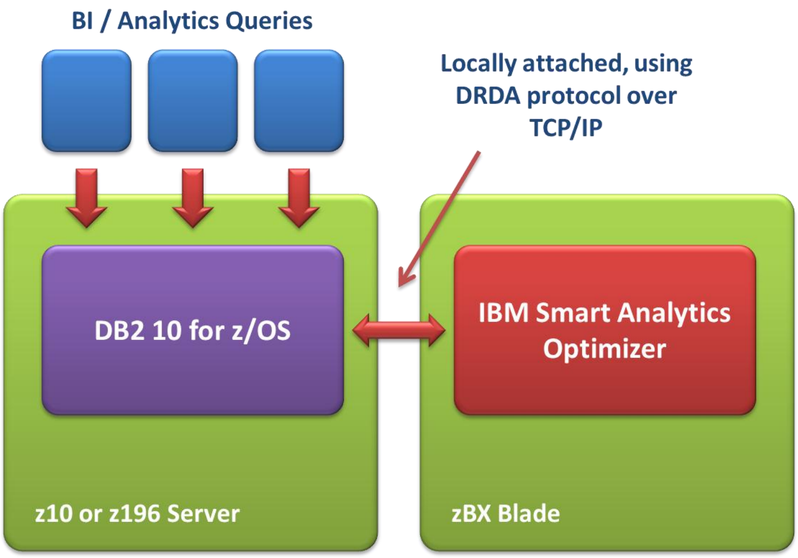

BI / Analytics Queries
Locally attached, using DRDA protocol over TCP/IP
DB2 10 for z/OS
IBM Smart Analytics Optimizer
z10 or z196 Server
zBX Blade

Figure 31 – Smart Analytics Optimizer Architecture

This architecture is able to deliver up to 10x performance gains for qualifying queries when compared to traditional DB2 processing.

This architecture is able to deliver up to 10x performance gains for qualifying queries[24] when compared to traditional DB2 processing. As the DB2 optimiser decides what work to offload to the appliance, no application changes are required in order to take advantage of the Smart Analytics Optimizer once it is made available.

Query Management Facility

IBM's Query Management Facility (QMF) tool is as old as DB2 itself, and provides a solid platform for executing many customer's analytics and analytics queries.

QMF 10 will be released at the same time as DB2 10 for z/OS. It provides many new or enhanced facilities, including:

- Integrated infrastructure, supporting the full spectrum of analytics capabilities, from table data editing and ad-hoc querying to graphical reporting and interactive visual dashboards.
- BI content that can be deployed to both workstation and browser-based users.
- Programming–free, drag-and-drop authoring model.
- Rich, graphical reports with a wide variety of output choices, including HTML, PDF, Excel and others.
- Interactive dashboards with ability to present data drawn concurrently from multiple heterogeneous data sources
- Business Intelligence and Reporting Tools (BIRT) report-format support.

[24] In the initial release, there are some restrictions on the type of queries eligible for offload to the Smart Analytics Optimizer blade. These include support for dynamic SQL only (no static SQL), SQL syntax and data type restrictions and a requirement for tables to be defined in advance to the SAO environment. Some or all of these may be lifted in a future release.

Other enhancements

A large number the DB2 10 enhancements described earlier in the paper will also directly benefit typical analytics / warehousing workloads. These include:

- **CPU Reductions**. The significant CPU reductions described on page 5 are directly applicable to analytics workloads, and should result in an immediate CPU/cost reduction of 5-10% for most environments.

- **Temporal tables**. This functionality is described on page 7, and should prove very valuable in warehousing / analysis environments, which commonly have to support a historical perspective.

- **Automatic statistics collection**. Most analytics environments consist of many hundreds or thousands of tables, many of which are growing rapidly over time. In this kind of environment maintaining up to date database statistics is essential to allow DB2 to continue to use the correct access path. The automatic statistics facilities described on page 11 will reduce the amount of manual effort required for this critical process.

- **Index INCLUDE**. Analytics environments have to support complex query workloads against large data volumes, which makes it necessary to have good indexing in place on key tables. The index INCLUDE enhancement described on page 12 could help to reduce the number of indexes needed in environments, with an associated reduction in the cost of regularly loading data into the warehouse.

Overall, these enhancements further enhance DB2's analytics credentials on the System z platform, and make it significantly more attractive from a cost and functionality perspective.

- **Buffer Pool Enhancements**. Many analytics environments require large buffer pools in order to support efficient access to large data volumes. The enhancements outlined on page 13 will reduce the costs associated with buffer pool access for analytics applications.

- **Optimiser Enhancements**. Analytics queries are frequently complex, and access large amounts of data. The DB2 optimiser enhancements described on page 17 and page 22 will help many such queries to complete more quickly, and with less CPU.

Overall, these enhancements further enhance DB2's analytics credentials on the System z platform, and make it significantly more attractive from a cost and functionality perspective. Preliminary IBM measurements show an average 20% CPU reduction from a TPC-H[25] like workload using 150 queries.

