DB2 11 for z/OS

Unmatched Efficiency for Big Data & Analytics

Julian Stuhler
Triton Consulting
September 2013

Contents

Abstract

This paper will provide a high-level overview of the major new features of IBM DB2 11 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 25 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 Champion. He can be contacted at **julian.stuhler@triton.co.uk**.

Executive Summary

DB2 11 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 11 can deliver.

Business Benefit Summary

DB2 11 for z/OS 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.

The potential business benefits from upgrading to DB2 11 can be summarised as follows:

Most customers can expect to see net DB2 CPU savings of up to 10% in their complex OLTP and update-intensive batch workloads when compared to DB2 10, and up to 40% for query workloads.

- **CPU Reductions**. DB2 includes a large number of new enhancements aimed at improving application performance and reducing CPU usage. Most customers can see net DB2 CPU savings of up to 10% in their complex OLTP and update-intensive batch workloads when compared to DB2 10. Complex reporting queries can see up to 25% DB2 CPU savings for uncompressed tables and up to 40% when running these queries against compressed tables, . These benefits can be achieved without any application changes being required[1]. Significant additional CPU savings are possible for other specific workloads, and with some application changes.

DB2 11 contains several major improvements designed to support ever-increasing data and transaction volumes.

- **Scalability Improvements**. DB2 11 contains several major improvements designed to support ever-increasing data and transaction volumes. Several of these allow greater exploitation of the larger memory footprints typically found in today's zEnterprise servers, while others such as the extended log record addressing resolve some fundamental architectural limitations and represent a major investment in DB2's future

New productivity features will reduce the effort required by developers and support staff to deliver robust DB2 applications.

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

Customers will be able to more rapidly exploit the benefits within the new release.

- **Improving time-to-value**. When upgrading to a new release, many customers find it challenging to implement the necessary pre-requisite work, and to schedule a dedicated change slot for the upgrade itself. DB2 11 includes significant enhancements that remove the hard dependency on making applications compatible with the new release prior to upgrading. It also reduces the operational impact of the upgrade process itself, making it easier for customers to perform the upgrade without impacting application availability. These changes will allow

[1] 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 itself.

customers to more rapidly exploit the benefits described above.

Conclusion

DB2 11 delivers additional out-of-the-box savings that are expected to match or exceed those within DB2 10 for some types of workload.

The end result is a hybrid solution which is capable of efficiently supporting very diverse workloads in a highly cost effective manner.

When DB2 10 for z/OS was released in 2010 many customers around the world quickly recognised the business value that could be exploited by upgrading. A combination of significant “out-of-the-box” CPU savings and compelling new functionality made this one of the most aggressively adopted new releases in DB2’s history.

DB2 11 builds upon this successful formula, delivering additional out-of-the-box savings that are expected to match or even exceed those within DB2 10 for some types of workload. At the same time, IBM continues to expand DB2’s formidable scalability and resilience to better support transactional workloads, while significantly improving the integration with specialised processing platforms such as BigInsights and the IBM DB2 Analytics Accelerator. The end result is a unique hybrid solution which is capable of efficiently supporting very diverse workloads in a highly cost effective manner.

Introduction

DB2 for z/OS in a Big Data World

Few IT professionals can have missed the big data phenomenon that has manifested itself in recent years. Industry publications and IT analysts have devoted a huge percentage of their output to the subject (creating a big data challenge all of their own in the process). There can be little doubt that the advent of new technologies and methods of customer and business interaction have created unique challenges for organisations wishing to create actionable insight from very large amounts of unstructured data. Innovative tools and techniques have been developed to cope with these "big data" challenges (and indeed some of them are discussed within this paper[2]).

However, beyond this somewhat narrow definition of big data many organisations have been dealing with the challenges of processing, maintaining and analysing ever increasing amounts of more traditionally structured data for many years. The inherent scalability and resilience of DB2 for z/OS and the underlying System z platform have proven to be a compelling combination for such applications, and IBM continues to invest in extending DB2's capabilities with each new release.

From transparent archiving to greater in-memory scalability through the use of 2GB page frames, DB2 11 for z/OS contains many new features specifically designed to help customers to address the challenges of managing traditional big data.

From transparent archiving to greater in-memory scalability through the use of 2GB page frames, DB2 11 for z/OS contains many new features specifically designed to help customers to address the challenges of managing traditional big data. A wealth of material exists on the technical changes within DB2 11, but finding descriptions of how those new features will improve your business results can be a challenge. The main body of 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 11 can deliver.

In the meantime, many customers are still running DB2 9 for z/OS (or earlier releases) and need to understand how DB2 10 can help their organisation. A brief summary of the business benefits offered by DB2 10 is provided in Appendix A – DB2 10 Review on page 57 of this document.

This is the fourth paper in this series, with previous editions highlighting the business value offered by DB2 for z/OS V8.1 [1], DB2 9 for z/OS [2] and DB2 10 for z/OS [3].

Please note: throughout the remainder of this document, all references to "DB2 9", "DB2 10" and "DB2 11" refer to the relevant release of IBM DB2 for z/OS.

[2] Please refer to the section on Hadoop and Big Data Support on page 41.

DB2 11 – The Database for Big Data and Analytics

In this section, we'll take a detailed look at the major features of DB2 11, 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 of these enhancements can deliver benefits "out-of-the-box" with little or no effort required to begin exploiting them, reducing the time-to-value for a DB2 11 upgrade. Please refer to Appendix B – DB2 11 New Features by Implementation Effort on page 65 for a breakdown of the effort required to exploit each new feature.

Many DB2 11 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.

This section is organised around the key DB2 11 themes:

- **Efficiency** – reducing cost and improving productivity
- **Resilience** – improving availability and data security
- **Business Analytics** – enhanced query and reporting

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 increasingly challenging business environment this continues to be a key factor for the survival and success of enterprises of all sizes.

This section examines the major DB2 11 enhancements that are aimed at delivering the highest efficiency for core 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.

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 very 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% (please see Figure 1 below). 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.

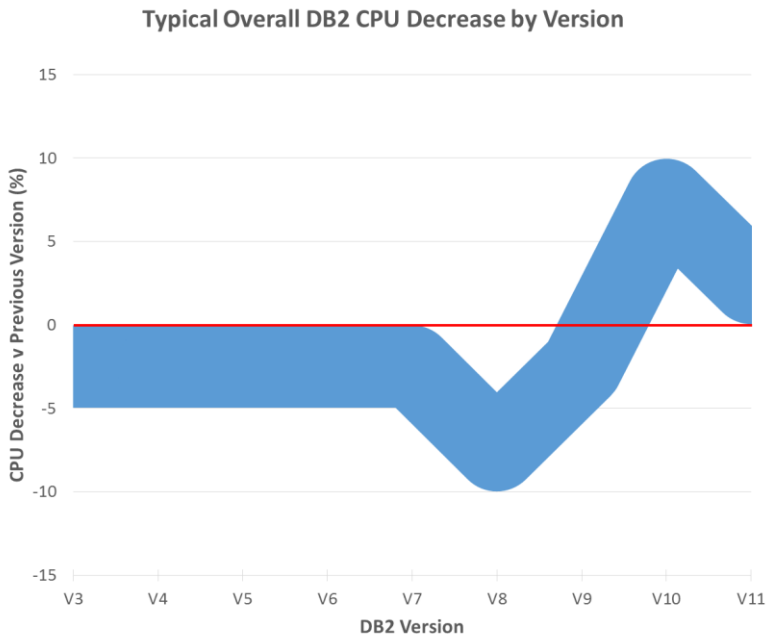

Figure 1 – Typical Overall CPU Decrease by Version

DB2 9 helped to redress the balance somewhat by delivering modest CPU improvements for many large customers, but the advent of DB2 10 completely changed the picture. IBM delivered the most aggressive performance improvements of any DB2 release in the last 20 years, with many customers seeing net CPU savings of 5-10% or more in their traditional DB2 OLTP workload without any application changes being required[3]. Unsurprisingly, these savings proved to be very popular and are consistently quoted as being one of the major reasons for customers to upgrade to DB2 10[4].

IBM has further developed the CPU reduction theme within DB2 11 with initial overall savings of up to 5% expected for simple OLTP workloads.

As Figure 1 shows IBM has further developed the CPU reduction theme within DB2 11, with initial savings of up to 5% expected for customers running simple OLTP workloads. Significantly higher savings are possible for complex OLTP and query workloads, as discussed below. As these improvements are due to internal DB2 code optimisation, they are available in DB2 11 Conversion Mode, without any application changes being required[3]. Additional CPU savings are possible once customers begin to use some of the other DB2 11 enhancements that require application change, as described elsewhere within this section of the paper. Please refer to Appendix B – DB2 11 New Features by Implementation Effort on page 65 for a summary of the work required to exploit specific enhancements.

Some workloads will benefit more than others from the performance enhancements offered by DB2 11, and Figure 2 below shows a breakdown of the anticipated CPU savings by workload type.

[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 itself.

[4] Please refer to Appendix A – DB2 10 Review on page 57 for a summary of the DB2 10 business benefits. Full details can be found in the previous version of this white paper [3].

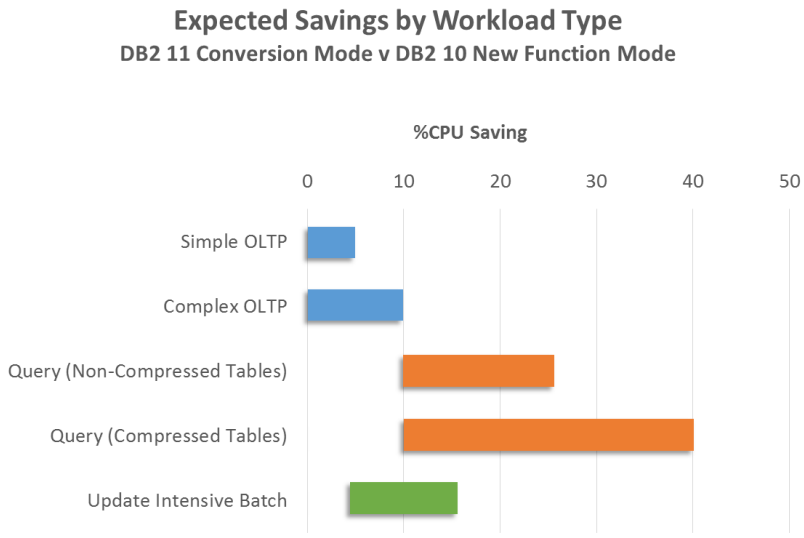

Figure 2 – Expected CPU Savings by Workload Type

The most significant benefits are expected to be seen within query workloads, with 10-40% CPU reductions anticipated.

OLTP workloads are also likely to benefit from the efficiency enhancements. Savings of up to 5% are expected for simple OLTP, with higher reductions up to 10% for more complex transactions.

Update-intensive batch workloads may enjoy CPU reductions of 5-15%.

The most significant benefits are expected to be seen within query workloads. Complex reporting queries can see up to 25% savings for uncompressed tables, and up to 40% for queries on compressed tables. Reporting queries with heavy sort processing may also see additional DB2 CPU savings.

Traditional OLTP workloads are also likely to benefit from the efficiency enhancements within DB2 11. Savings of up to 5% are expected[5] for simple OLTP, with higher reductions of up to 10% for more complex transactions. Finally, update-intensive batch workloads may enjoy CPU reductions of 5-15%.

Figure 3 and Figure 4 below depict some actual observed CPU reductions for sample workloads, run as part of IBM's internal performance testing for the new release. These figures are broadly in line with the high-level expectations detailed above.

[5] Note that achieving these savings may also require the new log address format to be implemented – please see Extended Log Record Addressing on page 32 for further details.

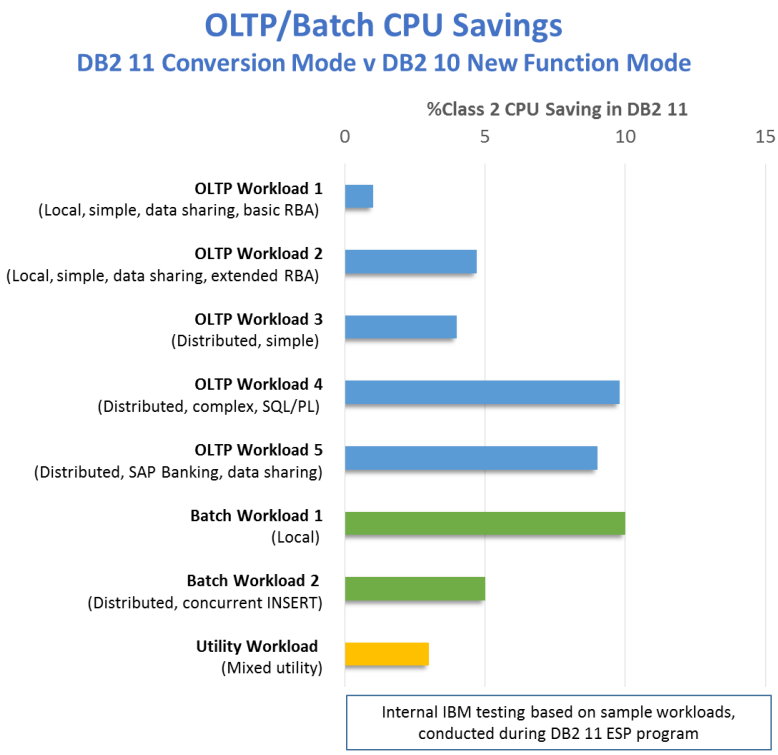

Figure 3 – Sample OLTP / Batch CPU Savings

"Virtual storage isn't a big limitation for us any more, but we expect the CPU savings in DB2 11 to provide the major business benefit for us."

BMW Group DB2 for z/OS Product Manager

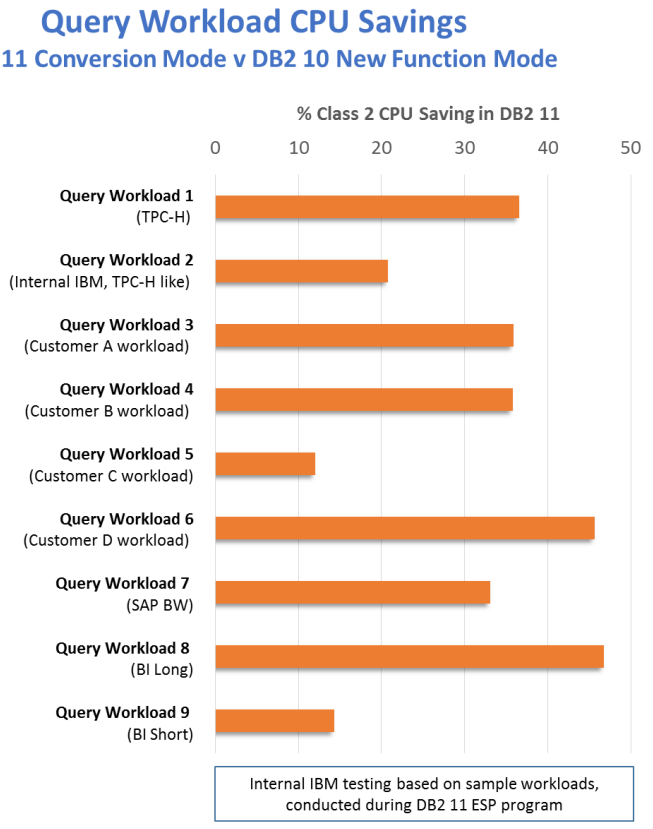

Figure 4 – Sample Query CPU Savings

The overall "out-of-the-box" CPU savings within DB2 11 are expected to be one of the major factors supporting the business case for upgrading to the new release.

The overall "out-of-the-box" CPU savings within DB2 11 expected to be one of the major factors supporting the business case for upgrading to the new release.

zEC12 Exploitation

IBM announced the latest generation zEC12 enterprise servers in August 2012, with up to 101 configurable processors per server, each running at an industry-leading 5.5GHz. In addition to an impressive list of general performance and capacity improvements over the previous-generation z196 enterprise servers, the zEC12 models include a number of features which will be specifically exploited by DB2 11.

In addition to an impressive list of general performance and capacity improvements over the previous-generation z196 enterprise servers, the zEC12 models include a number of features which will be specifically exploited by DB2 11.

- **2GB Page Frames**. DB2 10 for z/OS introduced support for 1MB "large page frames", an enhancement designed to reduce processing overheads for very large DB2 buffer pools by allowing z/OS to manage the underlying storage in fewer 1MB pieces rather than many more 4KB pieces.

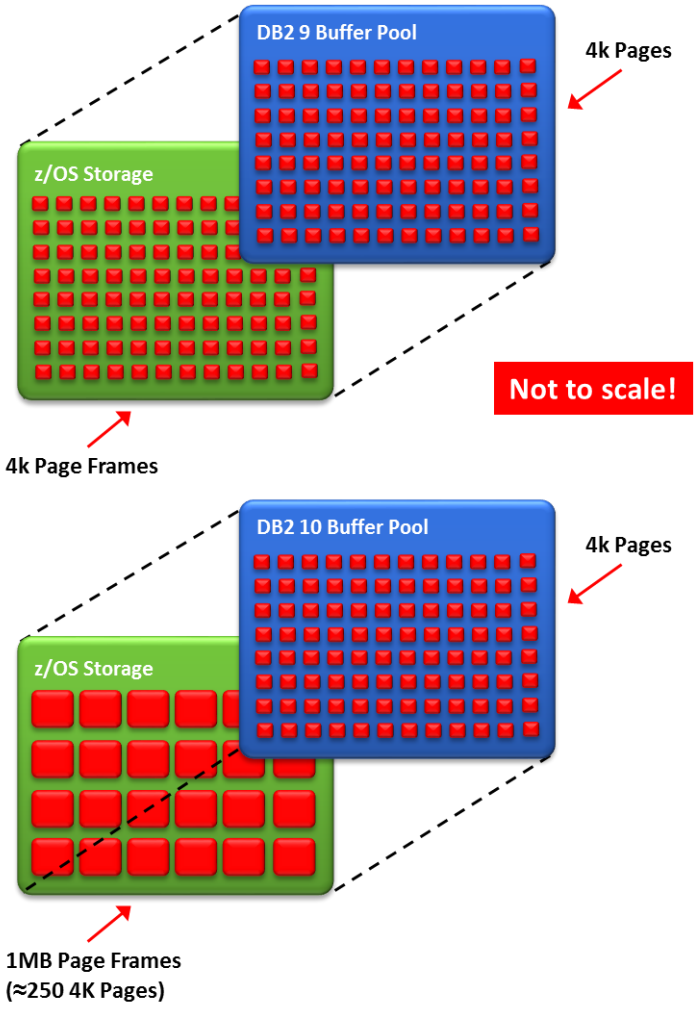

Figure 5 – DB2 10 Large Page Frame Enahncement

"We use very large bufferpools within all of our production DB2 systems, so we're anticipating further CPU savings when we can begin using 2GB frames."

