DB2 10 FOR z/OS

The Smarter, Faster Way to Upgrade

John Campbell, Cristian Molaro,
and Surekha Parekh

DB2 10 FOR z/OS

The Smarter, Faster Way to Upgrade

John Campbell
Cristian Molaro
Surekha Parekh

MC Press Online, LLC
Ketchum, ID 83340

DB2 10 for z/OS: The Smarter, Faster Way to Upgrade

By John Campbell, Cristian Molaro, and Surekha Parekh

First Edition
First Printing—October 2011

MC Press Online, LLC
Corporate Offices
P.O. Box 4886
Ketchum, ID 83340-4886 USA

For information regarding sales and/or customer service, please contact:
MC Press
P.O. Box 4300
Big Sandy, TX 75755-4300 USA
Toll Free: (877) 226-5394

For information regarding permissions or special orders, please contact:
mcbooks@mcpressonline.com

ISBN: 978-1-58347-345-0

About the Authors

John Campbell (campbelj@uk.ibm.com) is an IBM® Distinguished Engineer reporting to the Director for z/OS® Development at the IBM Silicon Valley Lab. He has extensive experience of DB2® in terms of systems, database, and applications design. John specializes in design for high performance and data sharing. He is one of IBM's foremost authorities for implementing high-end database/transaction-processing applications.

Cristian Molaro (cristian@molaro.be) is an independent DB2 specialist and an IBM Gold Consultant. He was recognized by IBM as an IBM Champion in 2009, 2010, and 2011. His main activity is focused on DB2 for z/OS administration and performance. Cristian has presented many papers at numerous international conferences and local user groups in Europe and North America. He is coauthor of five IBM Redbooks® related to DB2, including the recent *DB2 10 for z/OS Performance Topics*, and he serves on the IDUG EMEA Conference Planning Committee.

Surekha Parekh (surekhaparekh@uk.ibm.com) is IBM's Worldwide Marketing Manager Program Director for DB2 for z/OS. She is responsible for market strategy, planning, and promotion of DB2 on System z®. Based in Warwick, United Kingdom, she is a passionate marketer with more than 25 years of business experience, and she is also passionate about DB2. Surekha represents IBM on the IDUG committee. IDUG is an independent DB2 user group with more than 16,000 members in more than 100 countries.

Contents

Preface

We hope you enjoy our second book on DB2 10 for z/OS, which brings a focus on migration planning and financial benefits.

As you know, DB2 10 is the most significant release IBM has shipped in more than a decade, and we have focused our efforts in this release around three clear themes:

- ✓ *Cost savings* to our customers
- ✓ *Simplifying* database management with more automation
- ✓ Providing *proven* technology

The objective of this book is to help customers plan their migration strategy effectively and efficiently by providing the right information, facts, and guidance. The knowledge you'll find in these pages will ensure not only that you are adopting IBM best practices but also that you will reap the cost savings and business benefits of DB2 10. We believe the guidance and direction captured here will help our customers gain competitive advantages by doing more with less—more business insight, more performance, and more operational efficiency with less cost.

The launch of this book is timely, with support for IBM DB2 for z/OS Version 8 retiring in April 2012. For the first time in the history of DB2 for z/OS, customers can *skip a level* and not only take advantage of features in DB2 9 but also realize the immediate cost savings in DB2 10. Upgrading to the latest version of database software can be a time-consuming, expensive, and confusing process. Deciding whether you are the right candidate to skip a release is a difficult decision unless you have access to the right information, facts, and guidance to help you make that judgment. This book will help simplify some of these decisions.

For the first time in the history of DB2 for z/OS, customers can skip a level and not only take advantage of features in DB2 9 but also realize the immediate cost savings in DB2 10.

This book is segmented into two parts:

I. Planning for DB2 10 for z/OS upgrade
II. Gaining the financial benefits of DB2 10 for z/OS

Part I focuses on the planning stage of upgrading to IBM DB2 10 for z/OS. This section will help customers plan their migration strategy effectively and efficiently and will help support a successful upgrade.

Part II focuses on providing insight into how the performance benefits of DB2 10 for z/OS can positively impact your costs. One objective of this section is to give you valuable information that leads to a more in-depth analysis of your organization's opportunities to gain the most out of DB2 10.

I want to make sure you are all aware of our virtual community, "The World of DB2 for z/OS" (*http://db2forzos.ning.com*), which is available 24x7 and currently has more than 800 members. This site is a one-stop shop and hosts a collection of online resources, including webcasts, whitepapers, blogs, virtual chats, videos, and much more. Complementing this resource is IDUG, the independent DB2 user community (*http://www.idug.org*), which hosts conferences in Asia Pacific, Australasia, EMEA, and North America each year. Both of these communities are free and open to all interested parties. They are a great way to meet like-minded individuals, have fun, and at the same time grow and develop your DB2 skills.

I would like to thank John Campbell and Cristian Molaro for writing this fantastic book. I thank you as well, and I hope you enjoy the book. Please do not hesitate to contact me directly if I can be of help to you in any way.

Surekha Parekh
World-Wide Marketing Program Director
IBM DB2 for z/OS
October 2011

Useful URLs

DB2 for z/OS Family
http://www-01.ibm.com/software/data/db2/zos/family

DB2 10 for z/OS
http://www.ibm.com/software/data/db2/db210

The World of DB2 for z/OS
http://db2forzos.ning.com

Surekha Parekh's DB2 Blog
http://surekhaparekh.wordpress.com

DB2 for z/OS LinkedIn Group
http://www.linkedin.com/groups?gid=2821100

DB2 for z/OS Twitter
http://twitter.com/IBMDB2

International DB2 Community
http://www.idug.org

IDUG Australasia
http://www.linkedin.com/e/vgh/2656369

IDUG EMEA
http://www.linkedin.com/e/vgh/2648157

IDUG India
http://www.linkedin.com/e/vgh/2656395

IDUG North America
http://www.linkedin.com/e/vgh/2633325

IBM Redbooks

DB2 10 for z/OS Performance Topics
http://www.redbooks.ibm.com/abstracts/sg247942.html

DB2 10 for z/OS Technical Overview
http://www.redbooks.ibm.com/abstracts/sg247892.html

Why Read This Book?

Michael Perera, Director, DB2 for z/OS

Why should customers read this book?

We collectively spend a lot of time talking about the details of the line item improvements of any release, but we rarely spend enough time analyzing and communicating the overall return and value that those line items can collectively provide back to the business with the proper migration planning. That's at the core of this book.

What new features/benefits will customers find in DB2 10 for z/OS?

Aside from the obvious performance benefits, the improved concurrency and virtual storage relief can provide immediate returns by doing more with less. Longer term, bi-temporal support can have a huge impact as you either retrofit existing application logic, extend existing applications, or build it into new applications to unlock new insights and/or better manage your data.

About Michael Perera

Currently the Director, DB2 for z/OS, Michael (mperera@us.ibm.com) is responsible for product management, development, support, and quality assurance of one of IBM's flagship products. Prior to his current role, he held a number of other positions across product management, project management, operations, consulting, marketing, and mergers and acquisitions.

Jeff Berger, DB2 Performance, Senior Software Engineer

What new features/benefits will customers find in DB2 10 for z/OS?

Everyone is excited about the big CPU performance improvements and improved plan management in DB2 10, but they overlook the big I/O improvements—faster disorganized index scans using list prefetch and faster log I/O among them. Using prefetch to eliminate synchronous I/Os is critical to improving query performance and reducing the need to use REORG. Also, among my favorite DB2 10 features is inline LOBs for those shops that deal with small LOBs.

About Jeff Berger

Jeff (bergerja@us.ibm.com) has worked on IBM mainframe performance for 32 years, specializing in the synergy between disk storage and DB2. He is a frequent contributor at various conferences and has written numerous articles and papers, including, most recently, contributions to the *DB2 10 for z/OS Performance Topics* Redbook. Jeff's latest project revolves around using DB2 10 with SSD and High Performance FICON to improve the performance of disorganized data and indexes, and as a basis for reducing the need for REORGs.

Surekha Parekh, WW Marketing Program Director DB2 for z/OS

Why should customers read this book?

This is a "must read" book for anyone who works with DB2 for z/OS. We all know how difficult it is to catch any time with John Campbell. John is one of the most knowledgeable people when it comes to DB2 10 migration strategy and planning. With this book, we have had a chance to share this incredible knowledge and experience with everyone. The second part of this book is written by Cristian Molaro, an independent consultant; his performance section adds additional value and highlights the potential cost savings that can be reaped by upgrading to DB2 10 for z/OS.

What new features/benefits will customers find in DB2 10 for z/OS?

With support for DB2 8 scheduled to end in April 2012, there has never been a better time to start planning your DB2 10 upgrade. There are many more reasons to upgrade to DB2 10, and users will see significant business benefits. DB2 10 for z/OS provides exceptional CPU savings for most workloads, directly improving your bottom line and profitability. This benefit is timely under the current economic environment, in which companies are looking at reducing costs and getting more for less.

About Surekha Parekh

Surekha (surekhaparekh@uk.ibm.com) is IBM's World-wide Marketing Program Director for DB2 for z/OS. She is responsible for market strategy, planning, and promotion for DB2 for System z. Based in Warwick, United Kingdom, she is a passionate marketer with more than 25 years of business experience in various business areas, including Development, Project Management, Finance, and Operations Management. Surekha represents IBM on the International DB2 User Group (IDUG) committee. She recently completed her MBA dissertation on Social Marketing and has created an active virtual DB2 for z/OS community: The World of DB2 for z/OS (*http://db2forzos.ning.com*).

Namik Hrle, IBM Distinguished Engineer

Why should customers read this book?

This new DB2 10 book is really exciting. It will definitely help customers plan their migration strategy and make the right choices. The performance section adds additional value and highlights the potential cost savings that can be reaped. I would highly recommend this book to the DB2 for z/OS community.

What new features/benefits will customers find in DB2 10 for z/OS?

DB2 10 addresses the key scalability challenges of modern workloads and is filled with features that benefit demanding enterprise applications. It is very hard to single out any particular item because there are so many attractive features across the entire spectrum of database technology.

I have great expectations in the area of overall performance and particularly from features that improve insert and query performance.

About Namik Hrle

Namik (hrle@de.ibm.com) is an IBM Distinguished Engineer in the IBM Boeblingen Laboratory. He is a member of the IBM Academy of Technology, Information Management Architecture Board, SWG Architecture Board Steering Committee, and many other IBM expert teams. Namik specializes in information management technology and its use by enterprise applications. He works closely with IBM labs worldwide and helps customers deploy and optimally exploit information management products.

Terrie Jacopi, Program Director, DB2 for z/OS

Why should customers read this book?

This book provides a wonderful overview on some of the ways DB2 10 can help you drive down cost while increasing performance. Once you understand the what and whys, you can use information in this book to create a business case on how DB2 10 can help your company. The first part of the book is an invaluable resource that identifies migration best practices from one of the most well-known experts, John Campbell. The knowledge gained can help you develop and deploy a successful DB2 10 migration plan.

What new features/benefits will customers find in DB2 10 for z/OS?