[25] TPC-H is a standard ad-hoc, decision support benchmark. Please see http://www.tpc.org/tpch/ for further details.

Upgrading to DB2 10

This section outlines some of the considerations around the timing and structure of the DB2 10 upgrade process.

Skip Migration

Traditionally, IBM has only supported migration to a new release of DB2 from the release immediately preceding it (you could only migrate to DB2 9 from a Version 8 subsystem, for example). Up until now, the only recent exception to this rule was DB2 for z/OS Version 7, which supported direct migration from both Version 5 and Version 6. There were good reasons for IBM to offer this facility in 2001 when Version 7 became generally available, as “Y2K fever” had prevented many Version 5 customers from being able to migrate to Version 6 according to their usual timescales. Skip migration was a good way of helping those customers to catch up and reposition themselves to stay current as subsequent releases became available, but it wasn’t without its downsides: it required IBM to expend significant additional effort to develop and support, and left customers with twice the number of pre-requisites to manage and new function to absorb.

> ***With DB2 10 IBM will once again support skip migration, allowing Version 8 as well as DB2 9 systems to be migrated to DB2 10.***

With DB2 10 IBM will once again support skip migration, allowing Version 8 as well as DB2 9 systems to be migrated to DB2 10, and for very similar reasons to those that convinced IBM to support the jump from Version 5 to Version 7 way back in 2001. Despite DB2 9 containing some very attractive new function and being Generally Available since 2007, the recent global economic downturn has seriously impacted IT budgets and many customers still find themselves running DB2 Version 8 (or even earlier releases).

However, the availability of the skip migration feature does not mean that all DB2 customers currently on Version 8 should wait and go directly to DB2 10. As mentioned above, skip migrations do have some downsides in terms of increased complexity and risk, and the amount of planning and implementation effort required for a Version 8 to Version 10 upgrade is greater than that needed for a single-release (although still less than that needed for two separate upgrades).

If you’re on Version 8 today, the chances are that you’re missing out on some significant business benefits that DB2 9 for z/OS could provide, including some modest CPU savings for most workloads. If you need to move from Version 8 before you are ready for DB2 10, then moving to DB2 9 provides solid value now.

Skip migration is primarily intended to support three scenarios:

- Customers who are still running DB2 Version 7 and have yet to complete their upgrade to DB2 Version 8. Most of those will be planning an upgrade to Version 8 very soon, but Version 8 upgrade projects typically take at least 6-12 months to fully roll out. Once the Version 8 upgrade process is complete, those customers will be well placed to take advantage of the skip migration feature and go directly to DB2 10.

- Customers that have only just completed their Version 7 to Version 8 upgrade project and are unlikely to get business approval for another migration to DB2 9 so soon after the last one. These customers may want to consider staying with Version 8 for now and migrating directly to DB2 10 during the next 18-24 months.

- Customers who have usually migrated quickly to new versions, but did not move to DB2 9. If customers usually start working with new versions in the first year, they have tests and processes for dealing with the normal maturity pattern. These customers may be able to skip DB2 9 and deliver more value in a shorter timescale.

This decision making process has been summarised in Figure 32 below.