BMW Group DB2 for z/OS Product Manager

Many customers with larger DB2 buffer pools were able to achieve CPU savings of up to 4% by exploiting this capability. However, as memory prices fall and workloads increase. DB2 buffer pools continue to increase in size and the overheads of managing even the larger 1MB page frames start to become significant.

In recognition of these trends, when running on an zEC12 server DB2 11 will support even larger 2GB page frames,

each of which will map onto more than half a million 4K pages as shown in Figure 6 below.

Those customers using very large DB2 buffer pools will see further CPU reductions by moving to 2GB page frames.

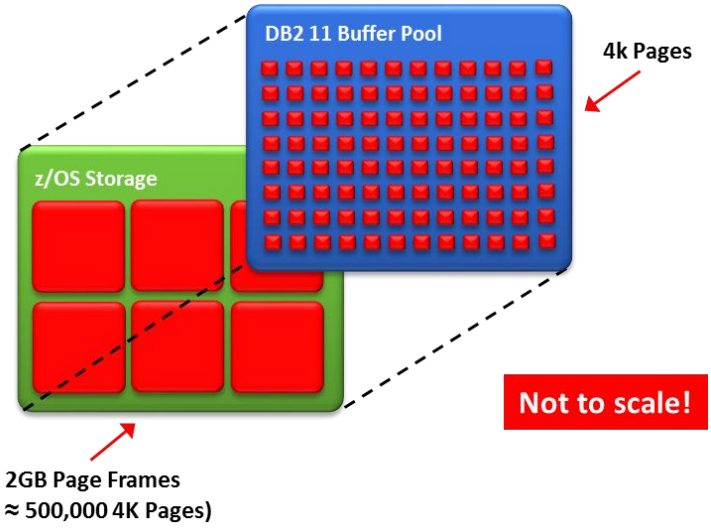

Figure 6 – DB2 11 Large Page Frame Enahncement

By moving early to support 2GB page frames IBM has recognised and eliminated an important future scalability issue.

Those customers using very large DB2 buffer pools will see further CPU reductions by moving to 2GB page frames. Other sites may not have sufficiently large pools for 1MB page frames to be a significant limitation today, but that situation will undoubtedly change in the future as buffer pool sizes continue to grow. By moving early to support 2GB page frames IBM has recognised and eliminated an important future scalability issue.

- **DB2 Code Using Large Page Frames**. As discussed in the previous section DB2 10 and DB2 11 have exploited 1MB and 2GB large page frames to allow more efficient handling of large buffer pools. However, despite the extensive use of large memory objects in the past few releases of DB2, the storage used for DB2 code (as opposed to the data held in buffer pools) remained backed by standard 4K page frames.

DB2 11 is able to utilise large page frames for DB2 code objects and log output buffers. This can reduce CPU consumption and lower operational costs.

DB2 11 is able to utilise large page frames for DB2 code objects and log output buffers, in addition to buffer pools[6]. This can reduce the z/OS overheads associated with DB2 code objects, reducing CPU consumption and lowering operational costs.

Application Compatibility

Many new releases of DB2 introduce enhancements or new features that require application and/or SQL code to be changed. These include additional SQL reserved words, changes to DB2 behaviour or processing and even changes to SQL return codes. Although IBM tries to minimise these “incompatible changes”, they cannot always be avoided – they may be required in order to ensure that DB2 adheres to evolving SQL standards, to support new functionality, or perhaps to address an earlier defect in the DB2 code.

[6] Support for running DB2 code in large page frames requires z/OS 2.1.

A major part of planning for a new release is to analyse the impact of these incompatible changes and arrange for the necessary amendments to be made to DB2 application code so it will continue to work as designed under the new release. This situation poses several challenges for DB2 customers:

- Analysis of the impact of incompatible changes can be difficult, time consuming and error-prone. Missing one or more of the required changes may result in application outages when DB2 is upgraded (or even worse, the application may continue to work but return unexpected results).

- Finding the necessary resource to undertake any required remedial work (and scheduling the associated change slots) can be expensive and require significant elapsed time. All of the changes within a given subsystem or data sharing group must be completed before the upgrade can commence, so a lack of resource within a single application team could impact the upgrade schedule for the entire environment.

These challenges are depicted in Figure 7 below.

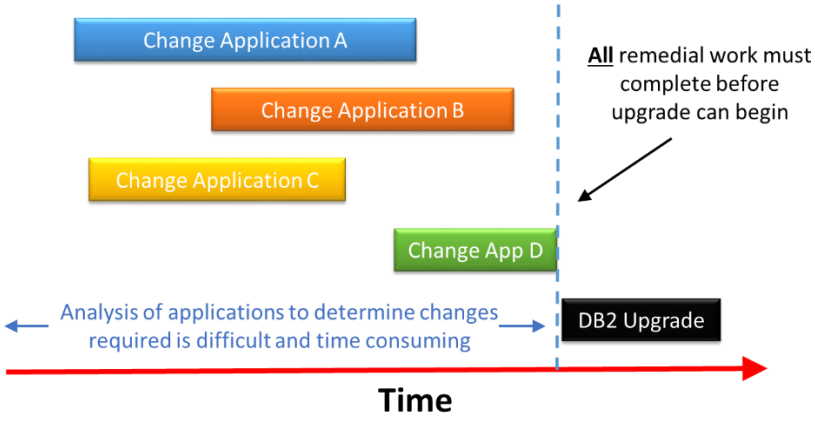

Figure 7 – Application Compatibility Issues

IBM has introduced new capabilities in DB2 11 that remove the hard dependency on all remedial work being conducted prior to a version upgrade, and allow the impact of incompatible changes to be more easily assessed.

In order to address these issues and allow customers to upgrade their DB2 systems with less effort and risk, IBM has introduced some new capabilities in DB2 11 for z/OS that remove the hard dependency on all remedial work being conducted prior to a version upgrade, and allow the impact of incompatible changes to be more easily assessed. These enhancements are summarised in Figure 8.

When upgrading to DB2 11, customers will be able to defer some or all of the remedial work for incompatible SQL DML and XML changes, and allow the DBA or developer to request that DB2 behaves the same as it did for DB2 10 on an application by application basis. Although the remedial work will still need to be done at some point, DBAs and developers are now free to schedule it a later date and in a more manageable, staged fashion that conforms to the requirements of the business (e.g. as part of a regular application release). In the meantime, other applications are able to benefit from the enhancements in the new release.

Furthermore, IBM has provided additional trace data in DB2 11 that can identify applications using incompatible SQL and XML statements after the version upgrade has been implemented. This

will allow DBAs and developers to identify applications requiring remedial work much more efficiently, and with less risk of some being accidentally missed.

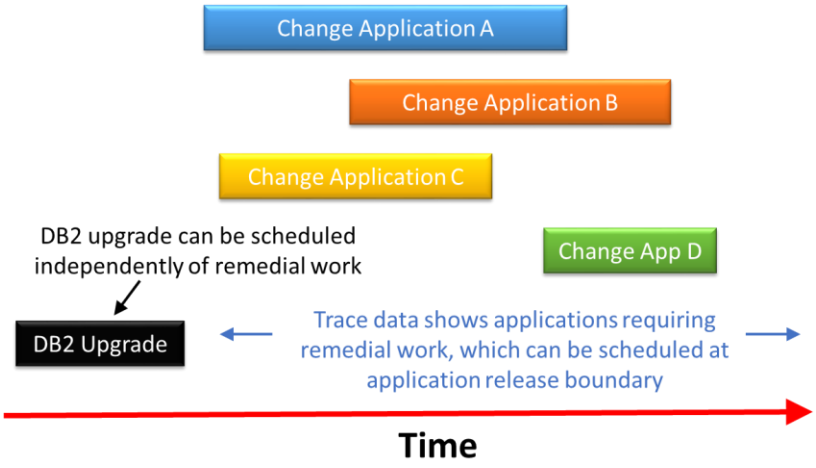

Figure 8 – DB2 11 Application Compatibility Feature

The DB2 11 Application Compatibility feature addresses many of the issues associated with handling incompatible changes in each new DB2 release. This capability should be a huge benefit to customers struggling to line up the necessary application development / DBA resource to address incompatible changes prior to DB2 11 being implemented.

As the intention of this feature is to allow more manageable implementation of remedial work not to defer it work indefinitely, this capability is limited in the number of previously supported releases. In DB2 11 this feature only provides backwards-compatibility for DB2 10. Beyond DB2 11, compatibility for up two previous releases will be provided. This means that the release following DB2 11 will support both DB2 10 and DB2 11 compatibility, thereby allowing plenty of time for any remedial work to be undertaken.

By breaking the hard dependency on performing all remedial work prior to an upgrade and providing valuable tools to assist with the identification of that work, the DB2 Application Compatibility feature addresses many of the issues associated with handling incompatible changes in each new DB2 release. This capability should be a huge benefit to customers struggling to line up the necessary application development / DBA resource to address incompatible changes prior to DB2 11 being implemented.

Transparent Archiving

A very common requirement for any IT application is to be able to archive old or less frequently accessed data. Regulatory restrictions may require data to be retained for many years, but access frequency tends to drop off dramatically as the data ages (see Figure 9 below). Moving older data to a separate archive can reduce the cost of retrieving and maintaining more frequently used data, and allow slower but much cheaper storage devices to be utilised.

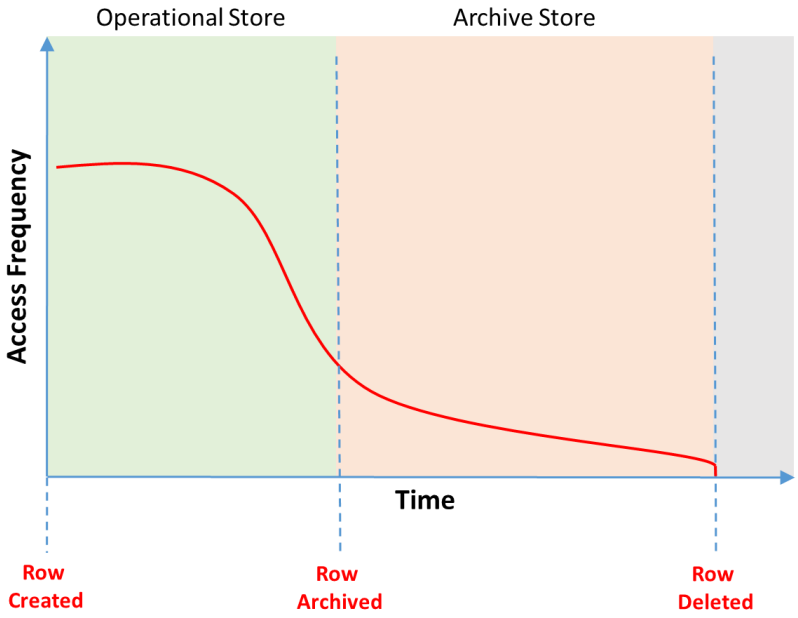

Figure 9 – Typical Data Access Life-Cycle

Unfortunately, archiving is usually one of the last areas to be considered and developed for a new application, and it is therefore common to see it deferred until later code releases (or bypassed completely) if time and/or funding is scarce. Even when it is properly implemented, many hundreds of man-hours can be spent in implementing and testing the necessary logic to allow older data to be placed automatically in the archive store while ensuring the application retains access to it when required.

DB2 11 introduces new features to simplify application development for archiving data, as well as improve consistency between applications and reduce the amount of time required for testing.

DB2 11 introduces some new features to simplify application development for archiving data, as well as improve consistency between applications and reduce the amount of time required for testing. When defining the operational DB2 table, the DBA also defines an identical archive table and connects the two via an ALTER TABLE…ENABLE ARCHIVE statement. Any subsequent changes to the operational table (e.g. adding a column) will be automatically made to the archive table so they remain in step. If required, the archive table can be placed on older, cheaper disk devices, with the more frequently accessed operational data residing on faster storage.

Once an archiving relationship has been defined, DB2 can automatically and transparently handle archiving and retrieval of data from the operation and archive tables. The new DB2 11 global variable support (see Global Variables on page 17) is used to provide simple application-level switches to enable or disable arching functionality at run-time, as shown in Figure 10 below. For static SQL, DB2 automatically prepares two access path strategies, one for use when archiving is enabled and another for use when it is not.

This approach is very flexible, providing automatic archiving and transparent access to archived data while also retaining the ability to disable that functionality via a simple SQL statement (or BIND option) when performance is critical and/or archiving functionality is not required.

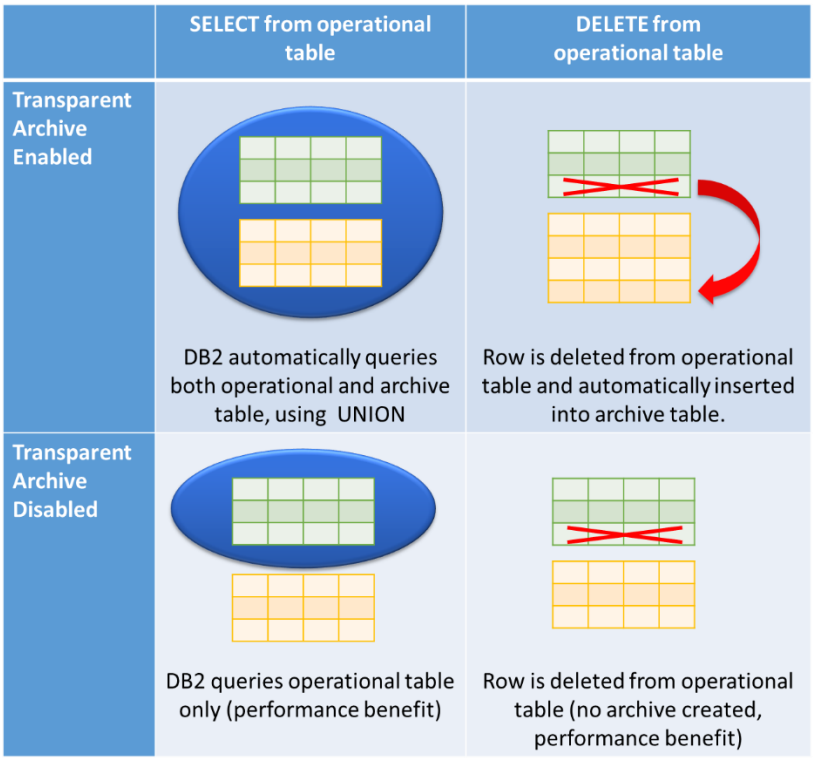

Figure 10 – Transparent Archive Behaviours

The new transparent archive feature promises to significantly reduce the cost of designing, developing and testing data archive processes for DB2 applications. It could reduce developer / DBA effort by hundreds of man-hours.

Please note that Version 3 of the IBM DB2 Analytics Accelerator product also offers some interesting options for handling archive data. Please refer to IBM DB2 Analytics Accelerator Enhancements on page 41 for further details.

Overall, the new transparent archive feature promises to significantly reduce the cost of designing, developing and testing data archive processes for DB2 applications. While it will be of limited value for those applications that have already implemented such functionality manually, it could reduce developer / DBA effort by hundreds of man-hours for newly developed applications (or existing applications that did not originally implement an archiving strategy). Solutions developed using this feature will also benefit from the ability to dynamically enable and disable access to archive tables at run-time, ensuring that no performance overhead exists for processes that only require access to the current operational data.

Temporal data enhancements

Temporal data support was first introduced in DB2 10, and provided a unique set of facilities to allow data to be queried as at a specific point in the past, present or future (please see Appendix A – DB2 10 Review on page 57 for further details). With so many IT systems needing to accommodate a historical perspective and maintain audit logs of changes made to sensitive data, DB2's temporal support can save many hundreds of hours of design, coding and testing that would otherwise be required to build this function manually for each application. However, the initial implementation had some restrictions and based on user feedback IBM has enhanced DB2's temporal capabilities within the new release in a number of important areas.

- **Temporal Special Registers**. Although the initial implementation of temporal query in DB2 10 allowed

existing tables to be easily converted via ALTER TABLE, it was still necessary to alter the SQL in applications to include the necessary temporal clauses.

DB2 11 introduces two new special registers (CURRENT TEMPORAL SYSTEM_TIME and CURRENT TEMPORAL BUSINESS_TIME) which implicitly provide temporal context to SQL queries without having to change them. When the special registers are left to the default NULL value, SQL executes without temporal context as usual. However, when they are set to a past or future timestamp value, that value is used to implicitly supply a temporal context to all SQL subsequently executed within the session. Provided that the underlying tables are temporal, this enhancement makes it possible to quickly and easily make DB2 applications temporally-aware without the need to make any code changes.

Temporal Special Registers make it possible to quickly and easily make DB2 applications temporally-aware without the need to make any code changes.

- **System Temporal Performance Optimisation**. Access to DB2 10 system temporal tables required DB2 to UNION together the base and history tables for all queries. As the majority of the access to such tables tends to be querying current data, the additional access to the history table often posed an unnecessary overhead.

  The new temporal special registers (described above) provide an opportunity to avoid this situation. Provided that a new option (SYSTIMESENSITIVE YES) is specified when BINDING the application, DB2 will prepare two separate access path strategies for queries against system temporal tables. As shown in Figure 11 below, DB2 will then use the relevant access strategy depending on whether the CURRENT TEMPORAL SYSTEM_TIME register is set or not.

The system temporal performance enhancement provides the best of both worlds, with applications able to access historical data when required but avoid unnecessary performance overheads when it is not.

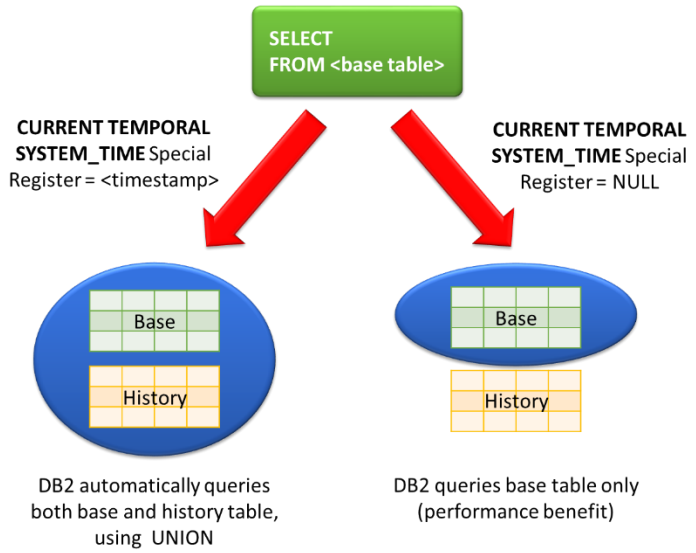

Figure 11 – System Temporal Performance Enhancement

This approach provides the best of both worlds, with applications able to access historical data when required but avoid unnecessary performance overheads when it is not.

- **Temporal View Support**. DB2 10 allowed temporal queries to be executed against DB2 tables, but did not allow those queries to be executed against any views that referred to those tables. As many customers use views heavily throughout their DB2 applications, this posed a significant restriction.