Along with CPU and thread relief savings, DB2 10 continues to deliver new technology to support the highest availability, reliability, and security requirements in the market. Included in that are simplified management capabilities that allow you to do even more with existing resources to meet growing business requirements. DB2 10 also delivers temporal capabilities, a first-in-the-market technology that can dramatically simplify key business applications.

About Terrie Jacopi

Terrie (jacopi@us.ibm.com) is the DB2 for z/OS Program Director at the IBM Silicon Valley Lab. She focuses on connecting the technical capabilities of DB2 and System z with the business value delivered to IBM customers.

Jeff Josten, IBM Distinguished Engineer

Why should customers read this book?

This book will help readers to understand the benefits that DB2 10 can bring to your business, especially the performance improvement opportunities. This book also gives pragmatic guidance for planning and executing the migration to DB2 10. There are no two better experts than John and Christian to present this information.

What new features/benefits will customers find in DB2 10 for z/OS?
DB2 10 delivers many performance improvements, and this translates directly into cost savings for many organizations. Many of the performance improvements can be harvested with no application changes and little or no DBA work. The support for bi-temporal data is an industry-leading innovation that can simplify application development and maintenance cost. And the scalability improvements in DB2 10 can allow some of our larger customers to consolidate their DB2 data sharing groups and to achieve significant performance and systems management savings.

About Jeff Josten
Jeff (josten@us.ibm.com) is an IBM Distinguished Engineer and lead architect for DB2 for z/OS. He has worked in the DB2 Development organization at the IBM Silicon Valley Lab since DB2 Version 2.1. His main areas of interest have included data sharing, performance, recovery, and locking. He owns several patents in the area of database technology and is a frequent speaker at DB2 user group meetings.

Julian Stuhler, Solutions Delivery Director, Triton Consulting

Why should customers read this book?
What can be said about DB2 10 for z/OS that hasn't already been covered in the many articles, papers, blogs, presentations, tweets, and books already devoted to the subject? As it happens, quite a lot. Most of the information delivered since DB2 10 became available has concentrated on the technical benefits and features in the new release, and there is certainly plenty to get excited about from that perspective. This book starts by making a very significant addition to that body of work, with John sharing his vast experience and detailed insights based on customer engagements all around the world.

However, technical information alone is not enough. With the ongoing challenges in today's global economic climate, it is more important than ever to justify any outlay in terms of specific and measureable savings. The second part of this book, by Cristian Molaro, provides the reader with many of the tools and much of the information necessary to begin constructing a solid business case to justify their DB2 10 migration project.

What new features/benefits will customers find in DB2 10 for z/OS?
DB2 10 for z/OS has got many customers excited, with its combination of solid technical enhancements and impressive potential cost savings. Many of my customers see the scalability enhancements as being the next most important item after the headline cost savings.

About Julian Stuhler
Julian Stuhler is Solutions Delivery Director and a Principal Consultant with Triton Consulting, which specializes in DB2 and IBM's other Information Management products. Julian is also an IBM Gold Consultant, an IBM Information Champion, and a Past President of the International DB2 User Group (IDUG). He has authored several white papers on new releases of DB2 for z/OS, including *DB2 10 for z/OS: A Smarter Database for a Smarter Planet*.

Introduction

In the current economic climate, businesses are under significant pressure to control costs and increase efficiency to improve their bottom line. DB2 for z/OS customers around the world are still trying to gain competitive advantage by doing more with less: more business insight, more performance, more operational efficiency, more functionality, more productivity with less cost, quicker time to market, and a lower TCO.

DB2 10 helps customers address key business issues by delivering innovations in the following key areas:

- **Improved operational efficiency for out-of-the-box savings.** Most customers can achieve out-of-the-box CPU savings of *5 percent to 10 percent* for traditional workloads and *up to 20 percent* for specific workloads.
- **Unsurpassed resiliency for business-critical information.** Uniquely integrated technology enables DB2 for z/OS and System z to support the efforts of keeping businesses running even when things go wrong or you need to make changes.
- **Rapid application and warehouse deployment for business growth.** SQL and pureXML® enhancements in DB2 10 help extend usability, improve performance, and ease application portability to DB2 for z/OS.
- **Enhanced business analytics and mathematical functions with QMF.** New analytic and mathematical functions and online analytical processing (OLAP) support dramatically enhance Query Management Facility's (QMF's) ability to deliver new function to business users.

We have seen many top 10 lists around DB2 10; in this section, we summarize the top 10 reasons why we feel it is the right time for customers to upgrade to DB2 10. We all know that *support for DB2 Version 8 is scheduled to end in April 2012*. However, there are many more reasons for the migration; users will see significant benefits from DB2 10.

Top 10 Reasons to Upgrade to DB2 10 for z/OS

Here are the top 10 reasons why you should upgrade to DB2 10:

1. Improve performance, reduce software license costs
2. Increase the number of concurrent users by a factor of 10
3. Reduce contention in database administration
4. Access more administrative capabilities while database is online
5. Improve security and auditing
6. Maintain "snapshots" of changing data (Temporal Data)
7. Improve portability via enhanced SQL
8. Enhance pureXML performance and usability
9. Improve productivity for database/systems administrators and application programmers
10. Better online transaction processing performance (Hash Access)

1. Improved performance and lower costs

IBM DB2 Version 10 improves performance, and that means fewer software licenses. Lower CPU utilization for transactions, queries, and batch processing yields performance improvements of 5 to 10 percent for traditional workloads (using a Type 2 connection) and 20 percent for new workloads (using a Type 4 connection through the Distributed Data Facility).

Other performance improvement features can be realized—without DBA intervention. These include:

- Improvements in the cost optimizer that factor in the uncertainty of predicate filtering when choosing between two indexes with similar cost
- Converting multiple `OR` predicates to a single range predicate to reduce the number of index scans
- Applying Stage 2 predicates (i.e., those applied after joins are completed) earlier, reducing the number of rows fetched for the join
- Enhanced parallelism that helps remove some DB2 V9 restrictions and automatically calls parallelism queries involving work files from view materializations
- Full outer joins
- Joining a sorted table with a second table using multiple column hybrid join; collecting statistics on tables and indexes autonomically, helping the cost optimizer for queries with fetch "one row only" clause

The number of interactions between the distributed relational data system and the host database is reduced from three to one. This means that, in one pass, the cursor is opened, the single row is fetched, and the cursor is closed. This functionality allows the sharing of dynamic SQL statements with a cached version, if the only difference is the literal value.

DB2 10 allows parallel insertions on tables with multiple indexes by pre-fetching pages in parallel from a table using multiple indexes. It also permits dynamic allocation of buffer pool storage and a parameter to mark an object using the buffer pool as in memory. This keeps the page resident on the buffer pool as long as the object remains open.

Indexes require less reorg and support the streaming of large objects (LOBs) and XML data, requiring less materialization and saving virtual storage.

There are improvements in Flashcopy® for backup and restore, achieved by copying the current object and then backing out uncommitted changes for consistency, versus recovering to a prior consistent copy and forward applying committed changes from the log.

The DBA lead performance improvement features including hash versus index access for static-sized tables with unique indexes and the use of inline LOBs for faster retrieval of small-sized binary objects. There are also improvements in access techniques to support more parallel access by reducing restrictions. More parallel access means more processing is offloaded to zIIP, further lowering costs.

2. Ten times more concurrent users

DB2 10 greatly improves the vertical scale-up of the DB2 subsystem by supporting 5 to 10 times more concurrent threads and up to 20,000 connections per subsystem. This is particularly advantageous for zGryphon, where the demand on DB2 for z/OS will be increased many times with applications, not only from zLinux but also from the application serving IBM System x® and System p® blades accessing DB2 on the host.

3. Reduce contention in database administration

DB2 10 provides more concurrency for catalog, utilities, and SQL, eliminating the use of UTSERIAL lock by DB2 utilities to prevent a timeout of the global UTSERIAL lock resource. UTSERIAL lock is replaced with more granular locks to reduce contention.

The catalog structure is modified to remove links and replace them with referential integrity. It allows row-level locking, versus page-level as in earlier versions, and converts large fields with repeating rows of data into inline LOB columns, eliminating the 64 GB limit.

The catalog tables are stored in individual partition-by-growth universal table spaces. Overall, contention among processes such as BIND, dynamic SQL, data definition, and utilities is lower.

4. More administrative capabilities while the database is online

DB2 10 includes more online changes for data definition, utilities, and subsystems. It supports online REORG of catalog and directory table spaces and schema changes for table spaces, tables, and indexes with a PENDING with ALTER and then an online REORG.

5. Improved security and auditing with more granularity

DB2 10 offers better granularity for managing security. You can manage table access at the column and row levels, and you can create multiple audit policies.

DB2 10 supports protection of sensitive data from privileged users, such as SYSADM or DBADM. Separate authority can be assigned to a security administrator for security-related tasks, and privileged users can also be audited.

Overall, access control is refined in several ways, including better granularity for administrative privileges and more precise access control at the row and column level, including the ability to mask access to some fields.

6. Maintain "snapshots" of changing data – Temporal Data

With DB2 10, temporal or versioned tables can be defined with system- and business-defined time periods for which start and end timestamp values for a row can be maintained. The system-defined time period is used for versioning the table, with current rows of data called the "system-maintained temporal table" and the archived, older version table called the "history table." By keeping two time periods (system and business), bi-temporal tables can be supported where system-based, historical information and user-defined "snapshots" can be maintained. This capability provides greater flexibility to query data, based on a given time period.

7. SQL enhancements to improve portability

With DB2 10, SQL enhancements including MERGE (allows updates and inserts of many rows of tables, from multiple transactions, with a single operation) and TRUNCATE (deletes all data rows in a table, without triggering DELETE trigger or altering the table attributes in the catalog) statements, INTERSECT and EXCEPT set operations, spatial support for geographical data, new DECFLOAT (decimal floating point with maximum precision of 34 digits), and VARBINARY (varying-length binary string) data types.

Several other enhancements improve application portability, such as:

- Currently committed locking semantics
- Implicit casting or loose typing
- Timestamp with time zone
- Variable timestamp precision—seconds to picoseconds
- Moving sum
- Moving average
- Non-null default values for inline LOBs
- And much more

8. pureXML performance and usability enhancements

DB2 10 adds an XML-type modifier composed of multiple schemas to an XML column to enforce the validity of the XML data. XML schema validation is provided as a built-in function and does not require a schema to be specified for validation. DB2 finds a schema automatically, from the schema repository.

DB2 10 has a CHECK DATA utility to verify the consistency of XML documents to make sure all XML documents of an XML column are valid against at least one XML schema specified in the XML-type modifier.

DB2 10 also supports multiple versions of XML documents for the same XML column. It has an XMLMODIFY built-in, scalar function that can be used to update parts of an XML document (i.e., INSERT, DELETE, or REPLACE nodes or values; supports binary XML format for interchange of XML data between applications and the database server) using Java™ Database Connectivity (JDBC), Structured Query Language for Java (SQLJ), or Open Database Connectivity (ODBC) connections.

The binary format uses a pre-tokenized format with all values at a pre-fixed length. This means there is no need to perform a byte-wise search for end-of-element names to look for values.

Date and time support for XML data types in DB2 10 supports time zones and arithmetic and comparison operators on date and time data types.