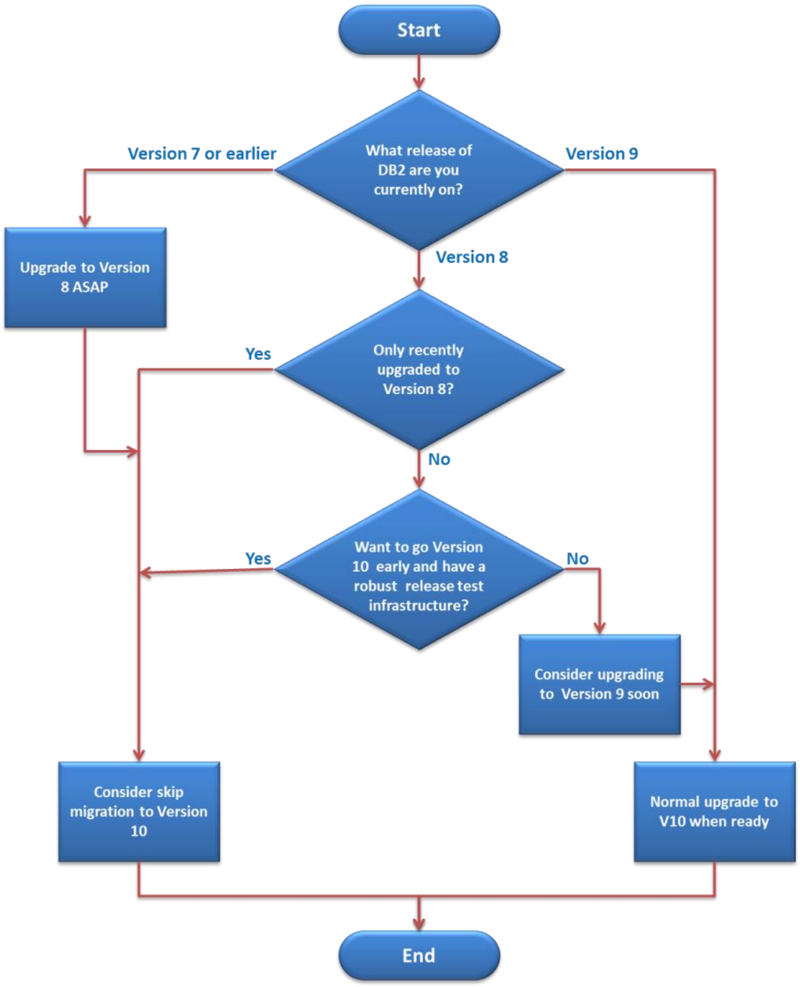

Start
What release of DB2 are you currently on?
Version 7 or earlier
Version 9
Upgrade to Version 8 ASAP
Version 8
Only recently upgraded to Version 8?
Yes
No
Want to go Version 10 early and have a robust release test infrastructure?
Yes
No
Consider upgrading to Version 9 soon
Consider skip migration to Version 10
Normal upgrade to V10 when ready
End

Figure 32 – DB2 10 Upgrade Decision Process

In order to reduce the financial impact of running two versions of DB2 concurrently, IBM may offer eligible customers a Single Version Charging[26] option for a period of up to 18 months when migrating directly from Version 8 to DB2 10.

[26] Single Version Charging (SVC) allows a customer to pay for only the most current version of program while running both the current and previous versions on the same CPU. This allows a customer a fixed period to upgrade from the previous version to the current version of the program while only paying for the newer version

Whichever path is taken, it is important for DB2 customers to ensure that they get good advice on the advantages and disadvantages of each option. In particular the effort involved in a skip migration should not be underestimated, and the “culture shock” for developers and support staff asked to take on two releases worth of new function in a single bound can be considerable.

DB2 Version 8 Support

IBM recently announced that support for DB2 for z/OS Version 8 will formally end on 30th April 2012. For many DB2 customers that have strict policies prohibiting the use of out-of-support software for mission-critical systems, this will clarify the timescales for their DB2 10 upgrade process.

IBM recently announced that support for DB2 for z/OS Version 8 will formally end on 30th April 2012. For many DB2 customers that have strict policies prohibiting the use of out-of-support software for mission-critical systems, this will clarify the timescales for their DB2 10 upgrade process.

Using the decision process in Figure 32 above, organisations just about to migrate to DB2 Version 8 have around 18 months from the date this paper was written to complete the Version 8 upgrade and then plan and implement their skip migration to DB2 10. Given the typical timescales involved with upgrade projects, these organisations need to begin planning the Version 8 upgrade immediately if they are going to complete the DB2 10 upgrade before Version 8 support is withdrawn in April 2012. Extended service for 6 months is being offered to customers who are skipping from Version 8 to DB2 10.

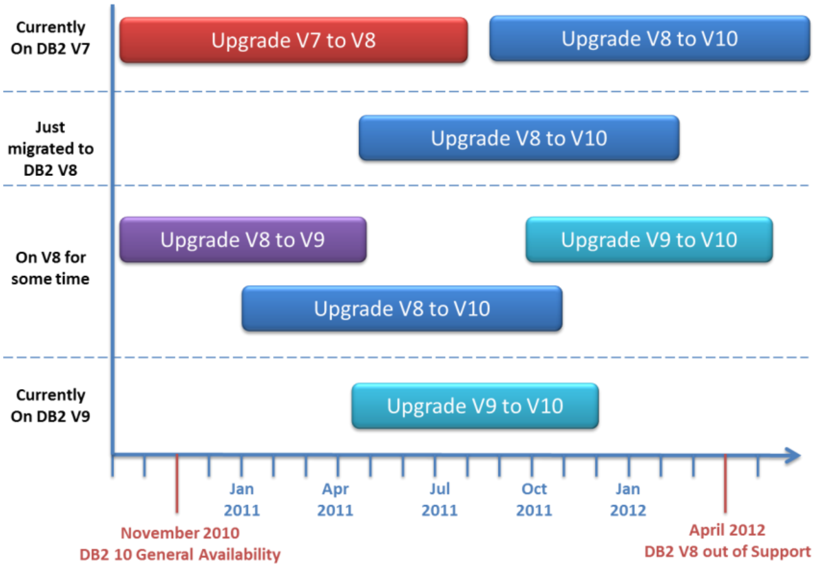


Figure 33 – Possible DB2 Upgrade Timescales

Figure 33 above shows some possible upgrade timescales for DB2 customers based on their current position.

DB2 10 Customer Case Studies

The section that follows is based upon interviews with some of the organisations that participated in the DB2 10 beta programme. Based on their early experiences with the product, they outline the business benefits they expect by exploiting the features in the new release.

Case Study 1 – Banco do Brasil

The Banco do Brasil is the oldest and largest active bank in Brazil, and one of the longest established financial institutions in the world. However, their IT infrastructure is definitely of the modern variety, and DB2 for z/OS sits right at the heart of its critical banking systems. With over 40 million clients in 22 countries around the world, coping with ever-increasing transaction volumes and an absolute need for 24x7 availability is a constant technology challenge.