  DB2 11 removes this limitation, allowing non-temporal and temporal tables to be freely mixed within a view definition. Temporal queries can then be executed against the view, with the relevant AS OF predicates being applied to any temporal tables. Similarly, temporal UPDATE / DELETE logic is applied for any temporal tables updated via a view.

The DB2 11 enhancements significantly expand the practical use cases for DB2 temporal tables, and will allow many more customers to take advantage of the substantial productivity and consistency benefits they have to offer.

Together, these enhancements significantly expand the practical use cases for DB2 temporal tables, and will allow many more customers to take advantage of the substantial productivity and consistency benefits they have to offer.

Global Variables

One of IBM's stated objectives is to make it easier and more cost effective to port applications from other databases to DB2. One of the major remaining barriers to this activity has been the fact that, unlike most other RDBMS, DB2 for z/OS did not offer support for global variables. These constructs make it possible for SQL statements within the same application context to share data without the need to use application logic or insert the data into temporary tables. Significant additional effort was required to re-write applications designed for other RDBMS that used global variables so that they would work against DB2.

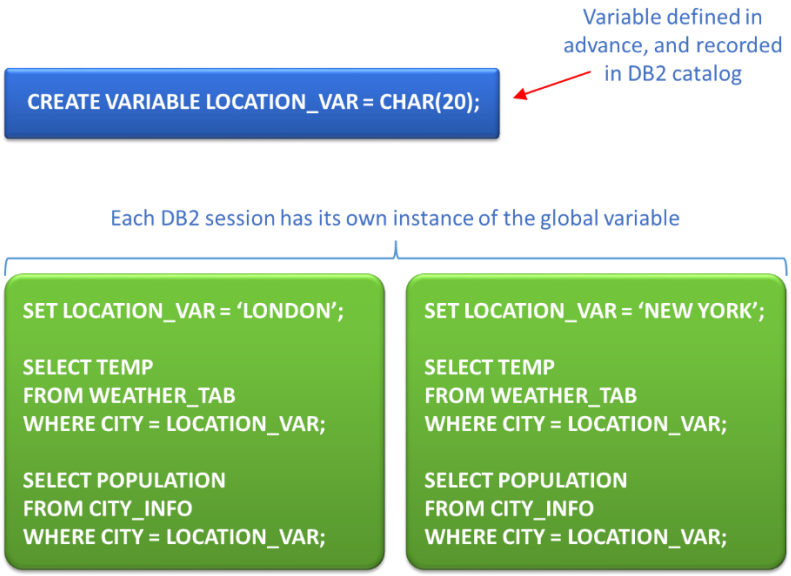

Figure 12 – DB2 Global Variables

The availability of global variables makes DB2 much more compatible with other RDBMS implementations, allowing applications to be more easily and cost-effectively ported to DB2.

DB2 11 introduces support for global variables, as shown in Figure 12 above. Global variables are defined to DB2 via the new CREATE VARIABLE SQL statement, with metadata details being recorded in the DB2 catalog tables. As shown in the example, each application referring to a given global variable then has its own instance of it, which can be set and read independently of any other connection.

The availability of this feature makes DB2 much more compatible with other RDBMS implementations, allowing applications to be more easily and cost-effectively ported to DB2. It also opens up some valuable new possibilities for developers and DBAs to more extensively parameterise SQL scripts, with associated productivity benefits.

Variable arrays

Many applications need to handle repeating groups of data with a variable number of elements. For example, a stored procedure to perform credit scoring may need to accept multiple customer account numbers as input to allow it to check several accounts at once. Variable arrays are a common means of addressing this kind of requirement, allowing an arbitrary number of elements sharing the same data type to be easily and elegantly addressed within an application.

Unfortunately, despite the growing popularity of SQL stored procedures for handling critical business logic, previous releases of DB2 did not provide support for array handling within the SQL PL language they are written in. Therefore such requirements had to be satisfied by various workarounds such as defining a long list of differently named stored procedure parameters, using temporary tables to store data, or concatenating all of the data into a long string and then splitting it up again. All of these solutions have serious drawbacks in terms of efficiency, developer productivity, application maintainability and code portability.

DB2 11 introduces formal support for variable arrays with SQL stored procedures. An array data type is formally declared to DB2 (via the new CREATE TYPE…ARRAY SQL statement), and can then be used instead of a standard data type when defining a PL SQL variable, or the input or output parameters of an SQL stored procedure.

Variable arrays remove a major gap in the capabilities of the SQL PL programming language. As a result, SQL stored procedures can be written with less developer effort, will be easier to maintain and will run more efficiently than their DB2 10 equivalents.

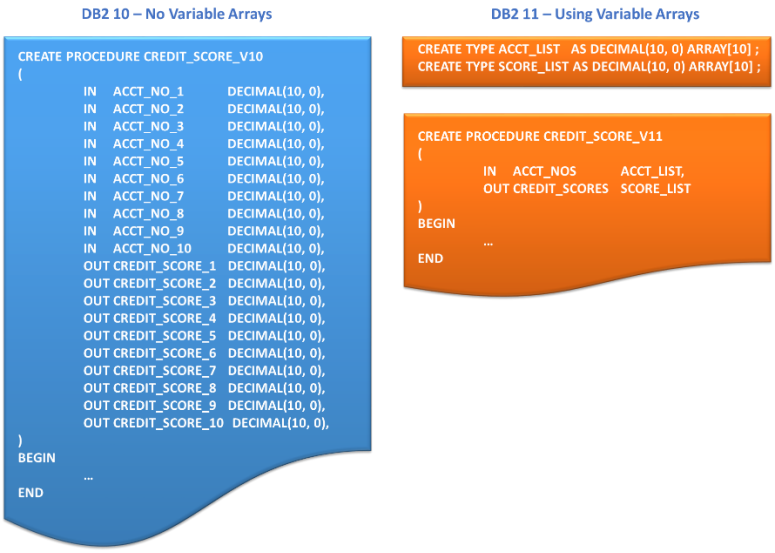

Figure 13 – DB2 Variable Arrays

Figure 13 above shows the previously quoted example of a credit scoring stored procedure that needs to accept multiple customer account numbers as input to allow it to check several accounts at once (up to 10 in this case). Prior to the availability of arrays, the

SQL PL developer would have been forced to define separate input/output parameters for each element in the repeating groups as shown on the left. The same result is achieved far more elegantly and straightforwardly using arrays in the DB2 11 example on the right of the diagram.

For customers making extensive use of SQL stored procedures, the availability of variable arrays removes a major gap in the capabilities of the SQL PL programming language. As a result, SQL stored procedures can be written with less developer effort, will be easier to maintain and will run more efficiently than their DB2 10 equivalents.

Java Stored Procedure Enhancements

Java long been considered a mainstream enterprise application development language, and many customers use it to write their DB2 stored procedures. As shown in the left of Figure 14 below, prior releases of DB2 used a 31-bit address space to execute these stored procedures, with each stored procedure running in its own Java Virtual Machine (JVM). This approach imposed a practical limit of 2-5 concurrent stored procedures within a single address space due to storage constraints, and created a significant overhead due to the need to keep starting multiple JVMs.

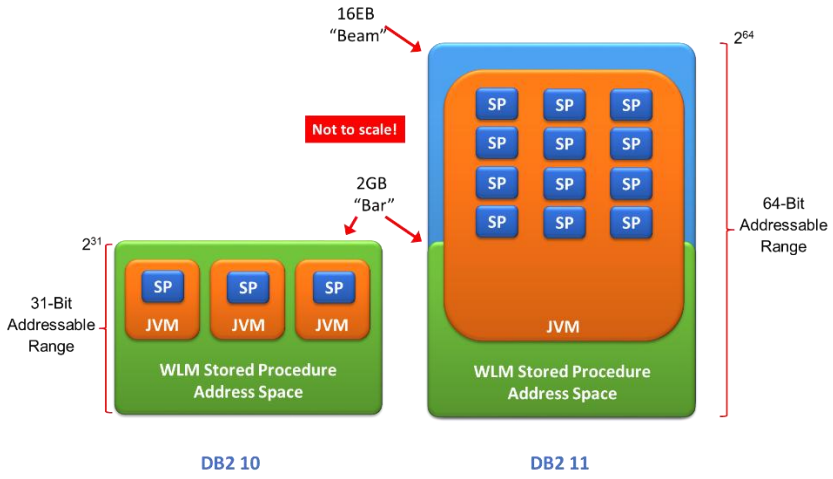

Figure 14 – DB2 Java Stored Procedure Enhancements

The DB2 11 enhancements greatly improve the scalability of Java stored procedures, reducing operating costs by lowering the CPU and storage overheads associated with many JVM instances.

DB2 11 addresses these limitations by supporting 64-bit multi-threaded WLM stored procedure address spaces[7]. As shown on the right of Figure 14 above, this allows a single JVM to support multiple stored procedures, with up to 25 per WLM address space observed in early IBM testing.

This enhancement greatly improves the scalability of Java stored procedures, reducing operating costs by lowering the CPU and storage overheads associated with many JVM instances.

pureXML enhancements

DB2 9 introduced pureXML: 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 improvements were provided in DB2 10, and DB2 11 introduces an even more

[7] 64-bit multi-threaded WLM stored procedure address space requires Java 64-bit JDK, version 1.6.

comprehensive package of performance and functionality enhancements[8]. These include:

- **XQuery Support**. XPath and XQuery are two of the most common languages used for accessing XML documents. XPath provides a simple and extremely efficient way of navigating the internal structure of an XML document, and extracting specific information from it. XQuery can be considered to be a superset of XPath: as shown in Figure 15 below, it includes all of the XPath syntax wrapped within a much more comprehensive general-purpose query language that is able to undertake more complex operations such as XML result set transformation and joins.

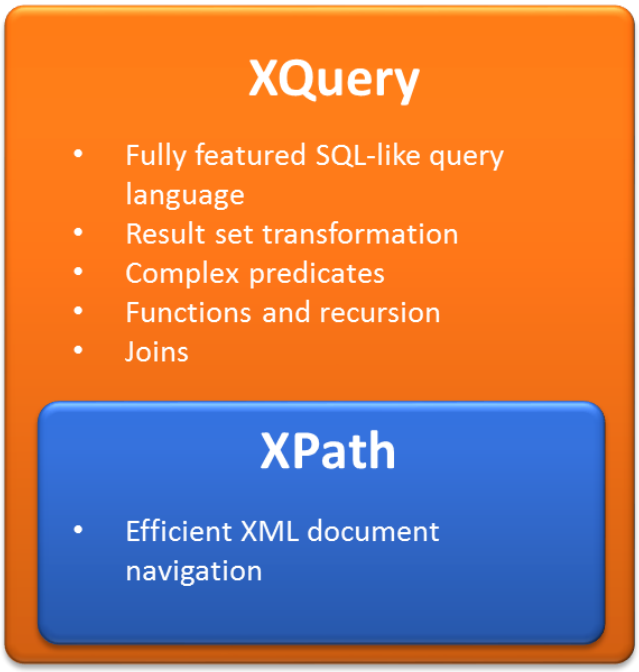

Figure 15 – XPath and Xquery

DB2 11 allows the full power of XQuery to be unleashed when querying XML data within DB2 for z/OS, improving developer productivity and DB2 family compatibility.

The pureXML feature within DB2 for Linux, UNIX and Windows has supported both XPath and XQuery for some time, but until now DB2 for z/OS only allowed developers to use XPath. DB2 11 redresses the balance, allowing the full power of XQuery to be unleashed when querying XML data within DB2 for z/OS. This will improve developer productivity and DB2 family compatibility.

- **pureXML Performance Enhancements**. The IBM development team has addressed a significant number of pureXML performance challenges within DB2 11. XML validation has been improved in several ways: the process can now be performed directly in binary format without having to convert to string XML first, LOAD is able to avoid re-validating XML if it has previously been validated, and DB2 is now able to revalidate only the changed parts of an XML document rather than the whole thing. Other performance enhancements include elimination of hotspots

[8] Please note that XQuery and some of the performance enhancements discussed within this section will also be retrofitted to DB2 10 for z/OS via the maintenance stream.

during XML INSERT and improvements to the XMLTABLE table function.

Figure 16 below shows some of the CPU savings seen during internal IBM testing conducted during the ESP programme.

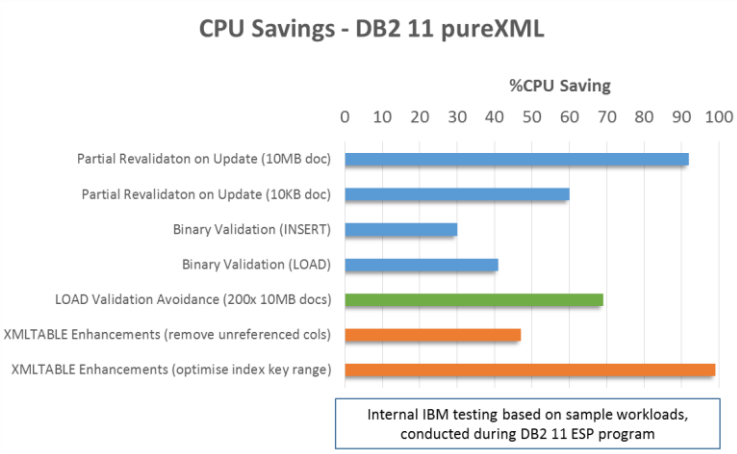

Figure 16 – Sample XML CPU Savings Using Internal IBM Workloads

pureXML has rapidly matured to the extent that many DB2 customers have adopted it as their strategic XML repository. The DB2 11 enhancements further consolidate this position.

pureXML has rapidly matured to the extent that many DB2 customers have adopted it as their strategic XML repository. The availability of the powerful XQuery language and some very welcome performance enhancements further consolidate this position.

Optimiser and Query Performance Improvements

No new release would be complete without further enhancements to DB2's industry-leading optimiser – the key component that allows it to pick the most efficient access path for a given query.

Unless otherwise specified below, all of these capabilities are available immediately on entry to Conversion Mode for dynamic SQL, and at REBIND time for static SQL, with no changes being required to the application or SQL code.

The major items delivered as part of DB2 11include:

- **Hash Join / Sparse Index Enhancements**. DB2 10 introduced hash join[9] support for access paths using an In Memory Data Cache (IMDC)[10]. DB2 11 further optimises memory usage by IMDCs in order to significantly improve query performance. At run-time DB2 will choose the most appropriate access strategy for each query based on the amount of memory currently available, as shown in Figure 17 below.

[9] Hash join is an alternative join method to the Merge Scan or Nested Loop techniques, and can be more efficient in certain instances.

[10] An In Memory Data Cache (IMDC) is an area of memory that holds intermediate work files when tables are joined. IMDCs were introduced in DB2 for z/OS Version 8, as a more cost-effective alternative to the sparse index technique that was used in prior versions of DB2 for z/OS (sparse indexes can still be used by DB2 where there is insufficient memory available to use an IMDC).

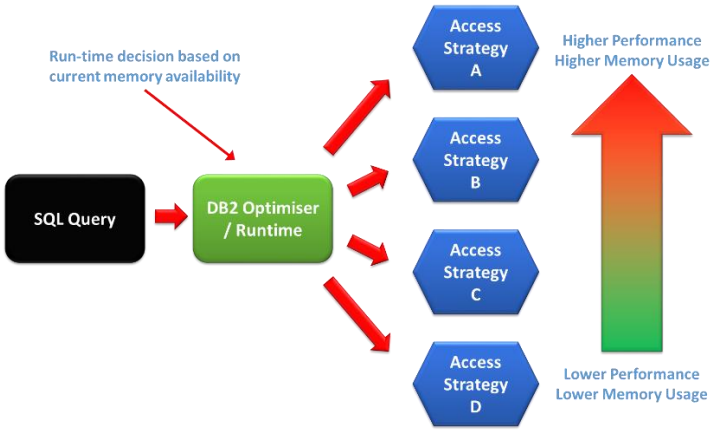

Figure 17 – DB2 11 In Memory Data Cache Optimisation

Using this approach, DB2 can dynamically alter its behaviour based on the specific workload executing at any given time, making best use of the memory available to provide optimum performance.

The hash join enhancements are expected to deliver measureable performance benefits "out-of-the-box", but significant additional benefits are expected for those customers running suitable query workloads that are able to increase the memory available for IMDC operations.

IBM has also provided new trace data within DB2 11 to allow DB2 administrators to easily determine where performance could be improved by making additional memory available for IMDC operations.

These enhancements are expected to deliver measureable performance benefits "out-of-the-box", but significant additional benefits are expected for those customers running suitable query workloads that are able to increase the memory available for IMDC operations[11] beyond DB2's default value of 20MB.

- **Predicate Indexability Improvements**. Using an index is generally the most efficient way to retrieve a small number of rows from a large table, but some predicates cannot be used to match indexes: these are known as non-indexable. It is often possible for a developer to rewrite a query so that the non-indexable predicates are replaced with indexable equivalents. This will allow the query to execute much more efficiently, but requires the developer to have a deep understanding of DB2 query performance.

  In some cases, DB2 is able to automatically rewrite queries to make predicates indexable, and DB2 11 adds a number of additional capabilities in this area. The following predicates have been specifically targeted:

  - `WHERE YEAR(DATE_COL) …`
  - `WHERE DATE(TIMESTAMP_COL) …`
  - `WHERE value BETWEEN C1 AND C2 …`
  - `WHERE SUBSTR(C1,1,n) …`

  DB2 11 is also able to rewrite queries using OR predicates that involve NULLs to use IN-lists instead. This makes them

[11] The memory available for IMDC operations is controlled by the MXDTCACH DB2 DSNZPARM parameter.

eligible for a more efficient single matching index access path.

Some examples of these new query rewrite capabilities are shown in Figure 18.

In some cases, DB2 is able to automatically rewrite queries to make predicates indexable, and DB2 11 adds a number of additional capabilities in this area.

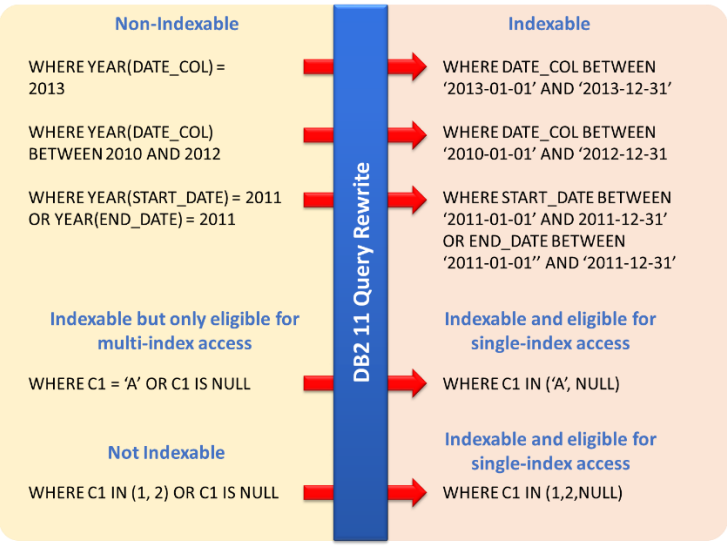

Figure 18 – DB2 11 Query Rewrite Examples

In addition, DB2 11 contains further enhancements to address SQL constructs typically seen in query generators and ERP applications. These include pruning "always true" predicates, extending the capabilities introduced in DB2 10 to prune "always false" predicates, allowing CASE predicates to be indexable and removing several other restrictions relating to conversion of correlated sub-queries and "pushing down" predicates into materialised views.