XML is supported as data types for parameters in native SQL procedures, SQL scalar functions, and table functions. With the DEFINE(NO) option, creation of XML and LOB table spaces can be deferred until the first INSERT or LOAD operation.

9. Productivity improvements for database/systems administrators and application programmers

DB2 10 offers improvements in DB2 QMF and an array of DB2 tools and new tools, such as the Optimization Service Center and DB2 Accessories Suite, to help make DBAs even more productive.

Temporal, or versioned, data improves productivity for applications such as SAP and data warehouses. Some of the productivity benefits are realized through auto statistics, easier scaling with simpler memory management, reduced contention, more online processing, access path stability, reduced need for REORG, and enhanced monitoring.

10. Faster OLTP performance – Hash Access

DB2 10 introduces a new access path, called *hash access*, for faster access to individual rows by using a fully qualified key and not a traditional index. Hash access helps reduce the load on the CPU, but it requires additional storage space to maintain a hash space.

Hash access paths do not take advantage of parallel I/O, CPU, or Sysplex structures. Hash access also can eliminate the need for table space scans and index scans for access to a single, unique key. If a good hash algorithm is used, only one I/O operation is required to access a single row. Hash access is efficient for certain types of tables and queries, such as queries that use equal predicates to access a single row on a table of a predictable and reasonably static size.

To create an effective hash algorithm, DB2 needs a close estimate of the volume of data that is expected to be in a table before the table is populated.

Restriction: Note that if a table is organized for hash access, index clustering is not available.

DB2 10 also provides the ability to alter and create unique indexes that contain additional non-key columns. The INDEX function is expanded by adding the optional INCLUDE clause to the CREATE INDEX and ALTER INDEX statements.

The use of INCLUDE columns is supported only on unique indexes, for the purpose of decreasing the execution time of DB2 transactions and the amount of physical storage required for additional indexes.

The SKIP LOCKED DATA option was added to allow some functions such as UNLOAD, DELETE, PREPARE, and SELECT INTO to skip rows on which incompatible locks are held by other transactions.

Hear What Our Customers Are Saying About DB2 10 for z/OS

DB2 10 for z/OS: Reduces Costs

"Based on the performance metrics from our controlled test environment, we see a significant amount of CPU and elapsed time savings. This release has many features that will help bring down our operating costs. We look forward to implementing DB2 10 in our environment early next year."

Morgan Stanley DB2 Team

"Over the past several months, BMW has tested the new version of DB2 10 for z/OS, focusing on specific features and comparing these directly with the same features in DB2 9 for z/OS. One of the IBM design goals expected a general improvement in massive parallel SQL-insert performance, where we achieved close to 40 percent CPU improvement and significant elapsed time reduction in direct comparison with DB2 9 for z/OS. For all of our critical tested select statements, the Version 10 optimizer chose the optimal access path, sometimes even improving previous access path choices in Version 9. Overall, we are very pleased with the added functionality and architectural enhancements and are looking forward to this exciting release."

Philipp Nowak, BMW Group DB2 Product Manager

"We are really excited about the plan stability enhancements in DB2 10. We have a sophisticated procedure to assure quality in access paths; this means all packages run with their optimal access path. To assure this, we have to look at each new statement and maybe at each new rebind—this can be a time-consuming, partly manual process. Plan stability offers an opportunity to reduce the daily effort and to improve the DBA's productivity."

Walter Janissen, ITERGO Informationstechnologie GmbH

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

DB2 10 Beta Customer – Large Insurance Company

DB2 10 for z/OS: CPU Reductions for Most Workloads

"We use very large buffer pools—some of them up to 3.2 GB in size. We rely on efficient access to buffered data, and any saving in the cost of accessing that data will be very beneficial."

Philipp Nowak, BMW Group DB2 Product Manager

"We are really thrilled about the Temporal Data feature. This feature has the potential to significantly reduce overheads. We have estimated that 80 percent of our existing temporal applications could have used the DB2 10 temporal features instead of application code; this feature will drastically save developer time, testing time, and—even more important—make applications easier to understand and so improve business efficiency and effectiveness."

Frank Petersen, System Programmer, Bankdata

"We feel that performance, scalability improvements, and security enhancements will really benefit our business. We are particularly interested in the performance improvements due to the potential CPU reductions that we realized during our DB2 10 Beta testing. Our early testing has shown out-of-the-box processing cost reductions of between 5 and 10 percent and for some workloads as high as 30 percent. Potential cost savings of this magnitude cannot be ignored given today's business climate."

DB2 for z/OS Beta Customer – Global Bank

"With the optimizations in DB2 10, we are hoping to see a significant CPU reduction for our main SAP workloads, perhaps enough to defer our annual hardware upgrade cycle."

Guenter Schinkel, Postbank Systems AG

"24/7 availability in parallel with reducing our mainframe total cost of ownership are two of our top operating priorities. DB2 10 for z/OS has numerous new features, such as increasing the capacity of DB2 and more efficient data access, that will assist us in achieving our goals."

Anthony Ciabattoni, Engineer Architect DB2, **Fiserv**

DB2 10 for z/OS: Simplifies Workload

"With IBM QMF 10 there are many enhancements that will make life easier. The biggest improvement for us was the enhancement regarding SAVE DATA support. We force customers to run their CPU-intensive queries in batch, and in prior releases they failed after running for hours just because the table already existed and there were minor differences in the column definition. With the new release, these failures are reduced; as a result, this improves productivity, saves cost, and improves customer satisfaction."

Walter Janissen, ITERGO Informationstechnologie GmbH

"We are excited about the scalability enhancements in DB2 10. In a failover situation, we currently have to be very careful about the amount of work that gets directed to a given DB2 system. The increased scalability in DB2 10 will allow us to simplify our failover automation and remain confident that DB2 can handle the increased workload."

Guenter Schinkel, Postbank Systems AG

"With the scalability improvements in DB2 10, we expect to be able to quickly reduce our production data sharing group from 20 members to 15. With DB2 10 able to handle 5 to 10 times as many threads as the previous version, the upgrade will immediately give the bank some much-needed room for future workload growth while simultaneously reducing our data sharing overhead. We will also save some CPU and storage from removing the five DB2 systems, and we will have to spend a lot less time monitoring our virtual storage."

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

DB2 10 for z/OS: Proven Technology

"The security enhancements answer a number of audit and compliance requirements that we have. There is an increased focus on reducing the capabilities for even the most powerful DB2 users to have the potential to read customer data. In addition, when or if they have this capability, there are strict auditing guidelines that mandate that any read or update of customer data be justified via a documented change ticket signed off by the business, and that this change ticket is then reconciled with an audit log that documents all of the actions taken during the change. This reconciliation must be done by the person who performed the change, and then a third party must verify that the contents of the change ticket and the audit logs match. DB2 10 contains a number of features that will allow us to satisfy these requirements. This will improve employee productivity and save costs."

DB2 for z/OS Beta Customer - Global Bank

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

Guenter Schinkel, Postbank Systems AG

"Not a big iron anymore, but an efficient, scalable, reliable, open, and business-oriented platform; this is now the mainframe. . .and with DB2 10 and its new exciting features and functions doing more work with the same resources achieving the same high availability. This is a goal that can be easily achieved by every company. DB2 10 reduces total cost of ownership but keeps intact the power for future business opportunities."

Massimo Scarpa, CESVE, Servizi Informatici Bancari S.p.A Consortile

---

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

Philipp Nowak, BMW DB2 Product Manager

"I have over 17 years DB2 DBA experience on System z and over 10 years Oracle DBA experience. My experience includes database design and definition, performance tuning, and systems programming. I am really excited about DB2 10—besides the CPU savings and usual system performance and functionality improvements, which just get better with each version, DB2 10 for z/OS delivers an excellent and far superior data server environment for applications portability and enablement."

Manuel Gómez Burriel,
Confederación Española de Cajas de Ahorros (CECA)

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

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

Planning for IBM DB2 10 for z/OS Upgrade

by John Campbell

Executive Summary

In the spring of 2010, DB2 10 for z/OS was released to 24 worldwide customers for beta testing. The evaluation focused on regression testing, "out-of-the-box" performance, and additional performance and scalability, as well as other new functions.

Customer experience and feedback about the program have been mainly positive, and most customers who were involved in the program plan to start migration to DB2 10 for z/OS in 2011. An incremental improvement was observed in the effectiveness of the program, in terms of the quality of the issues and problems found, relative to the respective programs for DB2 Version 8 and Version 9. Some customers did very well with regression and new function testing; others provided only limited qualification about what they did and what they achieved.

After the early stages of planning and execution, it is often difficult for customers to sustain the effort required during a six-month period, due to competing business and technical priorities. People, hardware, and time are usually constrained to varying degrees. As of the end of the beta program, no customers were in "true, business production."

The release of DB2 10 for z/OS provides many opportunities for *price/performance and scalability improvements*. But there is a *tradeoff* in terms of *some increased real storage consumption*. Customers need to carefully *plan, provision, and monitor* their real storage consumption.

The new, 64-bit SQL runtime can provide generous, 31-bit virtual storage constraint relief in the DB2 DBM1 address space. This support provides enhanced vertical performance scalability of an individual DB2 subsystem or DB2 member. It also opens opportunities for further price/performance improvement, through greater use of persistent threads running with the BIND option RELEASE(DEALLOCATE), DB2 member consolidation, and LPAR consolidation.

Introduction

This paper focuses on the *planning stage* of migrating to IBM DB2 10 for z/OS. The key points of emphasis are:

- ✓ Make sure everyone is educated as to what is needed to ensure project success.
- ✓ Production of a detailed project plan, communicated to all involved, is crucial for success.
- ✓ Some preparation can occur very early, in terms of understanding, obtaining, and installing the prerequisites.

The release of DB2 10 for z/OS was announced on February 9, 2010, and began shipping on March 12, 2010. It was the largest beta test program in the history of DB2 for z/OS.

The information in this paper is drawn from the lessons learned in cooperation with 24 of IBM's largest customers, representing a variety of industries and countries around the world. An extended beta test program started in Q3 2010 and lasted for six months. The program also included 73 parties in vendor programs.

These customers were looking mainly for 31-bit virtual storage constraint relief in the DBM1 address space and all opportunities for price/performance improvement. Other areas of interest included:

- Regression testing (*Be sure to approach regression testing in the order in which you plan to move to production.)*
- "Out-of-the-box" performance
- Additional performance improvements
- Scalability enhancements
- New functions

Stages of migration

The primary stages of migration to a new version are:

1. Planning
   - » Early stages:
     - Making the decision to migrate
     - Determining what can be gained
     - Planning for prerequisites

- Avoiding incompatibilities
- Planning performance and storage
- Assessing available resources

2. Migration
3. Implementation of the new improvements

Needed application changes can be made over a longer period to make the migration process easier and less costly. Plans for monitoring virtual and real storage resource consumption, as well as performance, are necessary. An early health check, communication of the required changes, and staging of the work will make the project go much more smoothly.

Highlights of the Beta Test

DB2 10 for z/OS delivers great value by ***reducing CPU resource consumption in most customer cases***. IBM internal testing and early beta customer results revealed that, depending on the specific workload, many customers could achieve ***"out-of-the-box" DB2 CPU savings of up to 10 percent*** for traditional OLTP workloads and up to ***20 percent for specific new workloads (e.g., native SQL procedures)***, compared with running the same workloads on DB2 9 for z/OS.