Despite the virtual storage savings provided by DB2 9 for z/OS, the bank currently has to employ a 20-way data sharing group to handle its main production workload, with the DB2 members being spread over six physical z10 servers. Even with 20 DB2 subsystems sharing the load, constant and careful virtual storage monitoring is required in order to maintain availability.

> ***“With the scalability improvements in DB2 10, we expect to be able to quickly reduce our production data sharing group from 20 members to 15. We will also save some CPU and storage from removing the five DB2 systems, and we will have to spend a lot less time monitoring our virtual storage.”***

“With the scalability improvements in DB2 10, we expect to be able to quickly reduce our production data sharing group from 20 members to 15”, said Paulo Sahadi, Senior Production Manager, Information Management Division at Banco do Brasil. With DB2 10 able to handle 5-10 times as many threads as the previous version, the upgrade will immediately give the bank some much-needed room for future workload growth while simultaneously reducing their data sharing overhead. “We will also save some CPU and storage from removing the five DB2 systems, and we will have to spend a lot less time monitoring our virtual storage”, added Paulo.

Banco do Brasil also expects to see some significant business benefits from DB2 10’s parallel index update enhancement, as many of their most demanding workloads involve heavy concurrent insert activity against tables with several indexes. “This will reduce the elapsed time for some of our most critical processes” said Paulo, “delivering a more responsive application and allowing us to sustain greater throughput.”

Today’s banking customers expect to be able to access their accounts 24 hours a day, so maintaining data availability is a very important consideration when scheduling routine housekeeping activities. In order to avoid any potential impact on business transactions, Paulo’s team is currently forced to defer the execution of many activities such as schema changes and REBINDs, and run them in tightly defined windows at weekends. “The re-structured catalog in DB2 10 will greatly reduce the impact of these routine processes and allow us to run them during the week so we can be more responsive to the requirements of the business” added Paulo.

Banco do Brasil has an unwavering focus on maintaining the availability and stability of their critical IT systems. In a testament to the robustness of the beta code and the compelling business advantages offered by the new release, the bank is considering beginning the DB2 10 upgrade process as early as the second half of 2011, with the intention of moving all systems to DB2 10 New Function Mode within another 12-18 months.

> ***“DB2 10 enhances our ability to support our rapidly growing workloads while delivering some very valuable new function with immediate business benefits.”***

Overall Paulo is pleased with what he has seen so far in the bank’s beta testing. “DB2 10 enhances our ability to support our rapidly growing workloads while delivering some very valuable new function with immediate business benefits.”

Case Study 2 – BMW Group

As one of the world’s most successful car manufacturers, the BMW Group is at the forefront of both automotive and IT innovation. DB2 for z/OS is a critical component of many of the company’s worldwide computer systems, from manufacturing to supplier management and customer ordering. In total the car company has around 130 DB2 subsystems, belonging to over 40 data sharing groups.

BMW completed their migration to DB2 9 in June 2009 and although they saw some significant benefits from the upgrade they also encountered some challenges with access path selection in that release. Philipp Nowak, BMW DB2 Product Manager involved in the upgrade, takes up the story: “With our DB2 10 testing so far, we have had quite a few surprises but all of them have been good. Every single SQL statement we have tested has been better or the same as our current optimal paths – we have yet to see any significant access path regression. We had to spend a lot of time tuning SQL with DB2 9, but we expect that to disappear when we upgrade to DB2 10.”

> ***“With our DB2 10 testing so far, we have had quite a few surprises but all of them have been good. Every single SQL statement we have tested has been better or the same as our current optimal paths – we have yet to see any significant access path regression. We had to spend a lot of time tuning SQL with DB2 9, but we expect that to disappear when we upgrade to DB2 10.”***

BMW has also been evaluating the concurrent INSERT enhancements in DB2 10, as this is an important part of many of their critical production workloads. “We have measured a 38% reduction in CPU and a 7% reduction in suspend time for some heavy insert workloads in a data sharing environment” said Philipp. “That’s a significant saving which provides immediate business benefit.”

Philipp is also expecting to see significant benefits from the z/OS large page support in DB2 10. “We use very large buffer pools – some of them up to 3.2GB in size”, he said. “Our systems have to support lots of different workloads at the same time including users connecting directly for example via Excel ODBC, so buffer pool performance is very important for us. We rely on efficient access to buffered data and any saving in the cost of accessing that data will be very beneficial.”

In common with many other organisations, BMW is constantly striving to minimise the costs of its IT operations. “We expect the virtual storage enhancements in DB2 10 to allow us to reduce in the production environment the number of DB2 group members we run”, said Peter Paetsch, an external DB2 Consultant. “That will reduce the amount of time we spend in monitoring virtual storage usage, and decrease our data sharing overheads too.”