- **Duplicate Removal**. Many SQL operations require DB2 to remove duplicate entries in a result set, including SELECT DISTINCT, GROUP BY and non-correlated sub-queries. Eliminating duplicates can be done via sorting but this can be expensive in performance terms, so several other techniques can also be used. DB2 11 introduces three significant enhancements in this area.

Many SQL operations require DB2 to remove duplicate entries in a result set. DB2 11 introduces three significant enhancements in this area.

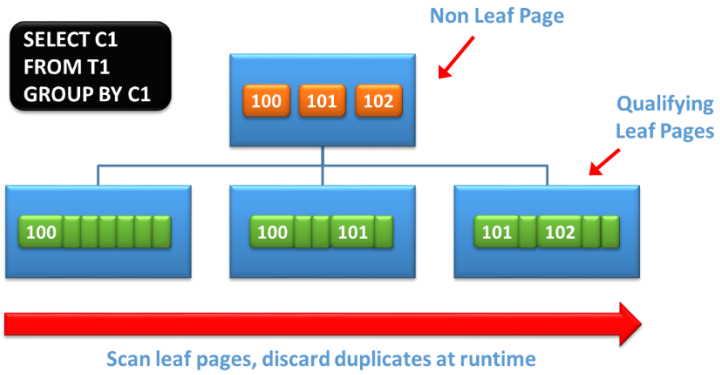

Figure 19 – Pre-DB2 11 Duplicate Removal Using Index Scan

Firstly, DB2 11 is able to more efficiently use an index to eliminate duplicates via a technique known as index

skipping. Rather than reading all of the qualifying bottom-level index leaf pages and throwing away duplicates at runtime (see Figure 19, above), DB2 11 is able to use a combination of leaf and non-leaf pages to skip directly to the next distinct value as shown in Figure 20.

In some cases, DB2 is able to automatically rewrite queries to make predicates indexable, and DB2 11 adds a number of additional capabilities in this area.

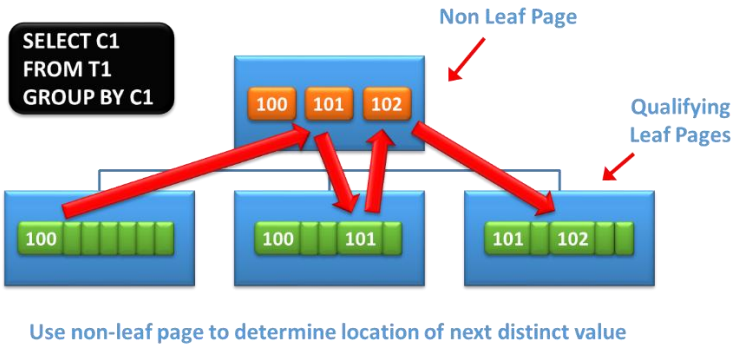

Figure 20 – DB2 11 Duplicate Removal Using Index Skipping

DB2 11 is also able to more efficiently process join queries that contain the GROUP BY or DISTINCT keywords, a common construct for ERP applications. This uses a process known as “early out” to abandon scanning the inner table after the first value has been returned, thereby avoiding the extra work needed to retrieve duplicate values and discard them later.

Finally, enhancements have been made to processing for correlated sub-queries. DB2 11 is able to use another new “early out” optimisation when the sub-query uses the MAX or MIN function, and make better use of a result cache first introduced in DB2 for z/OS Version 2 for saving the 100 most recent correlated sub-query execution results.

- **DPSI’s and Page Range Screening**. Data-Partitioned Secondary Indexes (DPSIs) were introduced in DB2 Version 8, as a means of providing many of the benefits of a secondary index while retaining good partition independence characteristics.

DB2 11 introduces some enhancements that significantly increase the “sweet spot” for DPSIs, allowing them to be more widely deployed.

While they can be very beneficial in some circumstances, DPSI’s can also introduce performance compromises for some queries if they do not include predicates on the partitioning key. DB2 11 introduces some enhancements that significantly increase the “sweet spot” for DPSIs, allowing them to be more widely deployed.

Firstly, page range screening for join predicates has been implemented. If a table is partitioned on one or more columns that are used in a join, and indexes exist on other local or join columns, then DB2 can apply page range screening from the join predicates and only access the qualified partitions for each table access. This could significantly improve the performance of joins involving DPSIs.

The second enhancement in the area involves improving the amount of parallelism available for queries involving

DPSIs. If a table has been partitioned by a non-join column and has a DPSI on the join columns, then DB2 can now choose to process each partition within a parallel child task. This can result in what was previously random I/O becoming sequential.

These enhancements are expected to provide a performance benefit for existing DPSI implementations, as well as significantly increasing the "sweet spot" for DPSIs and allowing them to be more widely deployed.

- **Optimiser RUNSTATS Feedback**. DB2's cost-based optimiser is highly dependent on both the quality and quantity of the table and index statistics available when it attempts to determine the optimum access path for a given query. While most DBAs have processes in place to ensure statistics are collected (via the DB2 RUNSTATS utility), most of these take the form of standard jobs using a common set of (or default) RUNSTATS options.

  If the DB2 10 optimiser encounters poor or incomplete statistics the DBA's first warning is typically when a poor access path is selected. He or she then has to determine the cause, and (if poor statistics are to blame) re-run RUNSTATS with the necessary additional options. This process is time-intensive, error-prone and relies on deep DBA knowledge of the correct RUNSTATS options for a given situation. This situation is depicted in Figure 21 below.

Optimiser RUNSTATS feedback can reduce the amount of DBA time spent addressing statistics-related access path issues, and help to drive down operational costs by allowing DB2 to select more efficient access paths.

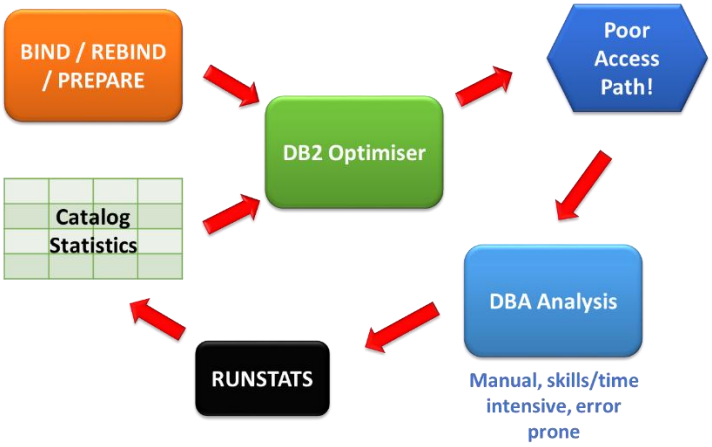

Figure 21 – DB2 10 Optimiser Feedback Issue

DB2 11 attempts to improve this situation by providing a means for the optimiser to signal when poor statistics are encountered – please refer to Figure 22 below. Two new tables[12] will be populated with the optimiser's recommendations on RUNSTATS options that would help it to select a better access path. Although that information unfortunately cannot be consumed directly by RUNSTATS, additional tooling can generate a suitable RUNSTATS job to address the issue. DB2 customers are free to write this themselves, or use a vendor tool such as IBM Optim Query

[12] SYSIBM.SYSSTATFEEDBACK for recommendations generated during optimisation of static/dynamic SQL during BIND, REBIND or PREPARE, and DSN_STAT_FEEDBACK for recommendations generated during SQL EXPLAIN.

Workload Tuner (which has been specifically enhanced to exploit this functionality).

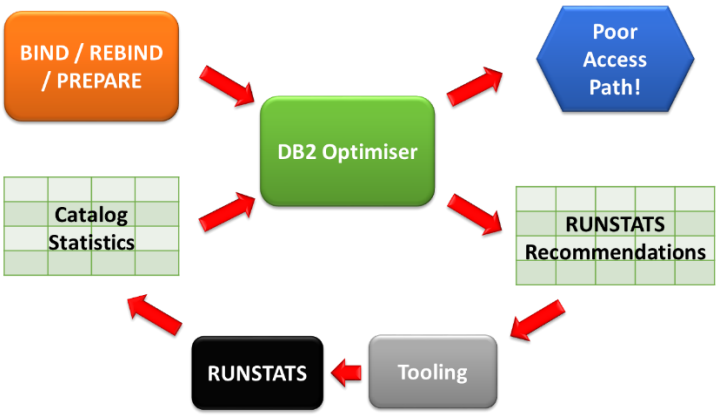

Figure 22 – DB2 11 Optimiser RUNSTATS Feedback

With the necessary tooling in place, this enhancement can reduce the amount of DBA time spent addressing statistics-related access path issues, and help to drive down operational costs by allowing DB2 to select more efficient access paths.

DB2 11's new extended optimisation capabilities give DBAs a better alternative to access path hints where the optimiser is unable to determine the best access strategy. This has the potential to improve DBA productivity as well as driving down operational costs for problematic queries through more efficient access path selection.

- **Extended Optimisation**. During access path selection, the optimiser needs to determine how many rows will be filtered by each step in the access plan in order to determine the overall cost of the query. Most of the time, it is able make good estimates, either based on specific statistics or using default "filter factors" based on the predicates in the SQL statement. However, there are some situations where the optimiser needs more help, especially with queries with pronounced data and/or execution skew[13] that use host variables or parameter markers. If this happens, the DBA would typically try to provide the optimiser with additional statistics (see section on Optimiser RUNSTATS feedback above) or, in more extreme circumstances, may have to resort to access path hints[14].

  DB2 11 provides an additional option by allowing the DBA to specify a "selectivity profile" for a specific SQL statement by inserting rows into a new table[15]. This profile contains information on both the predicate filter factors and the execution frequency for a problematic query. A process similar to that used for access path hints can then be followed to allow the optimiser to use this information and select a better access strategy. However unlike a hint this doesn't "lock in" a specific access path and still leaves the optimiser free to select the best available access path

[13] Data skew describes a common situation where data stored in a DB2 table is not uniformly distributed. For example, a table containing information about UK citizens may contain 60 million rows, and 200 different cities for primary address. A uniform distribution should result in each city having 60,000,000 / 200 = 300,000 citizens associated with it, but the city of London will have over 8 million. Execution skew describes a similar concept, where some literal values are used in queries much more than others. Again, queries WHERE CITY='LONDON' are likely to be far more common than those WHERE CITY='WELLS'.

[14] An access path hint allows the DBA to strongly suggest a specific access path to DB2, overriding the optimiser's usual cost-based analysis. This is generally considered to be a last resort, and is usually discouraged due to the additional effort required to create and subsequently maintain the hint.

[15] DSN_PREDICATE_SELECTIVITY

based on the improved data and execution information supplied in the selectivity profile.

As with the RUNSTATS recommendations previously described, the DBA is free to either manually insert rows into the selectivity table or use external tooling to assist (IBM's Optim Query Workload Tuner will provide this functionality).

DB2 11's new extended optimisation capabilities give DBAs a better alternative to access path hints where the optimiser is unable to determine the best access strategy. This has the potential to improve DBA productivity as well as drive down operational costs for problematic queries through more efficient access path selection.

In some cases, DB2 is able to automatically rewrite queries to make predicates indexable, and DB2 11 adds a number of additional capabilities in this area. The following predicates have been specifically targeted:

Data Sharing Performance Enhancements

Data sharing was introduced way back in DB2 V4 and over the next few releases rapidly established itself as an unbeatable solution for customers requiring the highest levels of resilience and scalability. However, relatively few performance enhancements have been delivered since.

Fortunately, DB2 data sharing performance has been selected as a focus area for DB2 11 and several significant enhancements have been included in the new release. These include:

Data sharing performance has been selected as a focus area for DB2 11 and several significant enhancements have been included in the new release.

- **Group Bufferpool and Castout Enhancements**. In a data sharing environment, the group bufferpools keep track of which data is being updated by which member, and will also usually[16] be used to cache copies of changed data. If an application updates a page which another member might want to access, DB2 copies the changed page from the local to the group buffer pool so other members can see the new version – please refer to Figure 23 below.

  There is no direct connection between the coupling facility and disk, so when DB2 needs to externalise the changed data it is written to disk via a dedicated pool in the owning member – a process known as "castout".

[16] It is possible to configure group buffer pools to not cache changed data, but this unusual.

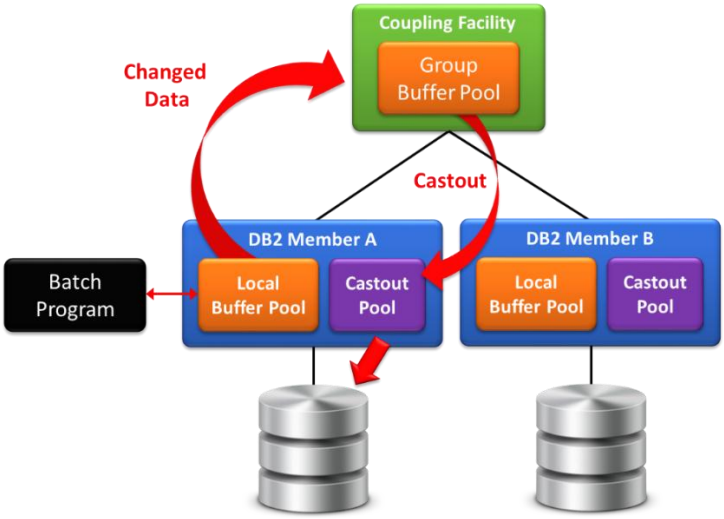

Figure 23 – DB2 Data Sharing Topology

In situations where there is heavy and sustained write activity to the group bufferpool (e.g. when running large batch or utility processes) a lot of changed data has to be written to the group bufferpool and in extreme circumstances this can lead to storage shortages and ultimately compromise data availability.

To help avoid this situation DB2 11 introduces a new capability[17] known as group bufferpool write-around. This allows DB2 to bypass the coupling facility and write changed pages directly to disk when the group bufferpool is under stress, significantly speeding up the process of offloading changed pages. Once the amount of changed data in the group bufferpool has returned to a reasonable level, normal processing is resumed again.

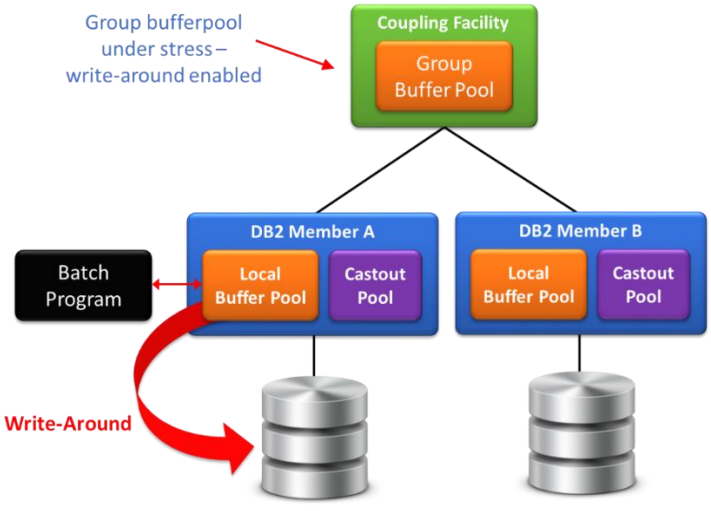

Figure 24 – DB2 Group Bufferpool Write-Around

In addition, castout processing itself has been improved to reduce the I/O elapsed time and optimise the

[17] Please note that this capability will also require a new coupling facility microcode level (CFLEVEL) to be installed. IBM intends to retrofit this function to DB2 10 for z/OS via the maintenance stream.

communications between the coupling facility and the castout owners.

- **Other Data Sharing Performance Enhancements**. Several other data sharing performance issues have been addressed within DB2 11. These include enhancements to coupling facility DELETE_NAME requests to supress unnecessary communications, a new option on light restart to include castout processing so that availability is improved, and various improvements to global locking and index split processing.

These enhancements promise to significantly improve data sharing performance while further increasing DB2's lead as the most scalable and available RDBMS in the industry.

Together, these enhancements promise to significantly improve data sharing performance while further increasing DB2's lead as the most scalable and available RDBMS in the industry.

Utility enhancements

DB2 utilities perform housekeeping and recovery functions that are essential to keep applications available and performing efficiently. IBM's objective is to continue to drive down the CPU and elapsed time for utility operations, while also reducing their availability impact and making them simpler to use. DB2 11 includes some valuable enhancements in these areas, as shown below.

- **REORG**. A large number of enhancements have been made to the REORG utility. Improvements to online REORGs[18] have reduced the elapsed time of the SWITCH phase by up to 91% in IBM testing and provided more control over the timing, thereby improving availability. DB2 11 also introduces automated mapping tables[19], improving DBA productivity and reducing the scope for human error.

Improvements to online REORG have reduced the elapsed time of the SWITCH phase by up to 91% and improved availability. Other enhancements will reduce CPU, improve DBA productivity and reduce the scope for human error.

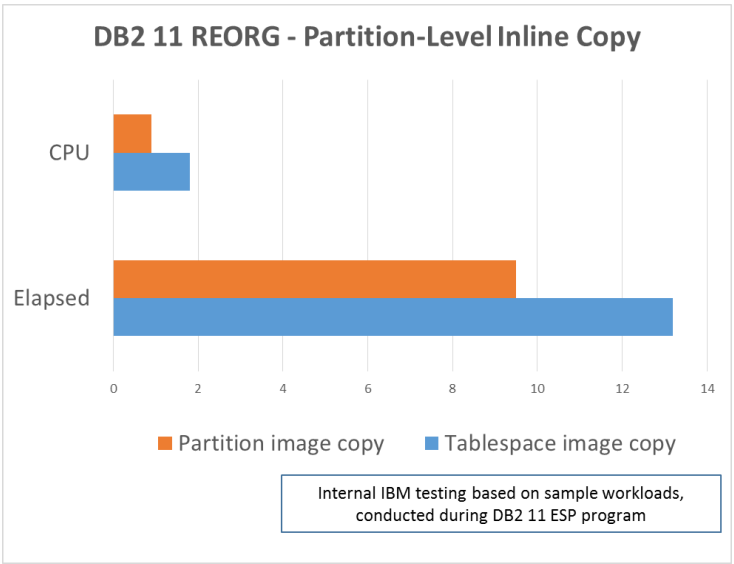

Figure 25 – REORG Partition-Level Inline Copy

Further new options allow the DBA to save CPU by choosing to reload data in the original order rather than

[18] Online REORGs allow the majority of the work to be done on a shadow copy of the data while full application access is maintained. The data is unavailable only for a short period at the end when access is switched to the shadow copy (known at the SWITCH phase).