The objective of providing and proving generous, 31-bit virtual storage constraint relief in the DBM1 address space was achieved by the end of the program. This achievement is significant in terms of the enhanced vertical scalability of an individual DB2 subsystem or DB2 member of a data sharing group. We are confident that customers can scale up, in practical terms, the number of active threads by 5 to 10 times to meet their demands.

Further opportunities for price/performance improvement are made possible through the use of persistent threads with the BIND option RELEASE(DEALLOCATE). Examples of using persistent threads include protected ENTRY threads with Customer Information Control System (CICS®), Wait For Input (WFI) regions with Information Management System/Transaction Manager (IMS/TM), and high-performance database access threads (DBATs) for incoming Distributed Data Facility (DDF) workloads.

Another goal was to improve INSERT performance, particularly in the area of universal table spaces (UTS). We wanted to ensure that insert performance for UTS was equal to, or better than, the classic table space types, such as segmented and partitioned. This goal was achieved in most cases.

Hash access was good, provided we hit the smaller-than-expected "sweet spot." Results for complex queries were also good.

Provided users chose the correct value, the performance of inline large objects (LOBs) was also impressive. Support for inline LOB column values has the potential to save even more on performance by avoiding indexed access to the auxiliary table space. However, it is important to note that the value you choose for the inline LOB value must ensure that most of the LOB column values are 100 percent inline in the base table space.

In the area of latch contention reduction, we focused on the hot latches in DB2 10 for z/OS in such a way that, once we solved the 31-bit virtual storage constraint in the DBM1 address space, enabling you to scale five to ten times, we wanted to be sure there were no secondary issues related to latch contention that would inhibit the vertical scalability of a single DB2 subsystem or DB2 member.

As the beta program progressed, the reliability of, and customer confidence in, DB2 10 for z/OS greatly improved.

Generally speaking, online transaction processing (OLTP) performance improvements were as predicted. We were aiming for a target of 5 percent to 10 percent reduction in CPU resource consumption for most traditional OLTP workloads. During testing, several customers ran benchmarks showing that such reductions could be achieved. However, in cases where the transactions consisted of a few very simple SQL statements, the 5 percent to 10 percent target was not achieved.

This is where the increase in package allocation cost outweighed the improvement in SQL runtime optimization. However, we did identify some steps that can be taken to improve this. We have delivered an Authorized Program Analysis Report (APAR) to reduce package allocation cost. It is also possible to mitigate this situation by making more use of persistent threads with the BIND option RELEASE(DEALLOCATE).

Another issue was single-thread BIND/REBIND performance. Even in Conversion Mode (CM), the performance, in terms of CPU resource consumption and elapsed time, was degraded. One reason for this result was that in DB2 10 for z/OS the default for access plan stability is EXTENDED. Also, DB2 10 for z/OS uses indexed access, even in CM, to access the respective DB2 Catalog and Directory tables.

Another area where we had mixed results was SQL Data Definition Language (DDL) concurrency. We had hoped that by restructuring the DB2 Catalog and

Directory to introduce row-level locking, remove hash link access, and more, we could improve concurrency when running parallel SQL DDL and parallel BIND/REBIND operations. The concurrency improvement was eventually achieved for parallel BIND/REBIND activity. Although it also helped in some cases with SQL DDL, most customers will still have to run SQL DDL activity single-threaded.

The final issue was access path lockdown. Two new options in DB2 10 for z/OS, APREUSE and APCOMPARE, enable you to generate a new SQL runtime while in most cases keeping the old access paths. Unfortunately, there were some issues with the underlying OPTHINTS infrastructure inherited by DB2 10 for z/OS, which is used by APREUSE and APCOMPARE. The introduction of APREUSE and APCOMPARE was delayed until these issues were addressed. These features are now available in the service stream via APARs, and their use is strongly recommended.

In general terms, the results of the beta program were mainly positive customer experiences, and we received good feedback about the program. A majority of customers in the beta program plan to start migrating to DB2 10 for z/OS in 2011. We observed incremental improvement in the program over what we experienced with the DB2 8 and DB2 9 for z/OS programs.

There was really no "single voice" or message across the customer set. We saw significant variation in terms of customer commitment and achievement. A small subset of customers did a very good job on regression and new function testing and provided good feedback. Others, due to limited resources, provided only limited qualification about what they were going to do and what they were able to achieve.

It is worth keeping in mind, for those who have never been involved in a Quality Partnership Program (QPP)/beta program, that it can be a challenge for customers to sustain the effort over a six-month period, due to competing business and technical priorities as well as constraints on people, hardware resources, and time.

By the end of the program, no customers were in true, business production. But we also need to appreciate that a QPP/beta program is *not* the same as an Early Support Program. We continue to develop and test the DB2 for z/OS product as the program progresses.

One of the benefits of DB2 10 for z/OS is that it provides many opportunities for price/performance (cost reduction) improvements. It is a major theme of this release. In discussions with customers, these opportunities for price/performance improvement are most welcome.

Also keep in mind that customers can be intimidated by some of the marketing "noise" about improved price/performance, often because of the raised expectation level of their respective CIOs. But in some cases, it is because when they run their own workloads, they do not see the anticipated improvements in CPU resource consumption and elapsed time performance. Many customers saw big improvements for certain workloads, while for other workloads, they saw little, if any, improvement.

Also note that if you have small test workloads that are untypical of the total mixed workload running in production, this can skew expectations on savings—either positively or negatively. Once DB2 10 for z/OS is in production, the results with the full, mixed workload may differ. We found that some measurements and quotes were overly positive and should be ignored.

A remaining question is: "How do you extrapolate from a small workload and project what the savings would be for the total, mixed workload in production?" Estimating with accuracy and high confidence is not practical, or possible, without proper benchmarking using a workload that truly represents production. Most customers reported incremental improvement over the DB2 8 and DB2 9 for z/OS programs.

Overall, most tests identified opportunities for price/performance (cost savings) improvements, which is the major theme of this release. Some customers reported big improvements in CPU and elapsed time reduction for certain workloads, while others did not. Keep in mind that smaller workloads may skew expectations on savings.

Summary of results

The DB2 10 for z/OS beta program confirmed improvements in the following areas:

- ✓ 31-bit virtual storage constraint relief in the DBM1 address space
- ✓ Insert performance
- ✓ Hash access good when hitting the smaller-than-expected sweet spot
- ✓ Complex queries
- ✓ Inline large objects (LOBs) and structured large objects (SLOBs)
- ✓ Latch contention reduction
- ✓ Quality of problems and issues found
- ✓ Reliability and confidence as program progressed

Performance and Scalability

One of the key lessons learned in the beta program was the need to plan on additional real storage. A 10 percent to 30 percent increase of real memory is a very rough estimate. For small systems with tiny buffer pools, the increase will be toward the high end of the range; for big systems with large buffer pools, it will be toward the low end of the range. It is important for customers to properly provision and monitor real storage consumption.

Many traditional OLTP workloads saw a 5 percent to 10 percent reduction in CPU utilization in CM mode after REBIND under DB2 10 for z/OS (some more, some less). On the initial migration to DB2 10 for z/OS, most customers will not perform a mass REBIND of all plans and packages. So, before REBINDing plans and packages, you may see little or no reduction in CPU resource consumption.

To maximize the price/performance benefits after migrating to CM, take these two steps:

1. REBIND your packages and plans to generate the new 64-bit SQL runtime. This way, you avoid the overhead of making the runtime for migrated packages from earlier releases look like the DB2 10 for z/OS runtime and re-enable fast column (SPROC) processing, which would otherwise be disabled.
2. Take advantage of 1 MB size real storage page frames to reduce translation lookaside buffer (TLB) misses. The 1 MB size real storage page frames are available on the z10™ and z196 processors. The prerequisite for using them is to specify the long-term page fix option for your local buffer pools. Long-term page fix buffer pools, which were introduced in DB2 8, provide an opportunity to reduce CPU resource consumption by avoiding the repetitive cost of page fix and page free operations for each page involved in an I/O operation.

The lesson is, be sure to use PGFIX=YES on your local buffer pools, provided there is sufficient real storage provisioned to fully back the requirement of the total DB2 working set below and above the 2 GB bar.

In a few cases, customers saw less than 5 percent saving in CPU resource consumption for traditional OLTP with very light transactions—"skinny" packages with a few simple SQL statements. This result is due partly to the increasing cost of package allocation, which overrides the benefit of the SQL runtime optimizations. APAR PM31614 may solve this issue by improving

package allocation performance. Another way to address this is to use persistent threads with the BIND option RELEASE(DEALLOCATE) to amortize away the repetitive cost of package allocation/deallocation per transaction.

Regarding customers' measurements, keep in mind that—unlike the DB2 Lab environment, where a dedicated environment is used—customer measurements are typically performed in a shared environment, and the measurement results are not always consistent and repeatable. There can be wide variation on measurement "noise" in customer measurements, especially regarding elapsed time performance.

In most cases, customers were *not* running in a dedicated environment or at the scale/size of true business production. Many customers ran a subset (maybe a high-volume subset) of the total production workload. Sometimes, they used a synthetic test workload to study specific enhancements.

In cases where customers had very large numbers that they were not able to reproduce, the numbers on CPU and elapsed time reductions were not trusted.

Recommendation

Customers should not spend anticipated price/performance (cost reduction) savings until they actually see the improvements in their own true business production environment.

Early results

Table 1.1 summarizes some of the beta program results reported by customers. Some of the additional savings were due to features such as using 1 MB size real storage page frames for selective buffer pools, enabling high-performance DBATs, and respecting the package BIND option RELEASE(DEALLOCATE). Another reason was the improvement in COMMIT processing for applications that commit frequently. We now perform parallel writes to the active log dataset pair even when rewriting a log control interval (CI) that was partially filed and written out previously.

Workload	*Customer results*
CICS online transactions	Approximately 7% CPU reduction in DB2 10 CM after REBIND; additional reduction when 1 MB size real storage page frames were used for selective buffer pools
CICS online transactions	Approximately 10% CPU reduction from DB2 9
CICS online transactions	Approximately 5% CPU reduction from DB2 8
CICS online transactions	10+% CPU increase
Distributed concurrent insert	50% DB2 elapsed time reduction; 15% chargeable CPU reduction after enabling high-performance DBAT
Data sharing heavy concurrent insert	38% CPU reduction
Queries	Average CPU reduction 28% from V8 to DB2 10 NFM
Batch	Overall 20–25% CPU reduction after rebind packages

Table 1.1: Workload results reported by DB2 10 for z/OS beta program customers

Now, let us discuss the use of the 1 MB size real storage page frames on the z10 and z196 processors. The potential exists for reduced CPU resource consumption through fewer TLB misses; however, the local buffer pools must be defined as long-term, page-fixed (PGFIX=YES). This feature was introduced in DB2 8 to mitigate CPU regression and reduce CPU resource consumption for I/O-intensive buffer pools.