What other DB2 10 features will BMW find particularly valuable? “The new catalog structure will help to reduce the operational pain of normal activities during the daily business such as package rebinds”, said Philipp. “Although we’re only in the early stages of investigating the technology, the pureXML enhancements in DB2 10 for special applications also look very useful" added Peter.

"Overall, we are very pleased with the added functionality and architectural enhancements, and are looking forward to this exciting release."

So, how would the BMW DB2 team sum up their experiences with DB2 10 during the beta program? "Overall, we are very pleased with the added functionality and architectural enhancements, and are looking forward to this exciting release".

Case Study 3 – Postbank

With 14 million domestic customers, 21,000 employees and total assets of €242 billion, Deutsche Postbank Group is one of Germany's major financial services providers and the leading multichannel bank in the German market. Its focus is on retail business with private customers, but it is also active in the corporate banking sector and performs back office services for other financial services providers.

Postbank uses the SAP for Banking solution to support its core banking activities, with DB2 9 for z/OS as the back-end data store. Currently, their production environment consists of nine DB2 data sharing groups, with two 4-way groups taking care of the heaviest accounts and savings workloads

Our main area of interest in DB2 10 is the CPU saving. With the optimisations in DB2 10, we are hoping to see a significant CPU reduction for our main SAP workloads, perhaps enough to defer our annual hardware upgrade cycle".

SAP for Banking uses a 3-tier architecture with all DB2 workload arriving via DRDA from the application servers so many of the DB2 10 CPU enhancements are expected to provide immediate benefit. "Our main area of interest in DB2 10 is the CPU saving" said Guenter Schinkel, a DB2 Systems Programmer who has been heavily involved in Postbank's beta testing. "With the optimisations in DB2 10, we are hoping to see a significant CPU reduction for our main SAP workloads, perhaps enough to defer our annual hardware upgrade cycle".

The scalability enhancements in DB2 10 are also being closely examined in Postbank. Consideration is being given to moving from a 4-way data sharing architecture to a 2-way configuration, but according to Guenter there are also other advantages: "In a failover situation we currently have to be very careful about the amount of work that gets directed to a given DB2 system. The increased scalability in DB2 10 will allow us to simplify our failover automation and remain confident that DB2 can handle the increased workload".

Guenter is also pleased with the new audit capabilities introduced in DB2 10. Unlike more traditional environments where DBAs get to explicitly create new DB2 objects, SAP environments often have tables being created by the system itself as part of a SAP transport mechanism. "This makes it difficult for us to quickly audit new tables", said Guenter, "as we have to ALTER the tables to enable the audit attribute and sometimes we have to wait weeks or months for a change slot. The new audit capabilities in DB2 10 will allow tables to be audited as soon as they are created, which is an obvious benefit for the business".

Summary

Even in the most favourable economic climate, businesses need to control costs and increase efficiency in order to improve their bottom line. In today's more challenging business environment this has become a key factor for the survival and success of enterprises of all sizes.

DB2 10 delivers significant "out of the box" benefits that many customers will be able to exploit with little or no additional effort. These include the most aggressive performance and CPU improvements of any DB2 release in the last 20 years, scalability enhancements to support ever-increasing workloads and productivity improvements to allow DB2 developers and support staff to do respond more rapidly to the demands of the business.

DB2 10 delivers significant "out of the box" benefits that many customers will be able to exploit with little or no additional effort. These include the most aggressive performance and CPU improvements of any DB2 release in the last 20 years, scalability enhancements to support ever-increasing workloads and productivity improvements to allow DB2 developers and support staff to do respond more rapidly to the demands of the business.

Collectively, these features deliver real and quantifiable business benefit, and many customers will be considering upgrading to DB2 10 much more quickly than they may have done for previous releases.

Appendix A – DB2 9 Review

DB2 is universally accepted as the premier database system for IBM's System z mainframe architecture.

Today, DB2 is universally accepted as the premier database system for IBM's System z mainframe architecture. Although other products do exist for this platform, DB2 sits at the heart of most of the business-critical mainframe IT applications that have been written during the last 20 years.

DB2 9 has been generally available since March 2007, and a significant amount of the DB2 worldwide workload is now running on that release. The scalability and reliability of IBM's zSeries platform makes it a very attractive choice for customer's high-volume, mission-critical applications.

In this section, we'll briefly review DB2 9 and look at some of the ways in which it helps to deliver competitive edge. For a more detailed review of DB2 9, please see the 2007 Triton White Paper ***DB2 9 for z/OS – Data on Demand*** (1).

Support for New Workloads

The demands being placed upon enterprise data stores are continually expanding. Many modern business applications have moved beyond the processing of traditional "structured" data, and now have to deal with new types such as XML and binary data (including images, audio and video). Databases have to handle these new data types while continuing to provide the same levels of performance, resilience and security that customers have come to rely on.

At the same time, IBM continues to evolve the mainframe platform and extend the range of applications it is able to efficiently support. Running ERP and Business Analytics applications on the system z platform are becoming increasingly popular as specific new function is added to DB2 (and other system software) to support them.