[19] Online REORGs require a separate mapping table to be defined, which allows DB2 to keep track of the relationship between the rows in the original table and those in the shadow copy. Prior to DB2 11, these must be manually defined by the DBA.

sorting it, and to reduce recovery CPU and elapsed time using partition-level inline image copies. Internal IBM testing showed a 28% reduction in elapsed time and a 49% reduction in CPU time when recovering a single partition of a 20-partition tablespace (see Figure 25 above).

Other REORG enhancements include improvements to Large Object (LOB) processing, better support for rebalancing partitions following a partitioning key change (please refer to Enhanced Dynamic Schema Change on page 34) and updated defaults to reflect best practices.

The RUNSTATS utility has been enhanced to allow up to 80% of the CPU consumed when collecting distribution statistics to be offloaded to zIIP.

- **Statistics**. Collecting statistics is vital for good performance, but can be expensive in CPU terms. The RUNSTATS utility has been enhanced to allow up to 80% of the CPU consumed when collecting distribution statistics to be offloaded to zIIP. Improvements have also been made to inline statistics[20], increasing the zIIP offload potential and allowing histogram and distributed statistics to be collected inline.

  The section on Optimiser and Query Performance Improvements on page 21 also describes an important new feature allowing the optimiser to recommend specific RUNSTATS options to improve query efficiency.

- **Backup and Recovery**. Improvements have been made to the elapsed time for DB2 catalog/directory recovery, reducing application outage in the event of a serious DB2 problem. Other enhancements include the removal of restrictions for point-in-time recovery following dynamic schema change (described in the section on Enhanced Dynamic Schema Change on page 34), and new options to allow more efficient system cloning when using RESTORE SYSTEM.

The DB2 11 utility enhancements can significantly reduce operational costs by driving down CPU consumption while also improving application availability and DBA productivity.

- **Other Utility Enhancements**. Other utility-related enhancements include cross-loader support for XML data, increase zIIP offload for LOAD REPLACE processing and significant elapsed time reductions due to parallel data conversion during load,

Collectively, these enhancements can significantly reduce operational costs by driving down CPU consumption while also improving application availability and DBA productivity.

Other Efficiency Enhancements

A number of other important performance and productivity enhancements are delivered in DB2 11, including:

DB2 11 allows logging to be disabled for DGTT processes, reducing CPU usage and log volumes.

- **Global Temporary Table enhancements**. Declared Global Temporary Tables (DGTTs) are commonly used as an efficient way of for holding intermediate results within complex processes such as stored procedures. DB2 11 allows logging to be disabled for DGTT processes, reducing CPU usage and log volumes (at the expense of some additional ROLLBACK considerations within the

[20] Inline statistics are taken while another utility (such as REORG) is running. As that utility typically has to access all of the data anyway, the overhead for simultaneously collecting statistics is much smaller than running RUNSTATS separately.

application). The CPU overheads for repeated COMMIT processing against DGTTs have also been significantly reduced by keeping prepared versions of the statement across COMMIT boundaries.

- **Pseudo-Deleted Index Entry Clean-Up**. Under most circumstances, deleting rows from a table does not cause the associated index entries to be physically deleted, but merely marked as deleted. These are known as pseudo-deleted entries, and large numbers of them can cause performance issues until they are actually deleted via a subsequent REORG INDEX. DB2 11 is able to utilise up to 128 asynchronous system tasks to clean up these pseudo-deleted index entries, using information it already maintains on the number of entries awaiting deletion within each index. IBM has provided comprehensive control options allowing this activity to be limited by number of tasks, index, time of day, etc and all CPU for the asynchronous clean-up tasks is zIIP eligible.

DB2 11 is able to automatically clean up pseudo-deleted index entries, reducing performance degradation over time and allowing REORGs to run less frequently.

Figure 26 below shows a comparison of CPU degradation over time for a DB2 10 index and the equivalent situation in DB2 11 with the clean-up process enabled.

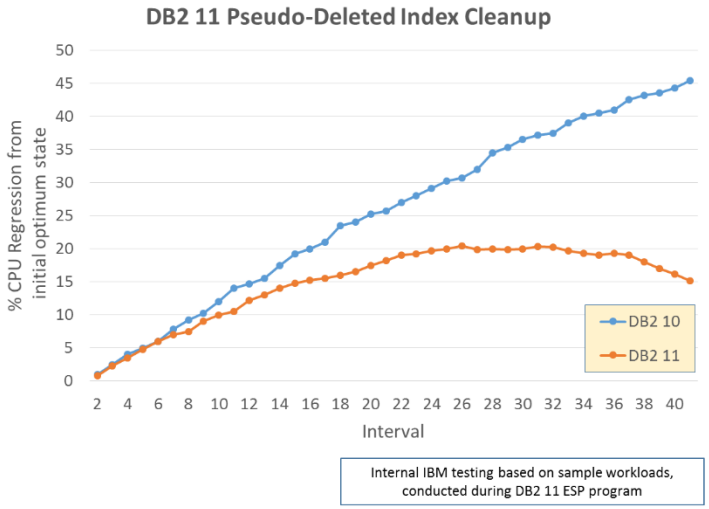

Figure 26 – CPU Benefit of Pseudo-Deleted Index Entry Clean-up

Several sort-related enhancements are included in DB2 11. As such a large proportion of SQL operations use sort these enhancements will benefit most DB2 workload.

- **Sort Improvement**. Several sort-related enhancements are included in DB2 11. These include increasing the amount of memory available for sorting, more aggressive use of in-memory sorting, and a reduction in latch contention. As such a large proportion of SQL operations use sort in one form or another, these enhancements will benefit most DB2 workloads.

- **SELECT from directory pagesets**. DB2 has always allowed users to issue SELECT statements against the metadata stored in the DB2 catalog, and this is a vital source of information for developers and DBAs. By contrast, the DB2 directory (which contains much of the same data but in an internal DB2 format optimised for performance) has remained firmly off limits. DB2 11 allows SQL SELECT

access to five directory pagesets[21] for the first time, allowing authorised users to view DB2's internal data for diagnostic purposes.

- **Enhanced Dynamic Schema Change**. In addition to reducing planned outages and the possibility of human error, the improvements described in Enhanced Dynamic Schema Change on page 34 can improve DBA productivity by significantly reducing the effort required to implement the relevant schema changes.

- **SQL Support for Analytic Workloads**. DB2 11 includes some SQL enhancements specifically aimed at supporting analytic workloads. These are described in SQL Aggregation Improvements on page 40 of this document.

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 the possible negative impacts of planned change, as well as increasing its ability to cope with ever-increasing workload volumes.

Extended Log Record Addressing

Current Issues

The DB2 recovery log records changes made to DB2 objects, and is critical for application and system restart/recovery. When DB2 was first released in 1983 it implemented a 6-byte log address[22], giving it the equivalent of 256TB of total logging capacity. This was considered to be an ample amount, allowing for many decades of logging at the volumes typical of the time.

Unfortunately, the relentless increase in transaction volumes combined with a significant increase in log record size due to support for Large Objects (LOBs) and other new data types, has resulted in an explosion in DB2 logging volume. This has been exacerbated by recent trends to consolidate workloads into fewer DB2 data sharing members due to DB2 10's scalability improvements (please refer to section on Growth on page 62 for more details), resulting in yet greater transaction volumes within each DB2 system.

Many customers are approaching the end of the DB2 log range in some of their older or more active systems.

The net result is that many customers are approaching the end of the DB2 log range in some of their older or more active systems, as shown in Figure 27 below. In some extreme cases with ultra-high logging volumes, this "log wrapping" is happening as often as every two months.

[21] SYSIBM.DBDR, SYSIBM.SCTR, SYSIBM.SPTR, SYSIBM.SYSLGRNX and SYSIBM.SYSUTIL.

[22] Known as a log Relative Byte Address, or log RBA

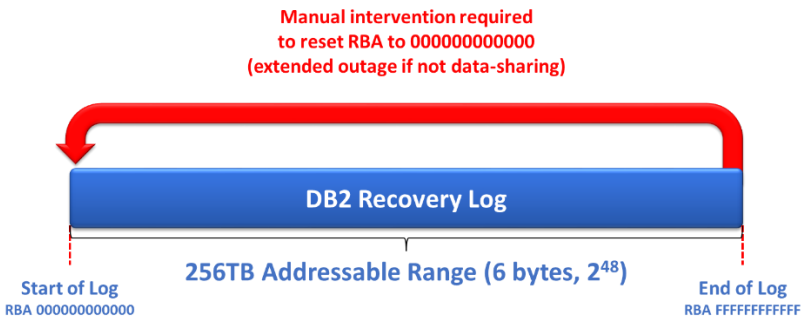

Figure 27 – 6-Byte DB2 Relative Byte Address

In previous releases, resetting the log relative byte address was a very painful process for non-data sharing customers, requiring extensive work and an extended outage of a day or more. Data sharing customers have a less disruptive option available, by "retiring" the DB2 system approaching the end of the log and adding a new one to replace it. However even this introduces additional cost and operational complexity with changes required to operator automation, batch scheduling, etc.

A similar situation exists with the Log Record Sequence Number (LRSN) used by data sharing to provide a common log identifier across all members in a data sharing group. This was also 6 bytes in length, but is based on a clock value with a maximum that will be reached in 2042. While that may appear to be of no immediate concern, under certain circumstances[23] a "delta" or offset is added to the current clock value and in extreme circumstances the offset can bring the actual LRSN value much closer to the 2042 high value. In Figure 28 below, Customer A has no LRSN delta and therefore has around 30 years of log capacity. However, Customer B has a very high LRSN delta of 25 years, leaving less than 5 years before running out of log capacity. Unfortunately, prior to DB2 11 there is no way to circumvent this issue.

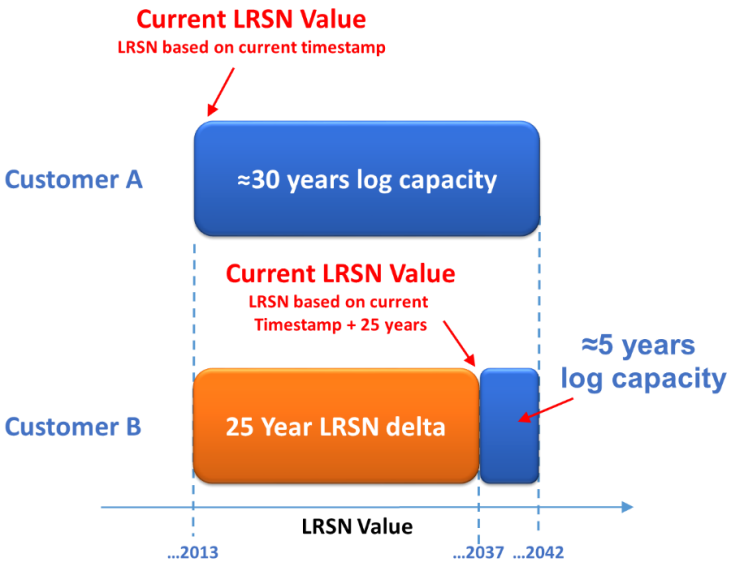

Figure 28 – 6-Byte Log Record Sequence Number

[23] An LRSN delta may be set when enabling data sharing for a non-data sharing system. Higher delta values are typical when enabling data sharing on older DB2 systems, or repeatedly enabling and disabling data sharing.

DB2 11 Enhancements

DB2 11 addresses all of the issues previously discussed by expanding both the log RBA and LRSN from 6 bytes to 10 bytes. For the log RBA, this provides 1 Yottabyte (2^{80}) of addressable range. To put that in context, the extreme DB2 customers currently wrapping the log every 2 days would be able to sustain 23.5 million years at the same logging rate before wrapping the expanded log.

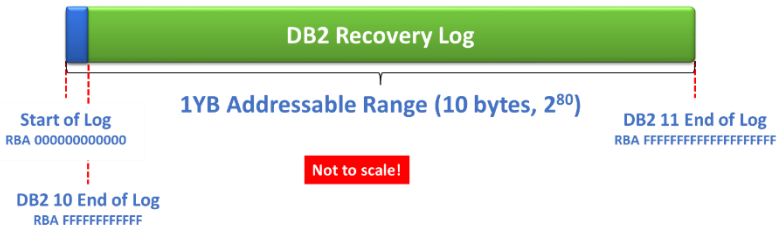

Figure 29 –10-Byte Log Record Sequence Number in DB2 11

For data sharing customers, one byte of the expanded LRSN value is used to extend the maximum clock value, with the remaining three bytes being used to increase its accuracy. This provides in excess of 30,000 years of additional log capacity, while also making the stored value 16 million times more precise (and thereby reducing performance issues associated with "LRSN spin").

Implementation of the expanded RBA/LRSN formats is optional, but highly recommended as DB2 11 will use 10-byte values internally once in New Function Mode, and there will be a small performance penalty for converting these to the old format. Data sharing customers will be able to maintain full application availability by converting one member at a time, while non-data sharing customers will need to schedule a small outage (typically less than a minute).

The move to 10-byte log addresses represents a major investment in DB2's future, and provides customers at or approaching the existing log limits with an elegant solution while also delivering some valuable data sharing performance benefits.

The move to 10-byte log addresses represents a major investment in DB2's future, and provides customers at or approaching the existing log limits with an elegant solution while also delivering some valuable data sharing performance benefits.

Enhanced Dynamic Schema Change

As applications evolve, the associated changes to DB2 objects such as tables and indexes remain one of the most common reasons for disruptive planned outages. Dynamic schema change allows DBAs to alter the definition of DB2 objects while maintaining access to the underlying data, with the change being fully materialised the next time the data is reorganised as part of routine housekeeping.

IBM began focusing on dynamic schema change back in DB2 Version 8, and has steadily expanded its capabilities in every release since. DB2 11 introduces the following additional enhancements:

- **Change Partitioning Limit Keys**. Partitioning a table can provide important scalability and manageability benefits, but usually requires the DBA to specify a "limit key" for each partition so that DB2 knows which partition a given row belongs in[24]. Previously, changing the limit keys

[24] Note that DB2 10 introduced an alternative approach known as "partition by growth" that does not require a partitioning key to be defined. However the traditional "partition by range" approach is more commonly used.

immediately made all of the affected data in the table unavailable until a REORG was executed to redistribute the data according to the new limit keys, making this a highly disruptive operation.

DB2 11 allows applications to continue to access a table after partitioning limit keys have been changed.

DB2 11 allows applications to continue to access the table after partitioning limit keys have been changed. The DBA still needs to run a REORG for the change to actually be implemented, but this can be scheduled at a convenient time and with data availability being maintained throughout.

- **Drop Column**. DB2 has long supported the ability to add new columns to an existing table, but dropping columns that are no longer needed meant dropping and recreating the entire table. Many DB2 applications therefore leave these unwanted columns in place, leading to wasted space, poorer performance and maintainability challenges.

  The new release allows columns to be immediately dropped from an existing DB2 table with no application outage.

  The new release extends the ALTER TABLE SQL statement to allow columns to be immediately dropped from an existing DB2 table with no application outage. As with partitioning limit key change described above, this is a "deferred change" so although the column will no longer be returned if an application accesses the table, a subsequent REORG will be required to physically remove the column from the table data.

- **Recovery support for deferred schema changes**. The deferred schema change capability first introduced in DB2 10 (and further enhanced in DB2 11 as discussed above) allows DBAs to make schema changes at any time but defer the materialisation of those changes until a REORG can be scheduled at a convenient time.

  However, the DB2 10 implementation included a significant restriction relating to point-in-time recoveries. As shown in Figure 30 below, once the REORG had been run to materialise the pending change (at T2 in the example) it was not possible to perform a recovery to a prior point in time. DB2 11 removes this restriction, allowing recovery to any valid prior point.

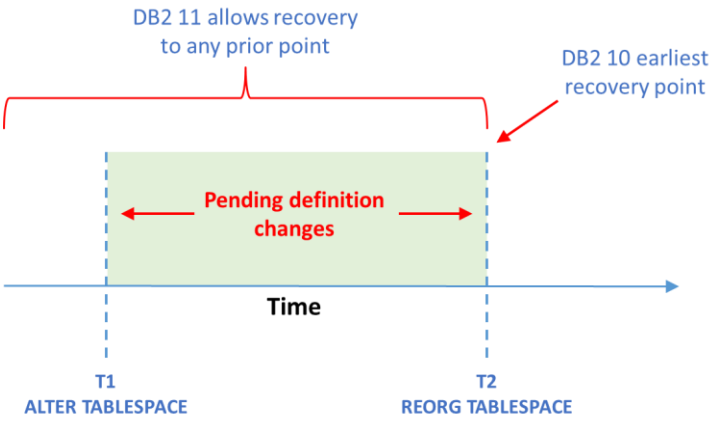

Figure 30 – Recovery Scenarios with Deferred Schema Change

The DB2 11 dynamic schema change enhancements significantly improve data availability, but also reduce the possibility of human error and improve DBA productivity.

Together, these dynamic schema change enhancements significantly improve data availability, but also reduce the possibility

of human error and improve DBA productivity as complex scripts to drop and recreate database objects can be replaced by a single SQL statement.

BIND / REBIND enhancements

The new plan management features[25] introduced in DB2 9 and DB2 10 went a long way to addressing customer concerns on potential access path regression during BIND / REBIND activity (please refer to Appendices

Appendix A – DB2 10 Review on page 57 for further details). In particular, a new BIND option allowed customers to ask DB2 to attempt to retain the existing access path by specifying APREUSE(ERROR). DB2 would then try to reuse the existing access path for all statements in a package, but if that wasn't possible for even one of the SQL statements in the package then **none** of them would be rebound. This situation is shown at the top half of Figure 31 below, and required the DBA to manually resolve the situation in order to successfully rebind the offending package.