Many customers are still reluctant to use the PGFIX=YES option because they are running too close to the edge on the usage of the amount of real storage provisioned and are in danger of paging to auxiliary (DASD) storage. They understand the value of PGFIX=YES, but it applies only for an hour or two each day. Another factor is that this decision is a long-term one; in most cases, implementing this buffer pool attribute requires a recycle of the DB2 subsystem. A change to the attribute goes pending and is materialized when the buffer pool goes through reallocation. It is also worth noting that a 75 percent cost reduction on real storage is incurred on the z196 processor relative to the z10 processor.

Here are a few more things to remember about the use of 1 MB size real storage page frames on the z10 and z196 processors: The actual amount of memory that is allocated as 1 MB size real storage page frames is specified

by the LFAREA=nn% parameter in the IEASYSnn parmlib member and is changeable only by IPL. You are partitioning out the total real storage provisioned between 4 K size frames and 1 MB size frames. 1 MB size real storage page frames are non-pageable. If these page frames are overcommitted, DB2 10 for z/OS will start using 4 K size real storage page frames.

Recommendation

Assuming you have provisioned sufficient real storage in production to fully back the total requirement of the DB2 working set:

1. Define all the local buffer pools as long-term page fixed (PGFIX=YES),
2. Sum up the total buffer pool storage requirement across all the local buffer pools defined as PGFIX=YES, and
3. Reflect that value in the LFAREA specification. (You may want to add an additional 10 percent to 20 percent in size to allow for some growth and tuning.)

Note

Make sure you have applied critical preventative z/OS maintenance *before* using 1 MB size real storage page frames. One of the lessons learned in the beta program is that the 1 MB size real storage page frames are relatively new and DB2 10 for z/OS is the first major subsystem to exploit them. We observed a reduction of up to 6 percent in CPU resource consumption. There is a customer requirement for a new parameter to be able to use PGFIX=YES independently from the use of 1 MB size real storage page frames. This requirement will be addressed in a future release of DB2 for z/OS.

DBM1 virtual storage constraint relief (VSCR) with a near-full 64-bit SQL runtime is available for use as soon as you go to CM. To accrue maximum benefit, you must REBIND static SQL plans and packages. We are confident that we have addressed the previous vertical scalability issue on the limited number of active threads that could be supported, and we have achieved very good results.

This support offers a "real-world" proposition of scaling up the number of active threads from, say, 500 active threads to 2,500–3,000 active threads or more per DB2 subsystem. The limiting factors now on vertical scalability (number of threads times average thread storage footprint) are most likely to be the amount of real storage provisioned on the logical partition (LPAR), followed by extended system queue area/extended common service area (ESQA/ECSA) (31-bit) storage constraints and the active log write performance (log latch contention).

Figure 1.1 shows three sets of customer measurements.

The first measurement (shown in the left column of the figure) is the virtual storage footprint of DB2 9 for z/OS.

The middle column shows the virtual storage footprint of DB2 10 for z/OS in CM without the REBIND of static SQL plans and packages. The issue here is that the footprint actually increased, compared with DB2 9 for z/OS. This issue was corrected ahead of GA of DB2 10 for z/OS.

The third column shows that once you do the REBIND of static SQL plans and packages, the 31-bit virtual thread storage footprint decreases dramatically. This result illustrates the value of the DBM1 31-bit virtual storage constraint relief in DB2 10 for z/OS.

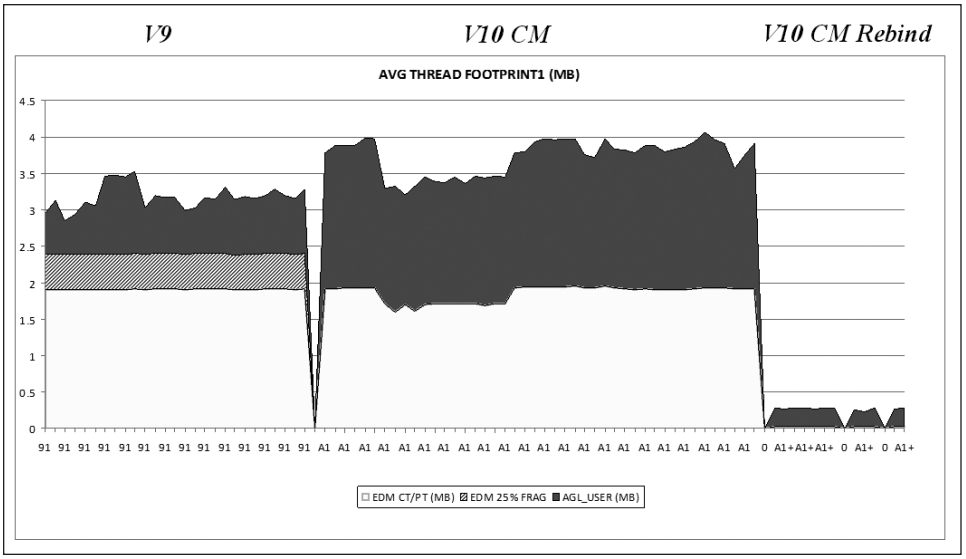

Figure 1.1: Initial DBM1 31-bit thread storage customer measurements in DB2 9 for z/OS vs. DB2 10 for z/OS (corrected prior to General Availability [GA])

Figure 1.2 shows another group of customer measurements.

Here, the first column is the DB2 9 for z/OS thread footprint. The second column is the DB2 10 for z/OS CM without the REBIND of static SQL plans and packages. In columns three and four, you can see that after the fix is applied (even without the REBIND), the thread storage footprint is greatly reduced.

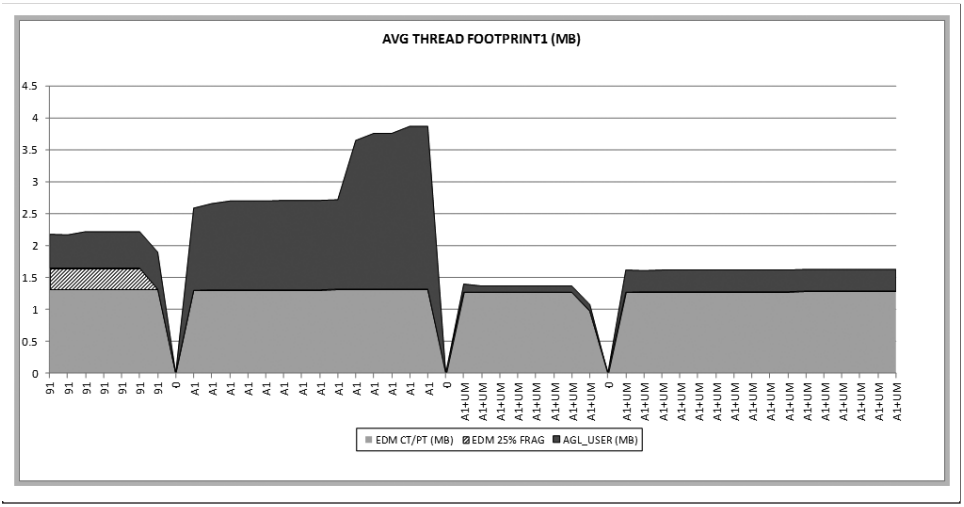

Figure 1.2: Initial DBM1 31-bit thread storage customer measurements in DB2 9 for z/OS vs. DB2 10 for z/OS [GA after fix applied])

With or without the REBIND of static SQL plans and packages, the 31-bit thread storage footprint in the DBM1 address space is reduced in DB2 10 for z/OS. However, to accrue maximum benefit in terms of 31-bit VSCR in the DBM1 address space, rebinding static SQL plans and packages is strongly recommended.

DBM1 virtual storage constraint relief with 64-bit SQL runtime

REBINDing static plans and packages maximizes the DBM1 31-bit VSCR and ensures we have a 64-bit SQL runtime. Not only does this step solve scalability issues, but it also can provide opportunities for further price/performance improvements—beyond the 5 percent to 10 percent.

For example, prior to DB2 10 for z/OS, many customers have been heavily constrained on available, 31-bit virtual storage in the DBM1 address space and, as a result, on the number of active threads that can be supported in a single DB2 subsystem or DB2 member. They have had to make compromises, trading additional CPU resource consumption to reduce the 31-bit virtual storage footprint and be able to support more active threads in a single DB2 subsystem or DB2 member.

This tradeoff involved reducing the number of persistent threads and restricting the use of the BIND option RELEASE(DEALLOCATE) for packages running

on those threads. These tactics saved on DBM1 31-bit virtual storage resource consumption at the cost of incurring additional CPU resource consumption.

With DB2 10 for z/OS, provided you have additional storage provisioned over and above the 10 percent to 30 percent previously mentioned, you can use more persistent threads and make more use of the BIND option RELEASE(DEALLOCATE) with existing or new persistent threads. This capability has the potential to reduce CPU resource consumption and improve price/performance (cost reduction) beyond the previously mentioned 5 percent to 10 percent. However, it does require additional real storage to be provisioned to support the increased number of persistent threads running with RELEASE(DEALLOCATE). This is in addition to the 10 percent to 30 percent increase in real storage requirement discussed previously.

The next, and new, opportunity for price/performance improvement is with regard to Distributed Relational Database Architecture™ (DRDA) and DDF server workloads. In DB2 10 for z/OS, starting with CM there is the potential to reduce CPU resource consumption for DRDA transactions by using high-performance database access threads. DB2 10 for z/OS provides the same opportunity for thread reuse with persistent threads that we have, for example, in CICS with protected ENTRY threads and/or by queuing on an unprotected ENTRY thread.

To take advantage of this improvement, the first prerequisite is that at least one of the packages associated with the transaction must be bound with RELEASE(DEALLOCATE). The second prerequisite is to issue the MODIFY DDF PKGREL(BNDOPT) command so that the BIND option RELEASE(COMMIT|DEALLOCATE) is respected.

After taking these steps, you will be able to achieve thread reuse for the same connection. At the same time, DDF will start respecting the BIND option of RELEASE(DEALLOCATE). Before DB2 10 for z/OS, you could BIND distributed packages with the RELEASE(DEALLOCATE) attribute, but the availability of this option was a moot point because RELEASE(COMMIT) was always forced at execution time (i.e., the BIND option of RELEASE(DEALLOCATE) was not respected).

Now, in DB2 10 for z/OS, we have the same possibility as with CICS and IMS/TM workloads—to have persistent threads, in this case with high-performance DBATs, and to have the BIND option of RELEASE(DEALLOCATE) respected.

The recommendation is that once you plan to start using high-performance DBATs, consider provisioning additional real storage—beyond the previously discussed 10 percent to 30 percent increase. *Do not* adopt a "one size fits all" strategy when using more persistent threads with the BIND option RELEASE(DEALLOCATE) with IMS/TM, CICS, or DDF workloads. Most installations cannot support making all threads persistent threads, with all the associated packages bound with RELEASE(DEALLOCATE), because of the potential for dramatic increase in the total real storage requirement.

You simply cannot afford to use the BIND option RELEASE(DEALLOCATE) for *all* plans and packages. Target persistent threads for thread reuse at high-volume simple transactions, and couple them with use of RELEASE(DEALLOCATE) for high-use packages with many SQL statements that are frequently executed.