New features delivered in Version 9 in support of these new workloads include:

"This is not a bolt-on or band-aid approach, this is XML without compromise"

DB2 9 Beta Customer

- **Integrated XML**. The pureXML feature provides a fully-fledged XML storage engine within DB2, allowing XML documents to be easily stored in their native format while retaining DB2's traditional strengths for structured relational data. XPath[27] and SQL can be used interchangeably (or in combination) to query XML and relational data. This provides immediate flexibility and productivity advantages, with traditional application developers able to query both relational and XML data with SQL, while XML specialists are able to use the power of XPath to access the same data

- **Large object support**. DB2 for z/OS has supported large objects (LOBs) for some time, but some significant restrictions existed that made it difficult for many customers to make use of these features. Most of these restrictions have been addressed within DB2 9, with enhanced utility support, more efficient LOB retrieval and a revised locking model. These enhancements make DB2 9 a more attractive host for LOB data, and will allow

[27] XPath is a W3C standard for a language specifically designed for navigating through and accessing parts of an XML document.

a greater range of applications to take advantage of the resilience and security offered by DB2.

DB2 9 allows SQL stored procedures to be executed natively by DB2 – no compilation step is necessary, the requirement for a C compiler has been removed and performance will be enhanced by up to 80%.

- **Native SQL Procedures**. Stored procedures can provide a number of very significant benefits, including lower maintenance costs, better performance and enhanced security. DB2 9 allows SQL stored procedures to be executed natively by DB2, removing the need for a z/OS C compiler and improving performance by up to 80% compared to Version 8. Additional enhancements to versioning and deployment facilities for SQL stored procedures significantly broaden both the scope and appeal of SQL procedures, allowing them to be deployed more efficiently and executed more quickly and at lower cost than before.

"IBM and SAP have cooperated very closely on DB2 9 for z/OS and we look forward to supporting our customers with these new capabilities."

Torsten Wittkugel, Vice President of Database and Operating System Platform Development, SAP

- **Enhanced ERP Support**. DB2 for z/OS is well established as the premier data server for high-end ERP systems such as SAP, and Version 9 delivered a large number of enhancements specifically designed to improve support for these workloads. These included new ways of partitioning tables, improvements to table row formats and volume based utilities. Together, these enhancements will reduce DB2 CPU usage and improve DBA productivity and data availability. They further consolidate DB2's position as the preferred database for ERP applications such as SAP.

Today, DB2 for z/OS is considered to be primarily an OLTP data server, with other databases being a more common choice for data warehousing duties. This approach is often dictated by cost concerns or historical inertia, but there are some good reasons for customers to re-consider this position.

- **Business Analytics**. Many customers are considering the advantages of DB2 for z/OS as an analytics platform, and DB2 9 provided some welcome enhancements in this area. These included indexing improvements, extensions to SQL and optimiser enhancements, all of which significantly enhanced DB2's analytics credentials on the System z platform, and underlined IBM's renewed emphasis in this area

Streamlined Security and Compliance

Compliance is one of the major business challenges facing today's enterprises, and nowhere is that challenge more keenly felt than within IT. Regulations such as the Sarbanes-Oxley Act, HIPAA and Basel II place serious and demanding obligations on a company to protect and secure the sensitive information held within its IT systems, and to be able to prove that they have done so.

The System z platform has a long history of providing a safe and secure environment for data, but these new regulations have raised requirements for new features within the operating system and other system software. DB2 9 has responded to these requirements with the following enhancements.

"Network trusted contexts is a very big deal for us. It will allow us to lock in access permissions to the Websphere servers and prevent the application user from being used elsewhere."

DB2 9 Beta Customer

- **Network Trusted Contexts and Roles**. DB2 9 introduces the concept of network trusted contexts: pre-defined incoming connections which originate from a trusted location and are therefore able to bypass normal authentication checks and reduce overhead. In the meantime, normal DB2 security rules will remain in place for non-trusted connections. This approach provides the best of both worlds, with robust security for non-trusted connections and good accountability with no performance compromises for trusted connections. Another new DB2 9 feature allows a set of DB2 privileges to be grouped into a construct known as a role. The combination of these two new features provides significantly enhanced security, better auditability and improved performance when compared to previous versions of DB2.

- **Enhanced Auditing.** The ability to audit changes made to data, and to the access given to that data, is a key aspect of many compliance regulations. DB2's audit capabilities have also been enhanced to encompass the role concept described above. DB2 9 also makes it possible to be much more selective when starting audit (and other) traces, allowing tracing to be confined to those areas where it is required and reducing the volume and overhead of trace data.

Instead of triggers remove the requirement to allow any sort of direct access to sensitive tables, thereby improving security.

- **Instead-Of Triggers.** A commonly-used technique for limiting application access to sensitive tables in DB2 is to use a view on the table that excludes the sensitive information, and only provide the application with access to the view. This approach works well for read access, but under some conditions it is not possible to perform inserts, updates or deletes through a view. Instead of triggers avoid this issue, by allowing an alternative action to be specified when an update operation is attempted against a view. This in turn removes the requirement to allow any sort of direct access to the sensitive table, thereby improving security

SSL support and encryption enhancements further strengthen DB2's security credentials, making it even more difficult to eavesdrop on, tamper with or forge messages passing to and from DB2 from a remote location.