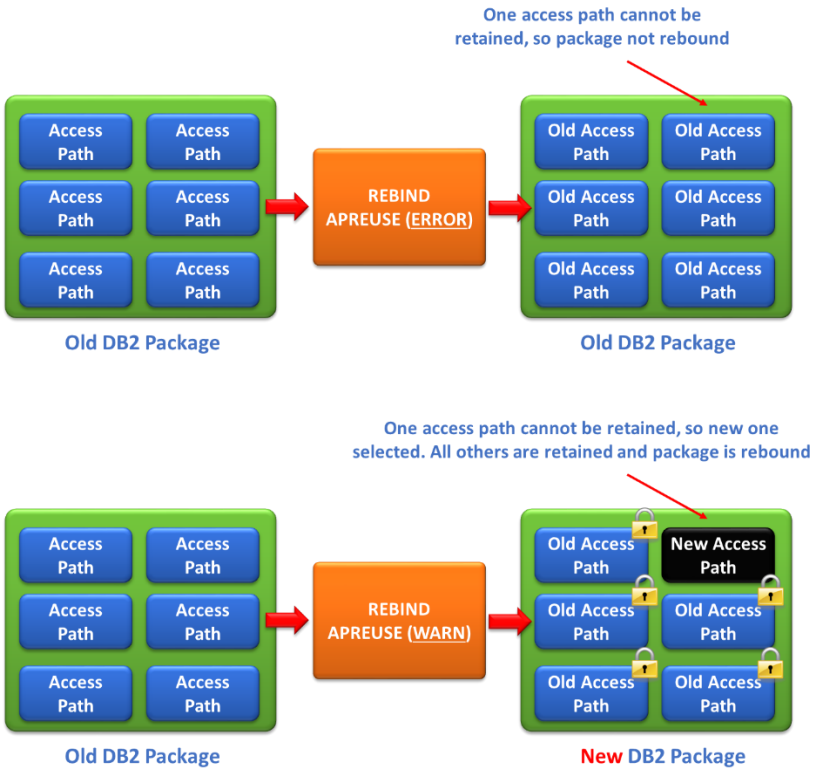

Figure 31 – Retaining Access Paths with APREUSE(WARN)

The new APREUSE(WARN) option greatly improves the usability of DB2 plan management, especially for large-scale rebind activity such as that performed when upgrading to a new version of DB2.

DB2 11 introduces APREUSE(WARN), a new BIND option that provides more flexibility for the DBA. As shown in the bottom half of Figure 31, any access paths that cannot be retained will go through normal optimisation to select a new one, while the access path for all of the remaining SQL statements will be retained.

This enhancement greatly improves the usability of APREUSE, especially for large-scale rebind activity such as that performed when upgrading to a new version of DB2. A greater proportion of packages can be successfully bound while retaining the majority of

[25] This feature has several commonly used names, including "plan management", "package management", and "plan stability".

the existing access paths, leaving the DBA to review any new access paths as time allows.

Security enhancements

DB2 received a comprehensive overhaul of its security features within the DB2 10 for z/OS release. DB2 11 adds some important new functionality, including:

- **RACF Exit Enhancements**. IBM offers two options for managing access to DB2 data: internal DB2 security or external security managed by a product such as RACF. Many customers choose external security, as this allows RACF administrators to handle access to DB2 objects using many of the same concepts, tools and procedures as for any other type of resource. However, the exit responsible for communication between RACF and DB2 had some significant limitations.

  Firstly, the RACF exit did not honour the use of the OWNER keyword[26] when plans/packages were bound or rebound, and would use the authorisation ID of the invoker of the BIND instead. Many sites rely on the use of OWNER within their DB2 security design, making use of external RACF security difficult or impossible. Similar restrictions also existed with the use of dynamic SQL if DYNAMICRULES(BIND) was specified.

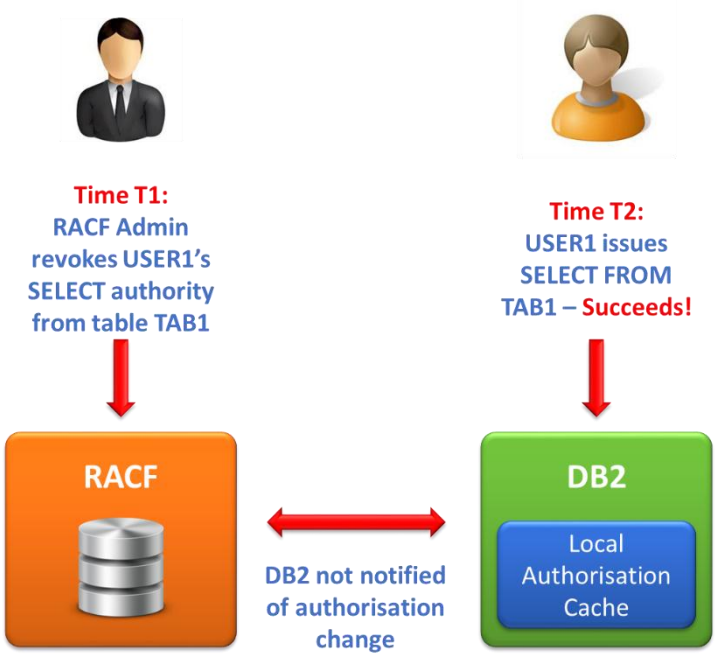

Figure 32 – Example of RACF/DB2 Authority Caching Issue

Secondly, some situations could occur where access to a DB2 resource had been revoked in RACF, but DB2 would continue to temporarily allow access. In the example shown in Figure 32 above, USER1 is still able to access TAB1 after the RACF administrator has removed her access, as DB2 uses internal DB2 authorisation cache which is not updated

[26] On BIND, the OWNER keyword specifies the owner of the plan or package being bound, which must have access to execute all of the static SQL inside the package being bound.

in line with the RACF database. Although techniques existed to force DB2 to refresh its internal authorisation cache, this required manual intervention and could be easily overlooked.

DB2 11 improves the consistency between external RACF security and internal DB2 authorisation management. It also includes new capabilities for ensuring that DB2's internal cache stays in step with changes made to RACF, thereby avoiding potential security exposures.

Both of these issues are resolved by the RACF exit changes delivered within DB2 11. A new installation parameter instructs DB2 to properly use the package owner for static and dynamic SQL authorisation checks, ensuring that external RACF security behaves in a similar way to internal DB2 authorisation management. Another new parameter enables a refresh function for DB2's internal authorisation cache. As shown in Figure 33 below, this ensures that DB2's internal cache stays in step with changes made to RACF, thereby avoiding the potential exposure.

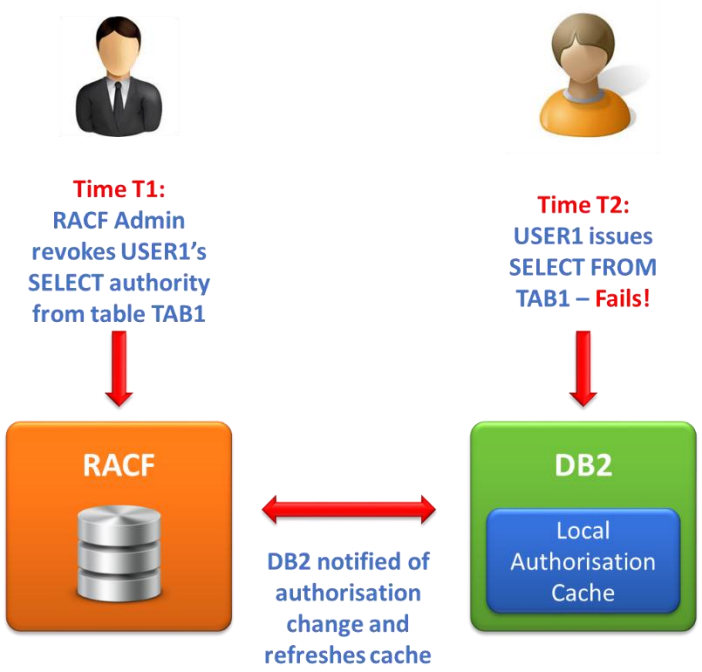

Figure 33 – DB2 11 RACF/DB2 Authority Cache Refresh

- **Column Masking**. The masking feature introduced in DB2 10 provided a powerful means of fully or partially concealing sensitive information by specifying a set of rules under which either the full value or a masked equivalent would be returned (typically this would be done according to the type of user making the request). DB2 11 removes some of the restrictions previously imposed on the use of the SQL GROUP BY and DISTINCT clauses when querying a masked table.

The DB2 11 security enhancements resolve some significant issues and will allow more customers to take advantage of important features such as external authorisation checking and column masking.

Although less extensive than the security extensions delivered in the previous release, the DB2 11 enhancements resolve some significant issues and will allow more customers to take advantage of important features such as external authorisation checking and column masking.

Other Resilience Enhancements

DB2 11 includes some enhancements for high-performance DBATs, allowing high-performance DBATs to be more widely deployed with associated benefits in performance and operational cost reduction.

- **Managing Persistent Threads**. The High-Performance DBAT feature introduced in DB2 10 delivered significant CPU savings for many customers by allowing distributed access threads (DBATs) to retain DB2 resources across a COMMIT point and thereby allow a similar type of thread reuse to that enjoyed by CICS users for many years. However this also introduced some challenges in performing routine BIND and DDL activities, which could struggle to obtain the necessary locks.

  DB2 11 includes some additional enhancements to allow it to selectively free accumulated resources for high-performance DBATs, making it easier for BIND and DDL activity to "break in" and complete successfully. This in turn allows high-performance DBATs to be more widely deployed, with associated benefits in performance and operational cost reduction.

"I expect this will allow us to use some BIND parameters that should give some significant CPU savings and especially for highly used packages driven through persistent threads which are used widely in our online systems."

Frank Petersen, JN Data DB2 Systems Programmer

- **Autonomics Improvements**. IBM has a stated objective of increasing DB2's ability to manage itself via new autonomic capabilities. DB2 11 includes new facilities to automatically clean up pseudo-deleted index entries (described in the section on Other Efficiency Enhancements on page 30), and some improved techniques for reserving free space for SQL UPDATE operations to complete on the optimal page. In addition to improving DBA productivity, these autonomic facilities improve the robustness and availability of DB2 applications.

The enhanced autonomic facilities promise to increase DBA productivity and improve the robustness and availability of DB2 applications.

- **Scalability Enhancements**. Several enhancements have been made to remove DB2 scalability limitations. These include doubling the maximum number of open datasets each DB2 system can access and increases to the maximum size of several internal objects to improve virtual storage usage.

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.

In a recent IBM survey[27], 72% of the respondents indicated that they would be using transactional data from enterprise applications as input to their big data analysis programs. Given that both Volume and Velocity are highly significant in managing big data/analytics workloads, it makes sense for organisations to consider performing the analysis on the platform that the source data resides upon.

DB2 9 and DB2 10 delivered some significant new functionality to support business analytics workloads, and DB2 11 further expands its capabilities in this area. However, some equally important

[27] 2012 IBM Global Big Data Online Survey. Base: 60 IT Professionals, multiple responses accepted.

developments are occurring within the supporting System z tools and infrastructure and we will also examine those within this section.

SQL Aggregation Improvements

DB2's basic aggregation capability satisfies simple requirements, but BI / analysis tools typically had to retrieve all of the detail / fact data from DB2 so they could perform more complex aggregation themselves.

Analytics workloads commonly require complex aggregation operations to be performed in order to summarise large amounts of data. DB2 has long supported simple aggregation via the SQL GROUP BY clause (as shown in the example in Figure 34 below). This capability satisfies many basic requirements, but more sophisticated BI / analysis tools typically had to retrieve all of the detail / fact data from DB2 so they could perform more complex aggregation themselves.

SALES_DATE	SALES_PERSON	SALE_VALUE	...
01/07/2013	John	123.50	
04/07/2013	Mary	43.75	
10/07/2013	John	4453.00	
10/07/2013	John	43.54	
11/07/2013	Mary	765.12	
12/07/2013	Mark	12.47	

```
SELECT WEEK(SALES_DATE) AS WEEK,
       SUM(SALE_VALUE) AS SALES
FROM SALES_DATA
GROUP BY WEEK(SALES_DATE)
```

WEEK	SALES
27	167.250
28	5274.13

WEEK(SALES_DATE)

Figure 34 – GROUP BY Example

DB2 11 extends the basic GROUP BY aggregation with three additional capabilities. Firstly, the GROUPING SETS clause effectively allows multiple GROUP by queries to be executed within a single SQL statement, as shown in Figure 35.

```
SELECT WEEK(SALES_DATE) AS WEEK,
       DAYOFWEEK(SALES_DATE) AS DAY,
       SALES_PERSON,
       SUM(SALE_VALUE) AS SALES
FROM SALES_DATA
GROUP BY GROUPING SETS (WEEK(SALES_DATE),
                        DAYOFWEEK(SALES_DATE),
                        SALES_PERSON)
```

WEEK	DAY	SALES_PERSON	SALES
NULL	NULL	John	4620.04
NULL	NULL	Mark	12.47
NULL	NULL	Mary	808.87
NULL	1	NULL	123.50
NULL	3	NULL	4496.54
NULL	4	NULL	808.87
NULL	5	NULL	12.47
27	NULL	NULL	167.250
28	NULL	NULL	5274.13

SALES_PERSON

DAYOFWEEK(SALES_DATE)

WEEK(SALES_DATE)

Figure 35 – GROUPING SETS example

The example shows three separate aggregations (for week number, day of week and sales person) being executed within a single SQL statement.

The second new capability introduced by DB2 11 allows nested groups with sub-totals to be produced, using the ROLLUP function as shown in Figure 36. The same three aggregation attributes as the previous example has been used, but this time they are nested within each other with a sub-total being provided at each level.

```
SELECT WEEK(SALES_DATE) AS WEEK,
       DAYOFWEEK(SALES_DATE) AS DAY,
       SALES_PERSON,
       SUM(SALE_VALUE) AS SALES
FROM SALES_DATA
GROUP BY ROLLUP (WEEK(SALES_DATE),
                 DAYOFWEEK(SALES_DATE),
                 SALES_PERSON)
```

SALES_PERSON

WEEK	DAY	SALES_PERSON	SALES
27	1	John	123.50
27	1	NULL	123.50
27	4	Mary	43.75
27	4	NULL	43.75
27	NULL	NULL	167.25
28	2	John	4496.54
28	2	NULL	4496.54
28	3	Mary	765.12
28	3	NULL	756.12
28	4	Mark	12.47
28	4	NULL	12.47
28	NULL	NULL	5274.13
NULL	NULL	NULL	5441.38

DAYOFWEEK (SALES_DATE) with sub-total

WEEK(SALES_DATE) with sub-total

Figure 36 – GROUP BY ROLLUP Example

The new GROUP BY functions significantly expand DB2's ability to natively aggregate large volumes of data, allowing BI / reporting tools to analyse DB2 for z/OS data with lower response times, and at less cost, than before.

Finally, a GROUP BY CUBE function provides a similar roll-up functionality to the GROUP BY ROLLUP feature described above, but with the addition of cross-tabulation rows to create the equivalent of a data cube for slice and dice analysis.

These new functions significantly expand DB2's ability to natively aggregate large volumes of data, allowing BI / reporting tools to analyse DB2 for z/OS data with lower response times, and at less cost, than before.

IBM DB2 Analytics Accelerator Enhancements

Although IBM continues to expand DB2's native capabilities to more efficiently handle high-volume analytic workloads, it is very challenging for any general-purpose RDBMS to deliver the kind of query performance and efficiency that dedicated operational analytics appliances such at Netezza can achieve without sacrificing OLTP performance. In recognition of this, IBM offers the DB2 Analytics Accelerator: a unique blend of System z and highly optimised Netezza technology for efficiently handling complex, performance-critical operational analytic workloads.

As shown in Figure 37 below, in an IBM DB2 Analytics Accelerator implementation no changes are required to applications: they continue to connect to DB2 in order to issue queries. The DB2

optimiser is responsible for deciding if a given query should be offloaded, and if so it is passed to the locally-connected accelerator host for execution. Upon completion of the IBM DB2 Analytics Accelerator query, the results are passed back to DB2 which then passes them back to the calling application as if they had been obtained locally. Depending on the query, it is not unusual for IBM DB2 Analytics Accelerator queries that require hours to execute natively on DB2 to be completed in seconds once offloaded to the accelerator.

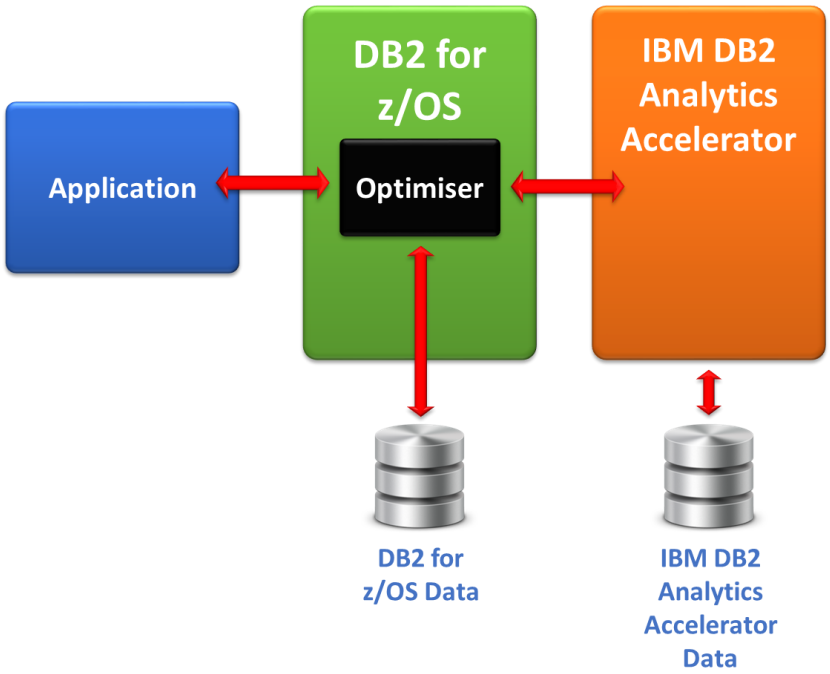

Figure 37 – IBM DB2 Analytics Accelerator Architecture

Version 3 the IBM DB2 Analytics Accelerator technology was announced by IBM in October 2012, and brings some significant enhancements that work in concert with DB2 10 and DB2 11 to expand the query workload capacity and capabilities.

Version 3 the IBM DB2 Analytics Accelerator technology was announced by IBM in October 2012, and brings some significant enhancements that work in concert with DB2 10 and DB2 11 to expand the query workload capacity and capabilities. These include:

- **Incremental Update**. As the IBM DB2 Analytics Accelerator stores its own local copy of the data to be queried, processes are required to keep the accelerator data in step with the operational copy held by DB2. Previous incarnations of IBM DB2 Analytics Accelerator only offered the ability to perform a full refresh of the data (at the table or partition level) which had obvious drawbacks for large tables with relatively low update frequency. IBM DB2 Analytics Accelerator V3 introduces incremental update capabilities which allow updates to be trickle-fed to the accelerator data store and provide for near-real time reporting.

- **Multi-Temperature Data**. The section on Transparent Archiving on page 13 of this document discusses one option for handling rarely-referenced DB2 historical data, but IBM DB2 Analytics Accelerator V3 provides another by allowing data to be moved from DB2 to reside in accelerator **only** as it ages. Also known as the High Performance Storage Server (HPSS), this provides a transparent means of moving rarely-referenced data to a cheaper storage platform while still providing access when required.

- **Increased Capacity**. IBM DB2 Analytics Accelerator V3 extends the maximum effective capacity of an IBM DB2 Analytics Accelerator instance to a massive 1.3PB.