For example, take your Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) packages as used by distributed client applications and BIND them twice—into two different package collections: BIND them with RELEASE(DEALLOCATE) in one collection, and BIND them with RELEASE(COMMIT) in the other collection.

In this way, you can target the high-volume transactions that would benefit the most from the use of persistent threads with BIND option RELEASE(DEALLOCATE) and connect those transactions to a data source that points to the collection where the packages are bound with RELEASE(DEALLOCATE). Packages must be bound with RELEASE(DEALLOCATE) to be eligible to use high-performance DBATs and be reused for the same connection. The remaining transactions would connect to a data source that points to the collection where the packages are bound with RELEASE(COMMIT).

The story is similar with CICS and IMS/TM. For CICS, you would choose only protected ENTRY threads for high-volume transactions and couple that with the use of BIND option RELEASE(DEALLOCATE) for frequently executed packages. Allow the rest of the transactions to run as POOL threads.

For DRDA workloads, do not overuse BIND option RELEASE(DEALLOCATE) on packages, because it will drive up the MAXDBAT requirement.

Another point to remember is that when you use persistent threads with RELEASE(DEALLOCATE), there is a tradeoff. Doing so will impact BIND/REBIND and SQL DDL concurrency. When you have a high-volume transaction that justifies use of persistent threads with RELEASE(DEALLOCATE), then BIND/REBIND and DDL activity cannot break in.

Many customers fail to see the benefit of thread reuse and avoiding the repetitive cost of thread create and thread terminate per transaction. Here is the explanation as it relates to CICS: If you are incurring the overhead of thread create and thread terminate, you cannot see the overhead in the DB2 accounting record. On the other hand, if you avoid the overhead of thread create and thread terminate, you also cannot see the overhead saved in the DB2 accounting record.

CICS uses the L8 TCB to process DB2 work, regardless of whether the application is thread safe or not. The CPU time associated with thread create and terminate (or the avoidance thereof) shows up in the CICS System Management Facilities (SMF) Record Type 110 record. Note that before the introduction of the Open Transaction Environment (OTE) in CICS, CICS did not even capture the cost of thread create and terminate in the SMF Record Type 110 record. The CPU cost of thread create and terminate was not captured. Provided successful thread reuse is achieved, the benefit of using BIND option RELEASE(DEALLOCATE) will show up in a reduction in the Class 2 TCB Time in the DB2 Accounting Record (SMF Record Type 101).

For some customer installations, DB2 10 for z/OS also has the potential to reduce the number of DB2 members in a data sharing group. Some customers had to grow their DB2 processing capacity horizontally due to the 31-bit virtual storage constraint in the DBM1 address space by growing the width of the data sharing group by adding additional DB2 members. Some of these same customers decided to run multiple members from the same DB2 sharing group on the same LPAR.

Why? They wanted to limit the number of LPARs running on the faster z10 and z196 systems because of increasing LPAR overheads. But they needed to keep the existing DB2 members, or even add DB2 members, to have enough thread processing capacity.

Now, with the generous DBM1 31-bit virtual storage constraint relief in DB2 10 for z/OS, such customers have the ability to reduce the total number of DB2 members in a data sharing group. This change can reduce the number of DB2 members from the same data sharing group running on the same LPAR down to one, and can possibly reduce the total number of LPARs as well. The ability to reduce the total number of DB2 members and/or the number of LPARs will provide further price/performance (cost reduction) improvements.

Before you consolidate DB2 members and LPARs, there are some issues to consider. For example, what will happen to the logging rate when you push more

workload through a single DB2 subsystem? And, can the size of the active log configuration, the dataset placement, and the I/O subsystem cope with the load? Will log latch contention be aggravated?

By running more workload through an individual DB2 subsystem, you will drive up the aggregate logging rate for that DB2 subsystem. Also, you need to consider the increase in SMF data volume per LPAR. In DB2 10 for z/OS, you can now enable DB2 compression of instrumentation record data written to SMF (e.g., DB2 accounting trace data) to reduce the SMF data volume. DB2 instrumentation data, such as statistics trace and accounting trace records, are typically written out to SMF and can benefit from this enhancement.

A new DB2 system parameter (ZPARM) called SMSCOMP, once enabled, turns on DB2 compression of the SMF output records. This compression applies to any instrumentation record, not just statistics and accounting, that is written out to SMF. We have observed a 70 percent to 80 percent reduction in the volume of SMF data when the DB2 compression is turned on. The CPU overhead incurred is only about 1 percent— representing a very good tradeoff.

This enhancement provides an opportunity for improved problem determination (PD) and problem source identification (PSI) by offering the possibility of turning off the use of accounting roll-up for DDF and Recovery Resource Services attachment facility (RRSAF) workloads (default). We introduced this support in DB2 8 to reduce SMF data volume, but one of the drawbacks of accounting roll-up was that it compromised performance PD/PSI.

By rolling up the transaction activity for multiple transactions into a single accounting record, we lose information about outlying, badly performing transactions. The information about the poor performance of outlying transactions gets "amortized" away by the accounting roll-up. Given the introduction of SMF data compression in DB2 10 for z/OS, SMF compression may be a better option to control SMF data volume than using the accounting roll-up.

Another consideration when migrating to any new DB2 for z/OS release is the impact of increased dump size due to growth in the total DB2 working set size (and the need to avoid partial dump capture). Make sure sufficient real storage is provisioned on the LPAR for the increased DUMPSRV and MAXSPACE requirement.

Finally, we want to re-emphasize the continued business and technical value of DB2 data sharing to differentiate the z/OS platform in terms of providing continuous availability across *both planned and unplanned* outages. You want to

avoid large single points of failure. For example, consider a minimum configuration of four-way data sharing for true, continuous availability, assuming a two-processor (CEC) configuration.

By "four-way data sharing," we mean when you have two boxes (CECs) and two LPARs on each box (a total of four LPARs). A single DB2 member would run on each LPAR. That is the minimum recommendation for true, continuous availability and to maintain performance, if you want to maintain your service level agreement (SLA).

In this four-way configuration, if you were to lose a DB2 member on one LPAR, the surviving DB2 member on the alternate LPAR on the same box can take on 100 percent of the workload and use all the CPU processing capacity available on the box.

Planning for real storage

Let us discuss now, in more detail, the need to carefully plan, provision, and monitor real storage consumption. Most DB2 8 and DB2 9 for z/OS customers are properly configured and provisioned in terms of real memory. However, some are running so low on available real memory that part of the DB2 working set is often being paged out, intermittently, to auxiliary (DASD) storage.

Worse still, if a dump were to be taken on the system, the dump capture would take several minutes instead of a few seconds to complete, and it could spread "sympathy sickness" around a data sharing group.

Information about real and auxiliary frames used is already recorded in the IFCID 225 record generated by DB2 for z/OS. However, although the provided information has been improved, with more details recorded in DB2 10 for z/OS, the information furnished in IFCID 225 has not allowed a customer installation to effectively monitor 64-bit shared and 64-bit common storage when running multiple DB2 subsystems on the same LPAR.

A new DB2 APAR PM24723 for DB2 10 for z/OS provides the needed capability. The new APAR uses the enhanced capability provided with MVS APAR OA35885, which provides a new callable service to Real Storage Manager (RSM) to report REAL and AUX usage for a given addressing range for shared objects. APAR PM24723 will have this new MVS APAR as a prerequisite.

The other advantage is that this same DB2 APAR provides a much-needed real storage management function within DB2 when available real storage is overcommitted and the system starts to be paged out.

Some customers have used a hidden system parameter (ZPARM) called SPRMRSMX (real storage "kill switch") when running multiple DB2 subsystems on the same LPAR. SPRMRSMX protects individual DB2 subsystems and other subsystems running on the LPAR such that if one of the DB2 subsystems were to "run away" in terms of virtual memory use, that subsystem would be "sacrificed" so that the other DB2 subsystems could continue to run.

Customers using system parameter SPRMRSMX have calculated the "normal" working set of a DB2 subsystem, multiplied that value by 2 (as a contingency), and used the resulting value as the SPRMRSMX setting. Customers currently using this system parameter will need to carefully re-evaluate the value set when migrating to DB2 10 for z/OS.

In DB2 10 for z/OS, you will need to factor in the increased use of 64-bit shared and common storage to establish the new DB2 for z/OS storage footprint. IPL amounts for the LPAR will need to be adjusted based on the number of DB2 members running on that LPAR. The following values are on a "per DB2 subsystem" (i.e., you would double these numbers when running two DB2 subsystems on an LPAR, triple them for three, and so on):

Storage area	IPL amount
64-bit private	1 TB
64-bit shared	128 GB
64-bit common	6 GB

Note carefully that these values are not indicative of real memory to be used, or even of virtual memory to be allocated; they simply represent reserving an addressing range for DB2 for z/OS to use. These large memory object areas are allocated above the 2 GB bar, and they will be sparsely populated. Virtual memory is not allocated until the pieces of storage are actually referenced.

INSERT performance

INSERT is one of the most important SQL statements in DB2 for z/OS. It is also one of the most challenging for any database management system (DBMS) to handle. Previous DB2 for z/OS releases have focused on improving INSERT performance. DB2 10 for z/OS provides some improvements for all table space types. There was particular focus on improving INSERT performance for universal table spaces, both partition by range (PBR) and partition by growth (PBG).

Over the longer term, what we want to do in DB2 for z/OS is converge all the classic table space types to be UTS and deprecate the old, classic table space types. DB2 10 for z/OS includes two specific enhancements to improve UTS performance. First, UTS now supports MEMBER CLUSTER to help where there is excessive page latch and page p-lock contention on space map pages and on data pages when using row-level locking. Second, changes were made to the space search algorithm, making the algorithm used by UTS now more like that used by the classic partitioned table space.

The performance goal for INSERT in DB2 10 for z/OS was for UTS to be equal to, or better than, the classic partitioned table space. While we are not there yet, the performance is dramatically improved. The improvement is very workload-dependent. There is still a tradeoff between space reuse versus throughput and reduced contention. We still have some work to do on UTS, in the area of both PBR/PBG with row-level locking and sequential insert activity.

Three specific improvements to INSERT in DB2 10 for z/OS should help all table space types. The first is reduced log record sequence number (LRSN) spin for inserts to the same index or data page. As processors become faster, such as z10 and z196, the possibility of duplicate LRSN values and spins having to occur increases. When a spin occurs, processing loops in the DB2 code wait for the LRSN value to change. The LRSN value is used in data sharing to serialize restart/recovery actions, and it is the high-order six bytes of the store clock (STCK) value. The LRSN is incremented every 16 microseconds. As processors get faster, there is increased potential for LRSN duplicate values and the need to spin. We already made some improvements in DB2 8 and DB2 9 for z/OS regarding this issue.

In DB2 10 for z/OS, when we have multi-row inserts (MRI) or single inserts within an application loop, we avoid the LRSN spins for the same page that would have occurred previously. The results have been very impressive. This improvement applies when you use multi-row inserts to the same page or have INSERT within an application loop to the same page, in a data sharing environment.

The second improvement, which works very well, is an optimization for "pocket" sequential insert activity. This is where you have multiple "hot spots" in the key range and the INSERTs are "piling in" on these hot spots. During insert, DB2 Index Manager (IM) identifies to the DB2 Data Manager the candidate row ID (RID) value (page) to be used to place the new data row. DB2 Index Manager

now returns the next lowest key RID value. The end result achieved is a much better chance to find the space and avoid a space search.