- **SSL Support & Encryption.** One of the great strengths of DB2 for z/OS has always been its ability to rapidly exploit advances in the underlying hardware and operating system. DB2 9 continues this trend by supporting the use of System z disk and tape controllers to encrypt data. Many organisations are required to encrypt their DB2 data for compliance reasons. Using the intelligence built into the new generation of storage hardware to perform these encryption functions can significantly reduce mainframe CPU costs for many clients.

Reducing Total Cost of Ownership

Despite its well-understood scalability and resilience advantages, the System z platform needs to continue to demonstrate that is represents good value for money in today's competitive IT environment. Reducing the total cost of ownership (TCO) for System z applications is therefore an ongoing theme, and DB2 9 introduced a number of important advances in this area.

Converting to clone tables will effectively remove an outage and allow us to keep the data available at all times. And of course, we can dispense with the tricks and use a well-documented DB2 function which makes our environment easier and cheaper to support

DB2 9 for z/OS beta customer

- **Clone tables**. In today's high-availability environments, some processes can cause an unacceptable amount of disruption to applications needing to access the data. One solution to this problem is to create a copy of the table, keep it in step with the original and temporarily allow applications to access the table copy when disruptive operations are being run against the original. This process is effective, but requires a large amount of design and implementation effort on behalf of the application developer and DBA. DB2 9 introduces direct support for "cloned" tables, allowing the DBA to create an exact copy of a table with a simple command, and allowing application access to be easily switched between the original and the clone.

- **Optimization Service Center**. Business pressures often lead to applications being written and implemented very quickly, with little time available for performance testing and tuning. The Optimization Service Center (OSC) is a no-charge GUI workstation tool that is offered as part of the optional DB2 Accessories Suite for z/OS. It provides a host of facilities to allow a DBA to identify and analyse problem SQL statements and perform various tuning tasks.

The Optimization Service Center is a no-charge GUI workstation tool that provides a host of facilities to allow a DBA to identify and analyse problem SQL statements and perform various tuning tasks

- **Universal table spaces**. The “partition by growth” feature described earlier allows DB2 to manage disk space growth for some tables, removing the requirement for the database administrator to closely monitor and manage it themselves.

- **Automatic object creation**. Unlike many other database implementations, prior versions of DB2 for z/OS used to insist on the DBA explicitly creating all of the pre-requisite DB2 objects (such as databases as table spaces) before a table could be created. DB2 9 improves compatibility with other database systems and reduces DBA workload by allowing these objects to be automatically created by DB2.

- **Database roles**. This new feature will allow a pre-defined set of DB2 authorities to be easily allocated to individuals and removed when no longer required, with associated savings in security administration effort.

- **Dynamic schema change**. DB2 Version 8 introduced some major enhancements to allow database structures to be altered dynamically. This significantly improves data availability, but also improves DBA productivity as complex scripts to drop and recreate database objects can be replaced by a single command. DB2 9 further enhances these abilities by allowing yet more changes to be made online.

To ability to dynamically alter database structures significantly improves data availability, but also improves DBA productivity. DB2 9 allows yet more changes to be made online, such as renaming tables and columns.

- **Recover to consistent point**. DBA productivity is a critical factor during application or DB2 system recovery, where every minute taken to recover can translate directly into lost revenue. One of the most time-consuming tasks during the recovery process is to identify a “consistent point” to recover to – a moment in time when no updates were pending. DB2 9 includes some enhancements to the recover utility to allow a consistent recovery point to be automatically selected by DB2 when the DBA requests a data recovery. This speeds up the recovery process and makes it less prone to error.

- **SQL Merge**. A common requirement when updating a database is to perform a merge operation, where source data is used to either update an existing record in the database if it already exists, or insert a new record if it does not (this is also known as an “upsert” – a combination of update and insert).The new MERGE SQL statement introduced in DB2 9 will perform this process entirely within DB2, improving developer productivity and enhancing the performance of the application, as shown in the diagram below.

The new MERGE SQL statement introduced in DB2 9 will improve developer productivity and enhance the performance of many applications.

- **Truncate table**. Under some circumstances, it is necessary for an administrator to clear down a DB2 table and remove all data from it. A new TRUNCATE TABLE command allows this to be done more efficiently than before.

- **SQL procedures**. The enhancements to SQL procedures will make it feasible for many more clients to begin writing their stored procedures in this language. As SQL procedures do not need any form of program preparation, they are quicker to prepare and prototype than those written in conventional languages.

- **Index on expression**. The new ability to create an index on an SQL expression has the potential to transform the performance of many queries, and is expected to one of the most valuable performance enhancements within DB2 9.