Version 4 of the IBM DB2 Analytics Accelerator should be announced by the time this paper is published, and is expected to include a number of additional enhancements:

- **Accelerator Modelling**. DB2 has long provided the ability for customers to determine the amount of CPU that could be saved by directing eligible workload to zIIP[28] engines. This allows customers to rapidly assess the ROI of potential future zIIP purchases. IBM DB2 Analytics Accelerator V4 provides a similar capability for accelerated queries, with new instrumentation to show the CPU time used for queries that would be eligible for offload to an accelerator (whether one is actually available or not).

- **More Eligible Queries**. IBM DB2 Analytics Accelerator V4 significantly expands the number of queries eligible for offload. Static SQL queries are now eligible, and more DB2 data types and functions are supported.

- **Increased Accelerator Efficiency**. A number of enhancements have been made to improve the efficiency of the accelerator itself. These include workload balancing across multiple accelerators, concurrent load/replication support, enhanced monitoring and improved workload prioritisation.

- **Enhanced High Performance Storage Saver Solution**. The HPSS capability introduced in IBM DB2 Analytics Accelerator V3 has been further enhanced with improvements to archiving functionality and the ability to more easily restore archived tables.

The combination of DB2 11 and IBM DB2 Analytics Accelerator V3 significantly progresses IBM's objective to transition DB2 into a truly universal DBMS that provides the best characteristics for both OLTP and analytical workload.

The combination of DB2 11 and the many new features in IBM DB2 Analytics Accelerator V3 and V4 significantly progresses IBM's objective to transition DB2 into a truly universal DBMS that provides the best characteristics for both OLTP and analytical workloads.

Hadoop and Big Data Support

DB2 11 delivers some highly significant new features to allow DB2 and Hadoop / Big Insights to work together and better leverage each platform's respective strengths.

Few IT professionals can have missed the big data phenomenon that has manifested itself in recent years. Despite the undeniable value of the highly structured operational data held within enterprise applications, a vast amount of less-structured data is being generated by social media streams, telemetry, clickstreams, and many other sources. Being able to analyse these big data sources undoubtedly holds significant value for many organisations, but the sheer volume and velocity at which the data is produced makes it a very challenging task for traditional database systems. In response to this a number of tools and techniques have emerged centred on the open source Hadoop framework, including IBM's InfoSphere BigInsights technology.

[28] 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

While these technologies address many of the challenges inherent in the analysis of big data, they also introduce new ones for organisations wanting to gain new insights by integrating the analysis of big data with core operational information.

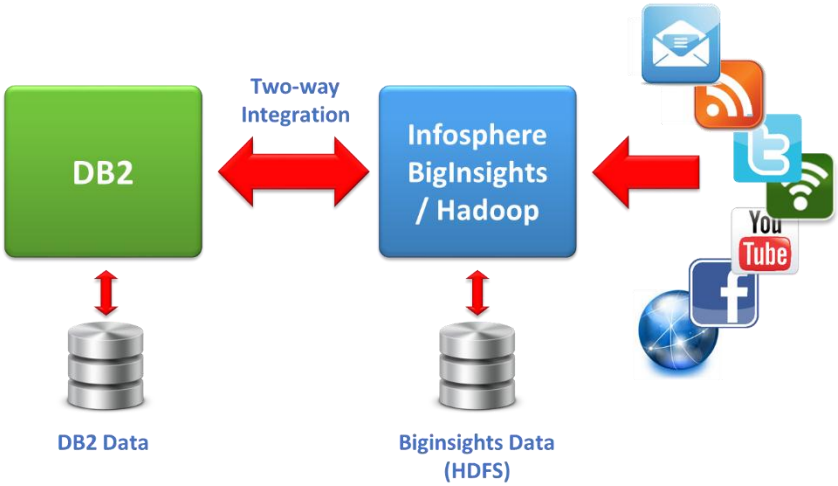

Figure 38 – DB2 11 / Big Data Integration

As shown in Figure 38 above, DB2 11 delivers some highly significant new features to allow DB2 and Hadoop/Big Insights to work together and better leverage each platform's respective strengths. This capability allows data to flow in both directions between DB2 and BigInsights, as follows:

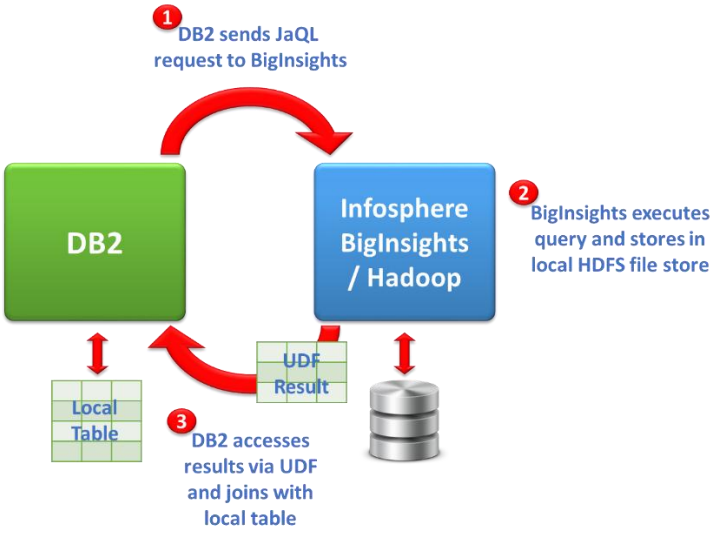

Figure 39 – DB2 11 Accessing BigIngights Data

- **From BigInsights to DB2**. DB2 11 provides new connectors and capabilities to allow it to easily and efficiently access BigInsights data. A query can be passed to BigInsights using a standard JSON-based query language called Jaql[29]. Once the query completes, BigInsights stores the results within its internal HDFS file system. A special new User Defined Function (UDF) within DB2 11 then allows this data to be read as if it were a DB2

[29] Jaql is a scripting language for enterprise data analysis based on the Hadoop MapReduce parallel programming framework. It was originally created by IBM, and donated to the open source community.

table, and even joined to other local DB2 tables if required. An example of this process is illustrated above in Figure 39.

- **From DB2 to BigInsights**. In the opposite direction, DB2 11 data can be accessed from within a Jaql query running within Big Insights. This capability uses a Jaql JDBC module within BigInsights to allow Jaql queries to access DB2 (or any other Java-enabled DBMS) as shown in the example in Figure 40 below. DB2 sees this request as just another incoming JDBC/DRDA remote connection, and returns the DB2 data to BigInsights where it will be processed in the same way as if it had accessed native HDFS data.

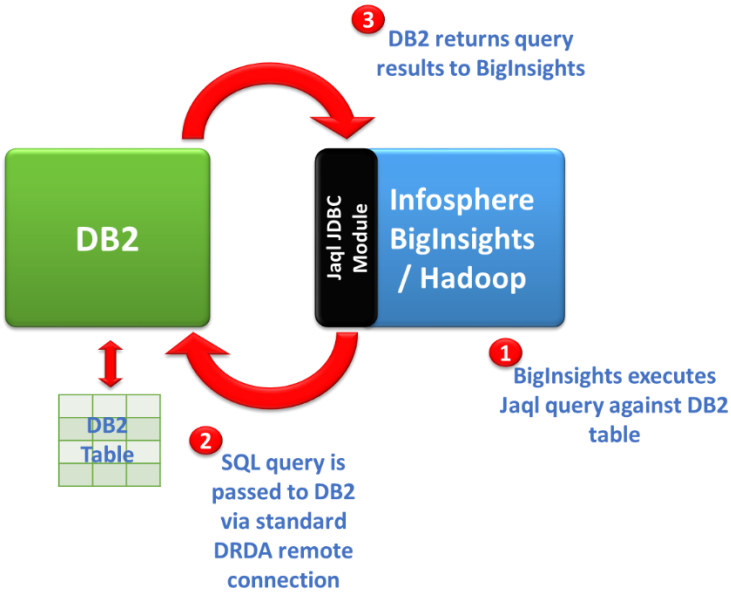

Figure 40 –BigIngights Accessing DB2 11 Data

This capability works well for low-volume queries that need access to current data, but some applications require a local copy of the DB2 data to be held within BigInsights/Hadoop for performance reasons. A tool known as Sqoop ("SQL-to-Hadoop") can be used for this purpose. Sqoop allows bulk data transfers between DB2 and BigInsights, and can load the extracted data into a local HDFS cache.

The new integration capabilities delivered within DB2 11 allow organisations to more easily and efficiently combine the results of big data analysis with up-to-date operational data from DB2 OLTP databases, significantly increasing the practical value of any insights obtained.

The new integration capabilities delivered within DB2 11 allow organisations to more easily and efficiently combine the results of big data analysis with up-to-date operational data from DB2 OLTP databases, significantly increasing the practical value of any insights obtained.

QMF 11

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

QMF 11 will be released at the same time as DB2 11 for z/OS. It provides many new or enhanced facilities, including:

- QMF Version 11 Enterprise Edition includes a new product, QMF Analytics for TSO, which provides statistical analysis,

forecasting functions, and additional chart types from an easy-to-use, menu-driven interface.

- Enhancements to QMF for TSO and CICS, including :
    - Big Data Integration, using the capabilities described in the section on Hadoop and Big Data Support above.
    - Support for analytical queries using the SQL enhancements described on page 40 and the IBM DB2 Analytics Accelerator enhancements on page 41.
    - Better support for large data types such as LOBs and XML
    - Mobile device support
    - Broader support for temporal data
    - Improved report storage management.
    - Customizable dashboards
    - Multiple database/platform support
    - Federated data support
    - Interoperability with other solutions such as SPSS
- Enhancements to QMF High Performance Option, including DB2 COMMIT scope improvements and the ability for a single SELECT statement to be included in a query with multiple SQL statements.
- Enhancements to DB2® QMF for Workstation and DB2 QMF for WebSphere, including ad-hoc reports, visual applications and support for text analytics.

Other Enhancements for Analytics Workloads

In addition to the specific analytic features discussed within this section, a number of other enhancements will be of direct benefit to BI/query workloads. These include

- **CPU Reductions**. The significant CPU reductions described on page 6 are directly applicable to analytics workloads, and are expected to provide an immediate CPU/cost reduction of 10-40% for complex reporting queries.
- **Optimiser Enhancements**. Analytics queries are frequently complex, and access large amounts of data. Many of the optimiser enhancements outlined on page 21 will result in faster, more efficient performance for query workloads as well as traditional OLTP.
- **Temporal Enhancements**. This functionality has already proven very valuable in many warehousing / analysis environments, which commonly have to support a historical perspective. The DB2 11 enhancements are described on page 15.
- **Transparent Archiving**. This new capability could provide a useful alternative or extension to the use of temporal

tables for managing historical warehouse data. It is described on page 13 of this document.

- **SQLPL enhancements for global variables and variable arrays**. These new features are described on page 17 and 18, and should be of considerable interest within BI environments for the additional flexibility they offer when constructing ETL processes.

Upgrading to DB2 11

This section outlines some of the high-level considerations around the timing and structure of the DB2 11 upgrade process.

DB2 Version Prerequisites

DB2 10 for z/OS broke with recent tradition by allowing customers to upgrade either from DB2 9 or from the earlier DB2 Version 8. IBM made it clear at the time that this was an extraordinary measure, specifically intended to help customers who had been unable to upgrade their Version 8 systems due to budget pressures caused by the global economic downturn.

True to its word, IBM is not offering a "skip migration" option for DB2 11: customers must be running DB2 10 in New Function Mode in order to begin the upgrade process. This means that customers still running DB2 9 (or earlier) systems must complete an upgrade to DB2 10 before they can begin planning for a further upgrade to DB2 11. Figure 41 provides an overview of a possible upgrade decision process, based upon the currently used DB2 release level.

IBM is not offering a "skip migration" option for DB2 11. Customers still running DB2 9 (or earlier) systems must complete an upgrade to DB2 10 before they can begin planning for a further upgrade to DB2 11.

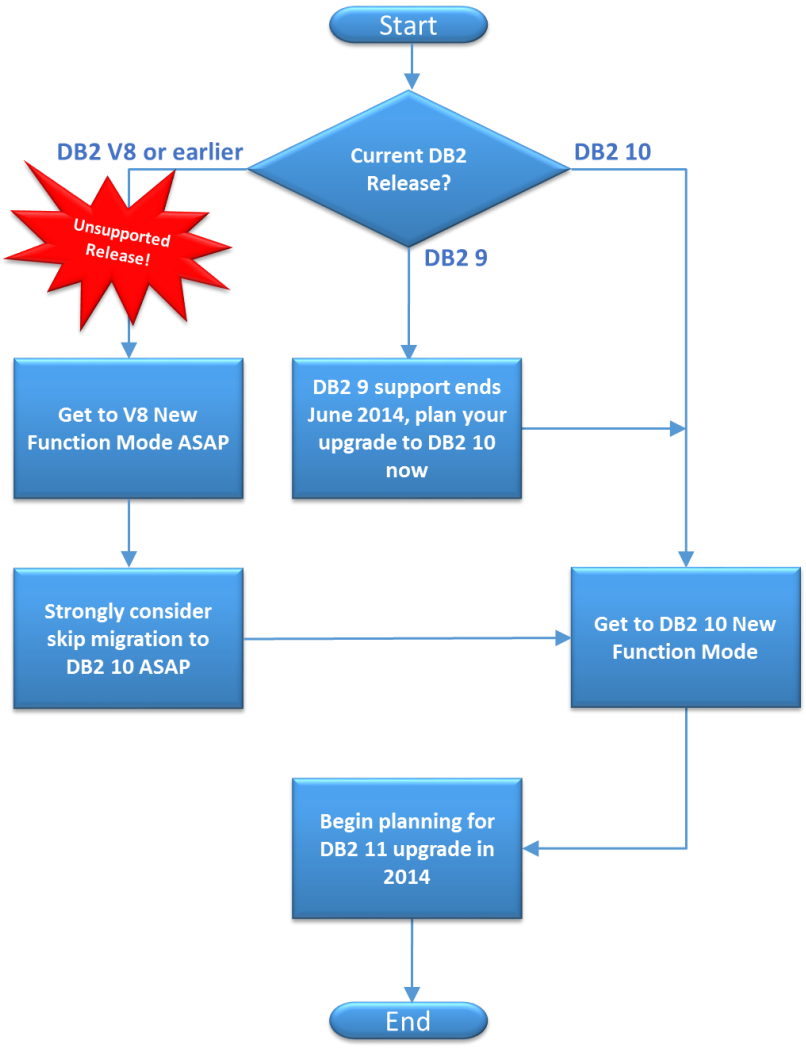

Figure 41 –DB2 11 Upgrade Decision Process

Support for DB2 9 will be withdrawn in June 2014. It is important for all remaining DB2 9 customers to begin planning to upgrade to DB2 10 as soon as possible.

Important note for DB2 9 customers: IBM has announced that support for DB2 9 will be withdrawn in June 2014. As most DB2 upgrade projects require 6-18 months to complete, it is important for all remaining DB2 9 customers to begin planning to upgrade to DB2 10 as soon as possible.

Other Prerequisites

In addition to the requirement to be running DB2 10 in New Function Mode, a number of other conditions must be met in order to upgrade to DB2 11. These are summarised below:

- **Hardware Requirements**. DB2 11 requires a zEC12, z196 or z10 server supporting z/Architecture. As outlined on page 10, some DB2 11 features such as 2GB page frames are only supported when running on a zEC12 server. DB2 11 is also expected to require increased real storage for a given workload when compared to DB2 10 for z/OS.

- **System software Requirements**. DB2 11 requires z/OS V1.13 or higher[30]. Some of the features described in the section on Security enhancements on page 37 also require z/OS Security Server (RACF) V1.13.

- **Application Requirements**. The new DB2 11 Application Compatibility feature (described on page 11 of this document) removes the usual requirement for application code to be altered to address incompatible changes. Although the code will still have to be changed at some point in the future, this is no longer a pre-requisite for upgrading to the new release.

  DB2 11 does not support packages bound prior to DB2 9, so these will need to be rebound in preparation for the upgrade. However, the plan management features introduced in DB2 10 should significantly reduce the effort and risk associated with this activity (please refer to the section on BIND / REBIND enhancements on page 36 for a discussion of the DB2 10 and DB2 11 plan management features).

Upgrade Timing

Most DB2 customers will wait for at least 12-24 months after a new release becomes available before beginning their upgrade project, depending upon their internal policies and the sophistication of their regression testing.

As shown in Figure 42 below, adoption rates for DB2 10 were significantly higher than normal due to the significant business benefits that were quickly available within that release.

[30] z/OS 1.13 has been available since July 2011, so this requirement should not be an issue for most customers.

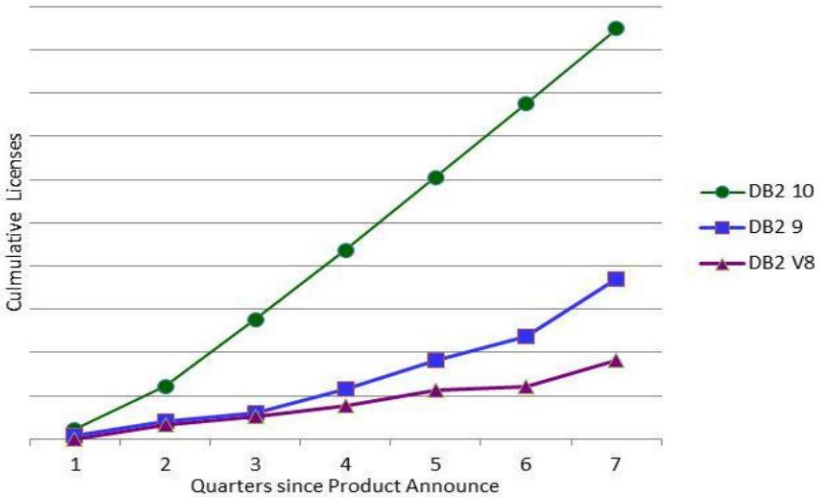

Figure 42 –DB2 10 Customer Adoption

The emphasis on early production-readiness, combined with the significant business benefits outlined within this document, are expected to result in higher than average customer adoption rates for DB2 11.

Recognising the requirement for customers to be able to exploit enhancements and obtain business value as soon as possible, IBM placed additional emphasis on code quality throughout the DB2 11 Early Support Program with the objective of allowing a limited number of customers to upgrade their Production systems before General Availability.

Each customer must carefully assess the business case for their specific environment in order to determine the timing for their DB2 11 upgrade project. However the emphasis on early production-readiness, combined with the significant business benefits outlined within the main body of this document, are expected to once again result in higher than average customer adoption rates for DB2 11.

Upgrade Process and Impact

The process of upgrading a DB2 environment from Version 10 to Version 11 follows the same overall structure as previous releases: each DB2 system is first moved into DB2 11 “Conversion Mode” (CM) before entering “New Function Mode” (NFM) via an intermediate step known as “Enabling New Function Mode” (ENFM). Most new releases require changes to be made to the DB2 catalog and directory (typically upon entry to the CM and ENFM phases) and this can cause some disruption to application workloads[31].

Although the IBM recommendation is to schedule a full DB2 outage during the upgrade, recent releases have made it possible to reduce the operational impact by permitting some application access to continue during the upgrade.