The third improvement relates to parallel index read I/O, which works very well and is best-suited when it is activated where there random index key inserts. This mechanism is used when three or more indices exist on the table and you are performing random index key INSERTs. Previously, you would have had a lot of random sync read I/O. We now do parallel index read I/O when there are three or more indices on the table. This reduces the elapsed time by taking the synchronous read I/O activity out of the elapsed time.

To compensate elsewhere for the increase in CPU resource consumption, DB2 will now make the CPU resource consumption associated with prefetch engines (sequential prefetch, list prefetch, and limit prefetch) and deferred write engines eligible for zIIP offload. These types of processing are now offloaded to zIIP processors to compensate for the increase in CPU when doing parallel index read I/O for random key INSERTs.

Accounting Trace Class 3 enhancement

In DB2 10 for z/OS, there are now separate counters for IRLM Lock/Latch Wait and DB2 Latch Wait events in the DB2 accounting trace. Previously, both types of wait events were included in a single counter. When analyzing application performance problems, you had to try to figure out which type of wait activity was elevated.

The next improvement relates to data sharing. One of the disadvantages of having very large, local buffer pools with many group buffer pool (GBP) dependent objects, was that DB2 for z/OS used to scan the local buffer pool for each GBP-dependent object during DB2 shutdown. These scans added a lot of delay in shutting down the DB2 subsystem. DB2 also used to scan the local buffer pool when an object went into or out of GBP dependency. This activity could add a lot of overhead, depending on how often these transitions were made.

In DB2 10 for z/OS, we expect faster DB2 shutdown times because we avoid the local buffer pool scan per GBP-dependent object during the shutdown. We now also avoid the local buffer pool scan when an individual object (pageset/partition) transitions into or out of GBP dependency.

Inline LOB column values are now supported in DB2 10 for z/OS. The size of the inline portion can be specified as a system parameter (ZPARM) or on

an individual object basis. There is no "one size fits all" value for the use of inline LOBs. So, using a general value as a system parameter is unlikely to be a good choice.

You will get more value by setting the inline LOB value on the SQL DDL for the specific object. The performance tuning goal is to avoid access to the auxiliary table space for the majority of LOB column values. This function is aimed primarily at applications that have many, small LOB columns values (i.e., up to a few hundred bytes).

The design goal for the inline LOB value is to store the complete column value inline, in the base table row, and to avoid access altogether to the auxiliary table space. The potential exists for significant CPU and elapsed time improvement if this can be achieved by setting the right value for the inline portion.

However, if you store the LOB column value inline, in the base table row, and then very rarely reference the LOB column value, you may impact performance elsewhere because you will get fewer rows per page. In any event, you may need to consider increasing the data page size.

In the worst case, if you have made a poor choice for the inline LOB column value, you will have the first part of most LOB column values in the base table and the remaining part of the LOB column values in the auxiliary table space. So, not only will you get no benefit, but you will actually increase CPU overhead and waste DASD space. But another advantage to inline LOBs is that the portion of the LOB that is stored in the base table row is now eligible for data compression.

Another performance enhancement to DB2 10 for z/OS relates to active log writes. Before DB2 10 for z/OS, DB2 active log writes were always done serially to log copy 1 and log copy 2 when rewriting a previously written log CI that was partially filled previously. DB2 would write to log copy 1, wait, and then, when it was successful, write to log copy 2. The reason for this was that, prior to RAID devices, we had single, large, expensive disks (SLEDs). We were always concerned that, when we rewrote a previously partially filled log CI, we might destroy the previous version and its contents.

With the increased reliability provided by RAID devices, there is no longer any reason to do these rewrites of log CIs serially. DB2 10 for z/OS now always performs active log writes in parallel. This enhancement can generate significant elapsed time improvements and improvements in applications that commit frequently or when other forced writes occur (e.g., related to index leaf page splits).

Hash access vs. index-only access

Hash access basically "competes" with clustered index access, and specifically with index-only access. In an effort to reduce CPU resource consumption, hash access tries to avoid going through an index B-tree structure with many levels to access the data row. The advantage that clustered index access has is that DB2 still tries to maintain clustered data row access. Index-only access avoids access to the data row completely. DB2 10 for z/OS also provides the opportunity to have a unique index with INCLUDE columns.

Today, you may have multiple indices on a table. One index is there to enforce the uniqueness of the primary key. You may have added another index to improve performance. The leading columns may be the same in both indices. You may now include additional columns in a unique index and still use that same index as before to enforce the unique constraint.

Now, the advantage of a unique index with INCLUDE columns is that it gives you the ability to satisfy the unique constraint check *and* provide the performance benefits you want for query. The result is that you can reduce the number of indexes required for performance reasons. For every index you can avoid, you will improve the performance of INSERT and DELETE and possibly improve UPDATE performance, as well.

A number of customers evaluated both methods to try to find the "sweet spot." There is definite value from hash access, provided you can determine that sweet spot. However, in practice, the sweet spot has proved to be relatively small. Here are guidelines for identifying the sweet spot:

- High NLEVELS in index (more than two)
- Access by applications needs to be purely direct row access by primary key
- Truly random access
- Read-intensive, not volatile
- No range queries (minimize BETWEENs, >, <, and so on)
- Many rows per page

One of the key points about hash access performance is that you want to "tune" the space allocation of the fixed-sized hash area so that you reduce the number of rows that go into the overflow index (i.e., control overflow). If the primary fixed hash area is too small, you will have many rows in the overflow

index; on the other hand, if the primary area is too large, you will have too much random I/O.

To help with sizing the fixed hash area size, DB2 10 for z/OS provides a new option on the REORG utility called AUTOESTSPACE(YES). When you perform REORG with this option, it uses information from Real Time Statistics (RTS) to resize the primary fixed hash area and reduce the number of rows in the overflow index. However, even after such a REORG, there may still be some small number of data rows in the overflow index.

Finally, when you migrate to hash access, you will see some degradation in the elapsed time for both LOAD and REORG utility executions.

Availability

There are a number of enhancements in DB2 10 for z/OS to reduce planned outages for applications and to improve the success of the online REORG utility.

Online schema evolution

"Deferred Alter" is a new feature in DB2 10 for z/OS. With this mechanism, when you make a schema change, the change goes "pending" and it is stored in the DB2 Catalog. The next time you perform an online REORG, the online REORG will materialize the pending changes. You can set up many deferred alters. Each of the changes will go pending in the DB2 Catalog until the subsequent online REORG, when the changes will be materialized.

Why is this important? This mechanism now gives you a migration path away from the classic table space types of simple, segmented, and partitioned—which contain a single table—over to universal table spaces.

Note

UTS is a prerequisite for some of the DB2 10 for z/OS functions, such as hash access, inline LOB, and currently committed. It is also a prerequisite for the cloned table function in DB2 9 for z/OS. If a table space is a simple table space or a segmented table space, you can have only one table per table space to be able to use this migration path to UTS, because UTS supports only one table per table space.

Note

This migration path to UTS is a *"one-way ticket" only*. Once you migrate to UTS, you *cannot go back* using the same Deferred Alter mechanism to simple, segmented, or partitioned table spaces. To return to using the classic table space types, you would have to unload the data, drop the table space, redefine the table space as it was before, and reload the data.

Note also that point-in-time recovery to a point *before* a successful materializing online REORG is not possible. If, for example, you have incorrect results from REORG, possibly because the wrong rows were discarded or an application change needs to be rolled back, you cannot recover to a point before the online REORG.

Now, once you have migrated to UTS PBG/PBR, you can change attributes such as DSSIZE and index page size. You can turn MEMBER CLUSTER on and off or migrate to and from hash access. These abilities are all provided by the Deferred Alter mechanism, followed by the online REORG. This function works very well and can help reduce the number of destructive database changes that previously caused database downtime.

To summarize, the benefits of Deferred Alter are:

- Streamlining the move to UTS
- Reducing the administrative time and cost associated with moving to UTS
- Helping minimize errors
- Reducing outages

Another new option is the FORCE option of online REORG. In the last part of the REORG, when you are in the final attempt to drain the object and are about to make the switch, if there are "active" threads blocking, the FORCE option allows DB2 10 for z/OS to kill the active threads.

Early beta customers found limited value to this function because if the threads were active in DB2, DB2 would cancel the threads (good). But if the threads were inactive, the FORCE function did not kill them, and the online REORG failed. Then, when the inactive threads came back to life after the online REORG failed, the threads were canceled on their way back in. So the FORCE option is not a

guaranteed way to kill all blocking threads and allow the online REORG to always make the switch.

Also new with DB2 10 for z/OS, the online REORG of LOB table spaces provides a DISCARD option. Early customers thought this feature was of limited value because it cannot handle LOB column values greater than 32 K.

Other Issues

First, there is the ability to create classic partitioned table spaces (PTS). In DB2 10 for z/OS, the classic PTS is now deprecated, meaning that, by default, you will not be able to create any new PTS. An attempt will be made to honor the request by creating a UTS PBR. However, a CREATE of UTS will support only the table-based controlled partitioning syntax. The legacy index-based control partitioning syntax is not supported for UTS.

So, by default, you may not be able to create any new, classic PTS. However, customers demanded the continued ability to create classic PTS because there are still a few areas where classic PTS has value over UTS.

The good news is that you can still create classic PTS in DB2 10 for z/OS, and these table spaces are still officially supported. There are two ways to continue to create classic PTS:

1. Specify SEGSIZE=0 on the CREATE TABLESPACE statement.
2. Set new system parameter (ZPARM) DPSEGSZ to zero (the default is 32).

Either of these methods will let you create classic PTS in DB2 10 for z/OS.

For customers who still have old COBOL and PL/1 programs, the DB2 7 lookalike precompiler (DSNHPC7) for COBOL and PL/I is *still* provided in DB2 10 for z/OS.

The concurrency issues with parallel SQL DDL execution are not absolutely solved in DB2 10 for z/OS, despite the DB2 Catalog restructure. While the restructure was eventually successful for parallel BIND/REBIND activity, most customers still experience deadlocks when running parallel jobs with heavy SQL DDL against different databases within the same commit scope. Therefore, some customers will still have to run their SQL DDL jobs single-threaded.

BIND/REBIND issues

With single-thread BIND/REBIND, early customers have reported degraded CPU and elapsed time performance on entry into DB2 10 for z/OS CM. There are two reasons for this experience:

- PLANMGMT is now ON by default, and its default value is EXTENDED.
- New indexes defined for post–Enable New Function Mode (ENFM) processing, when hash links are eliminated, are being used even in CM.

Because we have a single code path (no dual path processing) across the different modes of DB2 10 for z/OS, those indices are now used even in Conversion Mode. For most customers, single-thread BIND/REBIND performance remains important because there are no concurrency improvements until after the DB2 Catalog restructure is completed at the end of ENFM.