> ***"Someone must have put a turbo burner into DB2 9. Our LOB test case under DB2 9 takes only 50 to 60% of the CPU time DB2 Version 8 needs for the same work, and I see also big improvement in elapsed time"***
>
> ***DB2 9 Beta Customer***

- **Large Object enhancements**. The use of large objects (LOBs) within DB2 for z/OS is increasing rapidly, and a significant amount of effort has been expended by IBM to improve the performance of queries accessing LOBs in DB2 9. The way in which LOBs are locked to enforce consistency has been completely overhauled, and some other enhancements have been implemented to make the handling of large LOBs much more efficient.

- **Optimisation enhancements**. The optimiser enhancements previously described will result in significant performance benefits for other types of applications too.

- **INSERT enhancements**. Many applications need to be able to insert large volumes of data into DB2 tables in a limited amount of time. Several enhancements have been made to DB2 index structures and insert processing to speed up mass insert operations. Initial IBM testing has shown CPU reductions of up to 20% as a result of these changes,

- **Utility enhancements**. Utilities are a mundane but vital part of DB2, allowing the DBA to backup, restore, reorganise and maintain critical business data. IBM has enhanced most of these utilities in DB2 9, with CPU reduction of between 10% and 70% being measured internally by IBM.

- **Sort enhancements**. Several improvements have been made to various sorting operations within DB2. Depending upon the workload, these have reduced CPU by up to 50% for some IBM test queries.

- **Reordered Row Format**. The reordered row format can yield significant performance benefits when performing large queries against tables with varying-length columns. Although of limited value for most typical online applications, some heavier queries have shown a performance boost of up to 50% as a result of this feature.

Appendix B – DB2 10 New Features by Implementation Effort

One of the most compelling features of DB2 10 is the number of enhancements that can deliver business benefit with little or no change being required to existing applications. This section lists each of the DB2 10 features covered in this document, categorised according to the amount of effort required to exploit them:

- **Minor Implementation Effort – Immediate**. These features are available immediately after upgrading to DB2 10, with no database or application changes required:

- **Minor Implementation Effort – Deferred**. These features do not require any database or application changes, but will only be available when the DB2 system has been placed in New Function Mode:

- **Significant Database Changes Required**. These features require some changes to be made to DB2 objects and structures (typically by the DBA), but no application changes. These changes are typically quicker and cheaper to implement/test than application changes.

- **Significant Application Changes Required**. These enhancements require some degree of application change in order to implement, and will therefore be the most expensive to implement and test.

Minor Implementation Effort - Immediate	
CPU Reductions (some features)	Page 5
Improved Scalability	Page 9
Automated Statistics (some features)	Page 11
Buffer Pool Enhancements	Page 13
Optimiser Enhancements	Page 17
Plan Stability	Page 24
Minor Implementation Effort – Deferred	
CPU Reductions (some features)	Page 5
Automated Statistics (some features)	Page 11
Dynamic Statement Cache Enhancements	Page 15
Enhanced Dynamic Schema Change	Page 16
MEMBER CLUSTER for Universal Table Spaces	Page 20
Backup & Recovery Improvements	Page 21

Enhanced Audit Capabilities	Page 25
More Flexible Administrative Authorities (some features)	Page 26
DB2 Catalog Enhancements	Page 33
Significant Database Changes Required	
New Hash Access Method	Page 10
Include Additional Columns in Unique Index	Page 12
More Flexible Administrative Authorities (some features)	Page 26
Improved Data Access Control	Page 27
Significant Application Changes Required	
Temporal Tables	Page 7
Currently Committed	Page 20
pureXML Enhancements	Page 31
Enhanced OLAP SQL Functionality	Page 35

Appendix C – Acknowledgements

The author would like to thank the following people for their invaluable contributions to this paper:

Maria Sueli Almeida	IBM Silicon Valley Lab
John Campbell	Distinguished Engineer, DB2 for z/OS Development
Ian Cook	DB2 for z/OS Product Introduction Manager
Paul Fletcher	DB2 for z/OS Specialist - BetaWorks
John Iczkovits	IBM Advanced Technical Skills (ATS) Americas
Terrie Jacopi	DB2 for z/OS Program Manager
Philipp Nowak	BMW DB2 Product Manager
Luca Mussato	Senior DB2 System Specialist, DB2 Beta Customer
Roger Miller	DB2 for z/OS Technical Evangelist
Surekha Parekh	Worldwide DB2 for z/OS Senior Market Manager
Peter Paetsch	DB2 Consultant
Frank Petersen	DB2 Systems Programmer, Bankdata
Dil Pratheek	Manager - DB2/z Performance & Engineering , Morgan Stanley
Mark Rader	IBM Advanced Technical Skills (ATS) Americas
Paulo Sahadi	Senior Production Manager, Information Management Division at Banco do Brasil
Guenter Schinkel	DB2 Systems Programmer, Postbank

Bibliography

1. **Stuhler, Julian.** *DB2 9 for z/OS - Data on Demand.* s.l. : Triton Consulting, 2007. White Paper.

2. **IBM International Technical Support Organization.** *Using IBM System z As the Foundation for Your Information Management Architecture.* s.l. : IBM ITSO, 2010.