Although the IBM recommendation is to schedule a full DB2 outage during the transition to CM and ENFM, business demands for true 24x7 availability are making it increasingly difficult for customers to do so. Therefore, recent releases have made it possible to reduce the operational impact by permitting some application access to continue during the upgrade[32].

Further progress in has been made in this area within DB2 11, especially with regard to the second set of catalog changes made on entry to the ENFM phase. This process is expected to require considerably less elapsed time than the DB2 10 equivalent, and also

[31] The degree to which application access is disrupted depends on a number of factors such as whether data sharing is used, the size of the DB2 catalog/directory and the type of application workload that active during the upgrade.

[32] In order to minimise application impact, the catalog update process should be scheduled during a period of low DB2 system activity, when no changes are being made to the catalog by applications. This means that DDL and BIND/REBIND activity should not be allowed, but a small volume of normal application workload is acceptable.

One test conducted by IBM against a large catalog from an ESP customer showed an 18x elapsed time improvement for catalog changes when compared to DB2 10.

to be less disruptive while it is executing. One test conducted by IBM against a large catalog from an ESP customer showed an 18x elapsed time improvement for the ENFM catalog changes when compared to DB2 10.

Figure 43 below shows the impact of the DB2 11 upgrade process for non-data sharing customers who wish to minimise the operational disruption associated with the upgrade. A complete outage is still required in order to stop and start the DB2 subsystem and run the initial catalog update upon entry to Conversion Mode, but full availability can then be restored for the duration of the Conversion Mode phase. Further catalog updates are required during the ENFM phase, but these can run alongside some normal application workload if required[32].

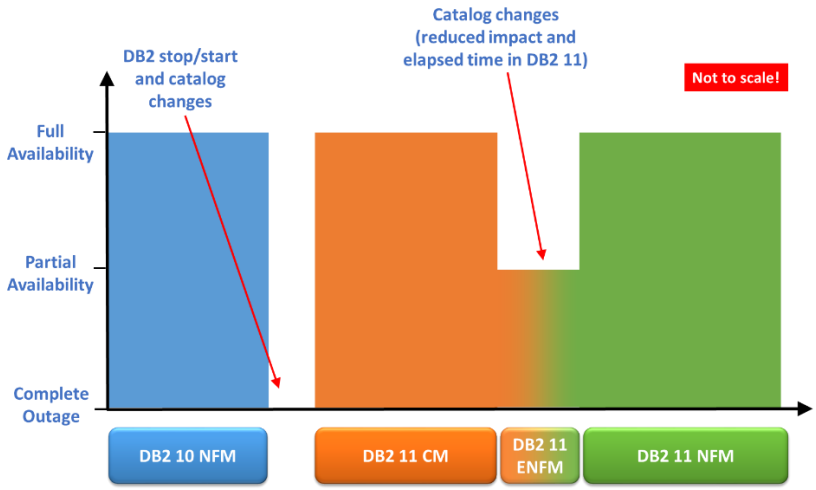

Figure 43 – Application Availability During DB2 11 Upgrade (Non Data Sharing)

Figure 44 shows the equivalent situation for data sharing environments. In this case, a complete DB2 outage can be avoided as one member of the data sharing group can be transitioned to Conversion Mode at a time, allowing applications to continue accessing the other members of the data sharing group.

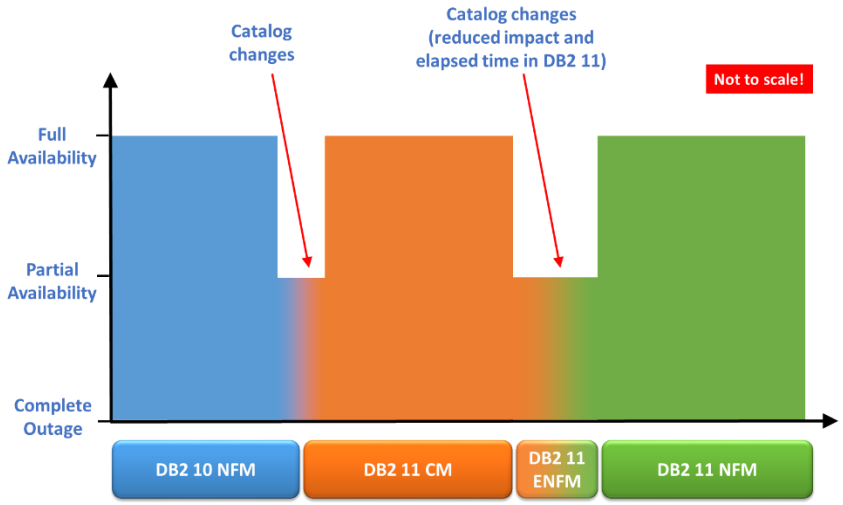

Figure 44 – Application Availability During DB2 11 Upgrade (Data Sharing)

The enhancements within DB2 11 further reduce the operational impact for customers choosing to conduct an online migration.

Although the IBM recommendation remains to schedule a dedicated outage during the transition to CM and ENFM, the enhancements within DB2 11 further reduce the operational impact for customers choosing to conduct an online migration.

DB2 11 Customer Case Studies

The section that follows is based upon interviews with some of the organisations that participated in the DB2 11 11 Early Support Program (ESP). Based on their early experiences with the product, they outline the business benefits they expect by exploiting the features in the new release.

BMW Group

As one of the world's biggest and 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 German car company has over 130 DB2 subsystems, belonging to over 40 data sharing groups and spread over eight z/OS LPARs.

"Virtual storage isn't a big limitation for us any more, but we expect the CPU savings in DB2 11 to provide the major business benefit for us."

BMW Group DB2 for z/OS Product Manager

BMW Group is no stranger to Early Support Programs. It participated in the ESP for DB2 for z/OS V2.1 way back in 1988, and more recently it was a key member of the DB2 10 for z/OS beta which ran for most of 2010. "We participated in the DB2 10 ESP because at that time we were facing some real challenges with workload growth", said BMW Group's DB2 for z/OS Product Manager. "The CPU savings and virtual storage constraint relief offered by DB2 10 really helped us to cope". Fast forward to mid-2013 and BMW Group is still enjoying healthy growth across its major markets, with the all of the associated pressures that can bring to the supporting IT infrastructure. "Virtual storage isn't a big limitation for us any more, but we expect the CPU savings in DB2 11 to provide the major business benefit for us", he added. "We are very early in our performance testing, but we have already seen CPU reductions of 8-13% on some workloads due to the more efficient decompression algorithms".

As part of the strategy to cope with the ongoing workload growth, BMW Group will also be upgrading their existing IBM system z10 EC servers for the latest generation zEC12 mainframes in the second half of 2013. BMW Group were early adopters of the 1MB large page frame support introduced by DB2 10 and the DB2 for z/OS Product Manager is looking forward to the opportunity to evaluate the benefits of the larger 2GB page frames supported by DB2 11 when running on the new servers. "We use very large bufferpools within all of our production DB2 systems, so we're anticipating further CPU savings when we can begin using 2GB frames".

BMW Group employs a highly centralised IT infrastructure, supporting all of its global business activities from within the Munich data centre. That makes getting dedicated change slots difficult, so some of DB2 11's availability improvements are also going to be most welcome for the DB2 team. "We operate in a very dynamic 24x7 environment and we often have to update our applications while they continue to operate. The ability to break into persistent threads and the enhanced dynamic schema change capabilities will allow us to react to business requirements more quickly and with less operational impact".

Another member of the BMW Group DB2 Team has been impressed with the stability of the new release. “We have encountered fewer install problems than we did at this stage with DB2 10, and our critical ISV tools are also stable so overall we’re very pleased with code quality. Support from the local IBM team, the IBM moderators and the lab members has also been first class”.

Perhaps the ultimate vote of confidence in the new release can be found in BMW Group’s provisional plans for rolling out DB2 11 once the ESP program ends. “We are currently scheduled to begin upgrading our main DB2 systems in February 2014” said the team member, “with the objective of completing the rollout in plenty of time for our annual change freeze the following November”. With DB2 11 likely to be less than 6 months old when the rollout begins, that’s a powerful endorsement of the business benefits and product stability that BMW Group are expecting.

Stadtwerke Bielefeld GmbH

Bielefeld is a large city in North-Rhine Westphalia, the most populous state of Germany. As one of the largest municipal enterprises within the German energy market, Stadtwerke Bielefeld is responsible for the delivery of electricity, gas, district heating and drinking water to the city’s 328,000 citizens. The company’s IT department runs IS-U, SAP’s industry-specific solution for the utilities industry, and this depends upon DB2 for z/OS as it’s back end data store.

Stadtwerke Bielefeld currently has around sixty active DB2 10 subsystems supporting SAP, spread across four LPARs of an IBM z196 enterprise server. After hearing about how DB2 11 could benefit the IS-U application at an SAP customer event, DB2 DBA Bernd Klawa recommended that Stadtwerke Bielefeld get involved in the Early Support Program. “The main reason to get involved in the ESP program is new functionality for the SAP Software” said Bernd. “The software layer for de-compression is very valuable for SAP databases, since all Unicode tablespaces are compressed.”

“The SAP IS-U unbilled revenue batch workload showed an elapsed time reduction of about 20% in Conversion Mode.”

Bernd Klawa, Stadtwerke Bielefeld DB2 DBA

Although the interview with Bernd was conducted before detailed performance metrics were available, he was able to share some encouraging initial results. “RECOVER of catalog and directory runs more than twice as fast as on DB2 10. The SAP IS-U unbilled revenue application (a batch workload) showed an elapsed time reduction of about 20% in Conversion Mode. After migration to DB2 11 New Function Mode I saw automatic index pseudo delete clean-up, which should reduce the need to REORG after big batch runs.”

Bernd also successfully tested the process of converting to DB2 11’s new longer 10-byte log addresses. “The old 6-byte log address is not really an issue for our data centre, however the conversion will still be done when we move to DB2 11 for improved logging performance.”

The experiences of Stadtwerke Bielefeld match those of many other ESP customers when it comes to the quality and stability of the new release. “I have been working with DB2 since Version 6 and I never saw such a robust release in such an early stage of development. The quality is very good” commented Bernd.

What about Stadtwerke Bielefeld's plans to upgrade their main DB2 systems to DB2 11 when the ESP program ends? Bernd is naturally keen to exploit the benefits within DB2 11 as soon as possible. "I expect to move to NFM in development and quality assurance systems as soon as DB2 11 is certified for usage by SAP."

JN Data

JN Data specialises in the provision of IT operations and engineering for large Danish financial institutions, including Jyske Bank, Nykredit, Bankdata, BEC and SDC. DB2 for z/OS is a key component within this shared infrastructure, with around 100 DB2 data sharing members hosting development and production services for JN Data's customers.

The company recognises the importance of staying abreast of the latest developments within critical infrastructure components such as DB2 so that it can continue to deliver the best service to its clients. JN Data was a prominent member of the DB2 10 Early Support Program (ESP) and was keen to repeat the experience when the DB2 11 ESP was announced.

Although new development features such as transparent archiving are important, according to Systems Programmer Frank Petersen it's the operational enhancements that have the most immediate business value. "Features such as temporal tables and transparent archiving can save many hundreds of developer hours, but there is often quite a delay before a suitable application comes along to use them" he said. "Operational items such as the ability to interrupt persistent threads can be used almost immediately, and will make a much bigger impact on us in the short term. "I expect this will allow us to use some BIND parameters that should give some significant CPU savings and especially for highly used packages driven through persistent threads which are used widely in our online systems".

What about other operational enhancements in DB2 11? "The move to 10-byte LRSN/RBA log addresses will also be important for us" said Frank. "We hit the RBA issue on one of our DB2 systems a while ago and had to take manual action to resolve it. We're expecting to encounter the same problem again within the next couple of years, so it's great to see a properly-engineered solution from IBM."

"We love autonomics. DB2 11 has some really nice features for reducing the burden on the DBA."

Frank Petersen, JN Data DB2 Systems Programmer

Like their counterparts in many other organisations, JN Data's DB2 support staff never seem to have enough hours in the day. "We love autonomics. DB2 11 has some really nice features for reducing the burden on the DBA, such as the automatic clean-up of pseudo-deleted index entries". Sticking with the operational theme, Frank is also impressed with the improvements in the DB2 11 upgrade process, which make online upgrades more feasible. During their ESP testing JN Data used IBM's Optim Query Capture Replay (OQCR) product to capture a few hours of workload on a real production system, then replayed that workload while attempting to upgrade one of their test systems to DB2 11. "It just worked, and the elapsed time for the catalog updates was much lower than before" said Frank. "Getting dedicated change slots for DB2 upgrades is a huge challenge with the number of systems we support. This will allow us to upgrade at a quiet time but without making the

application unavailable. We'll be performing online upgrades for our main DB2 systems when we move to DB2 11."

Other operational pressures will prevent JN data from beginning their DB2 11 upgrade project until 2014, but Frank is confident that DB2 will be ready when they are: "The ESP code mostly just worked from day one. The difference in code quality between DB2 11 and previous versions at this stage of the ESP is very noticeable".

Appendices

Appendix A – DB2 10 Review

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 10 has been generally available since October 2010, 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 10 and look at some of the ways in which it helps to deliver competitive edge. For a more detailed review of DB2 10, please see the 2010 Triton White Paper ***DB2 10 for z/OS – A Smarter Database for a Smarter Planet*** [3]

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.

- **CPU Reductions**. 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 being required to applications and further savings being possible with some database and/or application changes. Most customers migrating to DB2 10 have seen CPU savings within this range, which constitute 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 see these savings without any application changes being required.

- **Temporal Tables**. Many IT systems need to keep some form of historical information in addition to the current status for a given business object. 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. The new temporal data support in DB2 10 provides this functionality as part of the core database engine, potentially saving many hundreds of hours of design, coding

and testing that would otherwise be required to build this function manually for each application. While 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 Growth section below. In addition to supporting workload growth and providing more flexibility, these enhancements can also deliver some significant performance benefits, including reductions in data sharing overhead due to subsystem consolidation and improved dynamic statement caching due to removal of virtual storage constraints. 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. 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. Hash access tables are not without their disadvantages, but for some 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. 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, it was necessary to create a separate additional index. DB2 10 allows columns other than the unique key to be specified in the index definition. 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.

- **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 introduced a number of new enhancements for buffer pools which offer significant performance benefits. These include large page frames, in-memory pagesets and on-demand memory allocation.

- **Dynamic Statement Cache Enhancements.** 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, requiring the developer to take specific steps to handle parameters etc. DB2 10 delivered 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 feature can 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. DB2 10 added a number of additional capabilities including the ability to alter tablespace type, add MEMBER CLUSTER and alter index size, tablespace size, dataset size and segment size. 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.

- **Optimiser Enhancements**. DB2 10 included 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 optimiser enhancements further consolidated 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.

- **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, but did not support the MEMBER CLUSTER option. DB2 10 removes this restriction and allows MEMBER CLUSTER to be specified for UTS table spaces. This gives 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, but 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 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. For the first time DB2 10 provided DBAs with the option of backing out changes in the event of a database recovery, in addition to the more traditional recover/roll forward approach. Other related enhancements in DB2 10 remove some of the restrictions on use of FlashCopy[33] technology, and provide more options for the production of “consistent” backups without impacting application availability.

- **Other Efficiency Enhancements**. These include distributed access optimisations, native SQL procedure enhancements, improved zIIP exploitation, SSD support, parallel page updates for INSERT processing and inline large objects (LOBs).

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

[33] 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.

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 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. The DB2 10 plan stability feature removed 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.

- **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. DB2 10 provided 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.

- **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 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. DB2 10 introduced a number of new system authorities designed to allow proper separation of administration and data access. This increased flexibility allowed many sites to reduce their use of 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. DB2 10 introduced new capabilities for row and column level access control which are fully integrated into the database engine and defined using standard SQL constructs. These enhancements allowed security logic to be fully separated from application logic, providing significant additional flexibility and ensuring consistent security behaviour regardless of the mechanism used to access the data.

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.

- **Increased Scalability Though Full 64-bit Exploitation**. 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. Prior to DB2 10, virtual storage constraints were the single biggest factor in limiting the scalability of a DB2 system. In DB2 10, IBM completed the bulk of the work to make DB2 a true 64-bit RDBMS, first started back in DB2 V8. This enabled a spectacular increase in the number of threads that can be supported by a single subsystem – most customers have been able to achieve 5-10 times the number of concurrent connections compared to DB2 9. In addition to the cost and performance benefits, this vastly improves DB2's ability to support very high-volume workloads.

- **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 introduced several additional enhancements to pureXML functionality, including XML schema validation, partial update, binary XML format and CEHCK DATA support. Collectively, these enhancements addressed 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 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 included some important enhancements to the

Catalog, including migration to UTS format, and support for larger catalog sizes.

- **Other growth enhancements**. DB2 10 included a number of other enhancements designed to improve scalability and support future workload growth, including latch contention relief, support for extended address volumes, and timestamp enhancements.

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 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.

- **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. DB2 10 introduced support for moving sums, averages and aggregates, extending the OLAP SQL functionality previously delivered in DB2 9. 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. In recognition of this trend, IBM developed an innovative solution that combined 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, this solution is deeply integrated with DB2 10 and allows DB2 to offload eligible query components to the locally attached blade. This architecture is able to deliver up to 10x performance gains for qualifying queries 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 was released at the same time

as DB2 10 for z/OS and provided many new or enhanced facilities.

Appendix B – DB2 11 New Features by Implementation Effort

One of the most compelling features of DB2 11 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 11 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 11, with no database or application changes required. A REBIND may be 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/System 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.

[34] Also requires DB2 to be running on a zEC12-class server.

Appendix C – Acknowledgements

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

Mengchu Cai	DB2 for z/OS Development
John Campbell	Distinguished Engineer, DB2 for z/OS Development
Akiko Hoshikawa	STSM, DB2 for z/OS Performance
Namik Hrle	Distinguished Engineer, DB2 for z/OS Development
Terrie Jacopi	Program Director, DB2 for z/OS
Jeff Josten	Distinguished Engineer, DB2 for z/OS Development
Bernd Klawa	Stadtwerke Bielefeld GmbH
Sheryl Larsen	World Wide DB2 for z/OS Evangelist
Surekha Parekh	Worldwide DB2 for z/OS Marketing Program Director
Frank Petersen	DB2 Systems Programmer, JN Data
Terry Purcell	STSM, DB2 for z/OS Development
Jim Reed	Program Director, IBM Information Management
Maryela Weihrauch	Distinguished Engineer, DB2 for z/OS Development
Jay Yothers	DB2 for z/OS Development

Bibliography

[1] J. Stuhler, "DB2 for z/OS 8.1: Driving Business Value," Triton Consulting, 2004.

[2] J. Stuhler, "DB2 9 for z/OS - Data on Demand," Triton Consulting, 2007.

[3] J. Stuhler, "DB2 10 for z/OS - A Smarter Database for a Smarter Planet," Triton Consulting, 2010.