With parallel BIND/REBIND jobs, particularly in data sharing mode, we identified and addressed a number of concurrency and performance problems prior to general availability, including:

- Performance problems related to the repetitive DELETE/INSERT process
- Space growth in SPT01 for both LOB table spaces and base table spaces

The concurrency of parallel BIND/REBIND jobs is now working well. There are several relevant APARs:

APAR	Description
PM24721	Inefficient space search for out-of-line LOB in data sharing
PM27073	Inline LOB with compression for SPT01 to address SPT01 growth
PM27973	More efficient space reuse for base table and UTS

With these APARs applied, concurrent BIND/REBIND activity in data sharing mode works well after you get past ENFM processing.

Once beyond ENFM processing, we recommend that customers change existing procedures to run BIND/REBIND activity in parallel (but you should not do this until after ENFM). Doing so gives customer installations the opportunity to get back to and improve upon the elapsed time performance levels experienced in DB2 8 and DB2 9 for z/OS and to reduce application downtime when implementing new enterprise application releases.

Incompatible Changes

The most important incompatibility relates to the CHAR() scalar function. As an application programmer, you may want to use this function and apply it against a decimal column value to pull out a numeric value to assign to particular fields.

The incompatible change is documented in the install guide. The challenge for customers is how to identify what the rogue applications are that need be corrected. How do you identify what the exposure is? How can you support a phased migration?

By working with customers in the beta program, we were able to identify the issue. APAR PM29124 was created to restore the compatible behavior of pre–DB2 10 for z/OS, by default, for the CHAR() scalar function. In a subsequent APAR, we will give you the capability to put on a new trace that will identify those applications that are potentially exposed and require investigation. You will then be able, at the individual package BIND level, to indicate whether you want the new behavior.

The next incompatibility issue is with SQL stored procedures. If you have a native SQL procedure that was implemented and/or regenerated under DB2 10 for z/OS and you need to fall back to DB2 9 for z/OS, that native SQL procedure will not run. The workaround is to run ALTER PROCEDURE REGENERATE on the DB2 9 for z/OS member. APAR PM13525 will deal with this issue automatically for you.

Finally, there is an issue with Create Trigger for triggers that are created on DB2 10 for z/OS. If you fall back to DB2 9 for z/OS, such triggers will not work. The workaround is to drop and re-create these triggers under DB2 9 for z/OS after fallback.

Migration and Planning Considerations

This section reviews key migration and planning considerations to take note of in planning for DB 10 for z/OS.

Migration strategy

As in previous releases, we recommend a short time for mixed-release coexistence in data sharing. A short period for ENFM is also highly recommended. Support from vendors may affect the migration staging. One concern for CM is that some new performance improvements cannot be used.

The timing for moving from Test to QA to Production involves more options to consider. There are better controls for preventing the use of new functions, but a long gap between Test and Production levels is not advisable. You now have more granularity in the migration process and can move through mode by mode. Some customers migrate both Test and Production to CM and then change to New Function Mode (NFM) in a short time.

The chart shown in Figure 1.3 summarizes the history of DB2 releases. The top line tracks the year when each release became generally available (GA). The arrows show that the only releases where it was possible to skip a release were from DB2 5 to DB2 7 and from DB2 8 to DB2 10 for z/OS.

The lower part of the chart indicates the steps within the upgrade path from DB2 8 or DB2 9 for z/OS to DB2 10 for z/OS. The double-headed arrows indicate where you can "go back" a step, if necessary.

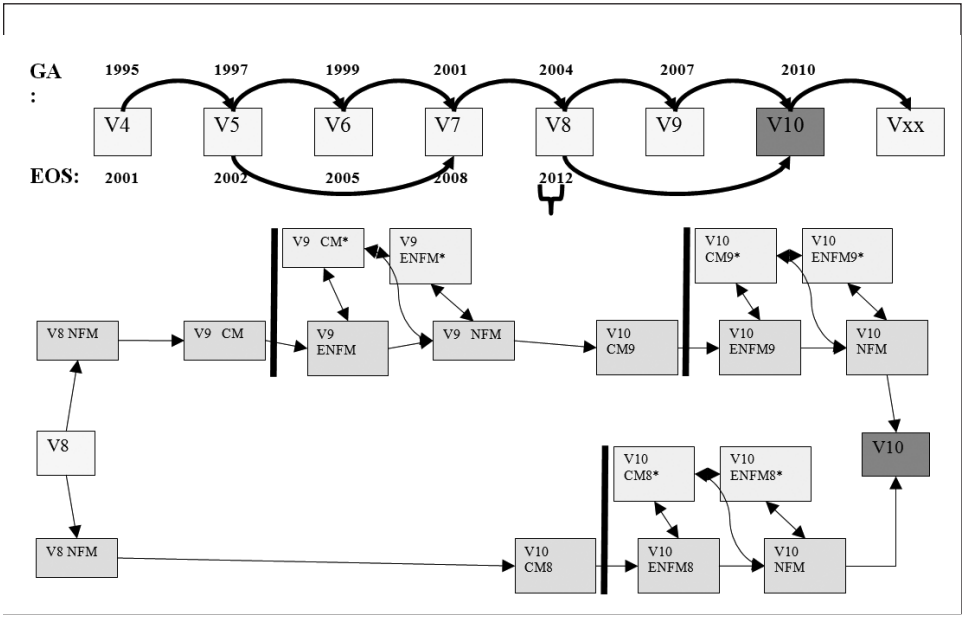

Figure 1.3: Timeline of DB2 releases and upgrade paths.

Note

If you are migrating from DB2 8, you have a decision to make. Should you go to DB2 9 for z/OS, or skip it and go directly to DB2 10 for z/OS? Once you decide to migrate to DB2 10 for z/OS CM8, you can still return to DB2 8. But you *cannot* then try to migrate to DB2 9 for z/OS CM.

Planning considerations

In general, the DB2 10 for z/OS migration process is very similar to that for both DB2 8 and DB2 9 for z/OS. It works well, with few customers experiencing problems with migration fallback. The ENFM process in DB2 10 for z/OS runs a lot longer than it did for DB2 9 for z/OS and even longer than it was on DB2 8.

You can migrate to DB2 10 for z/OS CM from either DB2 8 for z/OS NFM or DB2 9 for z/OS NFM. You cannot migrate through either of the following two scenarios:

1. Once you migrate forward from V8 NFM to DB2 10 for z/OS CM8, you can always fall back to V8 NFM, but you *cannot* then migrate forward to DB2 9 for z/OS CM.
2. Once you migrate forward from V8 NFM to DB2 9 for z/OS CM, you can always fall back to V8 NFM, but you *cannot* then migrate forward to DB2 10 for z/OS CM8.

Here are some important APARs to remember:

- Fallback Toleration SPE:
  - APAR PK56922
- Early Code for DB2 V8/V9:
  - APAR PK87280 (supersedes APAR PK61766)
- Information APARs:
  - II14474: V8 to V10
  - II14477: V9 to V10

If you are migrating from DB2 8 NFM, the bootstrap data set (BSDS) must be reformatted for the larger number of active/archive log tracking.

For those who operate DB2 Connect™, the minimum level supported is DB2 9.1 FP1. DB2 9.7 FP3A is required to support the new DB2 for z/OS functions.

Many customers still use DDF Private Protocol under DB2 8 and DB2 9 for z/OS. There is *zero* tolerance for DDF Private Protocol in DB2 10 for z/OS. You

must absolutely eliminate all use of DDF Private Protocol before starting DB2 10 for z/OS in CM.

Many customers have local plans and packages (CICS, IMS™, batch, and so on) that have been accidentally mistagged as requiring the use of DDF Private Protocol. These plans and packages, which have been mistagged, will be tolerated. However, if any of these packages *really* do perform an external call that uses DDF Private Protocol, the call will be prevented and the application will fail immediately.

In DB2 10 for z/OS, database request modules (DBRMs) bound directly into plans are no longer supported. However, if any DBRMs bound into plans are found at execution time, DB2 will automatically trigger AUTOBIND to generate packages on first allocation after entry into DB2 10 for z/OS. We choose a standard collection name to put these packages in, but the recommended best practice is to deal with DBRMs bound directly into plans *before* migrating to DB2 10 for z/OS. Any old plans and packages bound prior to DB2 V6 will also be invalidated and go through an AUTOBIND.

During ENFM processing on DB2 10 for z/OS, all the new indexes and new table spaces in the DB2 Catalog and Directory will be created as SMS-controlled, requiring extended addressability (EA) and extended format (EF) attributes. Some customers still do not use SMS management for the DB2 Catalog and Directory. Once you get to the ENFM process in DB2 10 for z/OS, the datasets of the DB2 Catalog and Directory must be SMS-managed.

For those of you coming from DB2 8, partitioned data sets extended (PDSEs)—as opposed to partitioned data sets (PDSs)—are required for SDSNLOAD, SDSNLOD2, and ADSNLOAD libraries.

The environment created by the DSNTIJSS job is *only* for DB2 Catalog and Directory data sets, which *must* be SMS-controlled in DB2 10 for z/OS. Other DB2 subsystem data sets, such as logs and the BSDS, are *not* accounted for in this environment.

The DSNHDECP module supports the NEWFUN parameter with the following options: V10, V9, or V8. This provides a way of stopping both static and dynamic SQL applications from using new SQL functions.

Many customers have old EXPLAIN table formats. DB2 10 for z/OS brings some changes in this space. First, if you have any plan tables that use a format prior to DB2 8, they will not work with EXPLAIN in DB2 10 for z/OS. The format and the ASCII/EBCDIC Coded Character Set Identifier (CCSID) from previous

releases are deprecated in DB2 10 for z/OS. They will fail with an SQLCODE –20008. If you have plan tables in DB2 8 or DB2 9 for z/OS format, you can still use them, but they will generate a warning SQLCODE +20520, regardless of whether they are CCSID EBCDIC or UNICODE.

If you use the DB2 10 for z/OS format, you must use UNICODE as the CCSID. If you try to use CCSID EBCDIC with DB2 10 for z/OS format, you will get the following errors:

1. EXPLAIN fails with RC=8 DSNT408I SQLCODE = -878.
2. BIND with EXPLAIN fails with RC=8 DSNX200I.

We recommend using the DB2 10 for z/OS extended format of the plan tables with a CCSID value of UNICODE. APAR PK85068 can help you migrate existing plan tables in DB2 8 or DB2 9 for z/OS table format over to the new DB2 10 for z/OS format with a CCSID of UNICODE.

Should you "skip" DB2 9?

If you decide to migrate from DB2 8 directly to DB2 10 for z/OS, you are, by definition, an early adopter of the new DB2 10 for z/OS release. This is because the end of support for DB2 8 is the end of April 2012. Quite clearly, the DB2 8 to DB2 9 for z/OS migration is the safer path to take because DB2 9 for z/OS has been in the field for almost four years and is quite stable.

Early customer adopters of DB2 10 for z/OS, whether migrating from DB2 8 or DB2 9 for z/OS, should expand their plans and take extra care to mitigate the risk of instability. This is not a statement of, nor an implication that, the DB2 10 for z/OS release has any endemic problems of instability. These same recommendations would apply to any release of DB2 or any other major software product.

First, you should perform application regression and stress testing to keep problems away from production. Next, plan to be proactive with regard to the continual application of preventive service maintenance. Plan to stay more current than two full, major preventive service maintenance drops per year. Regular, full, major preventive service maintenance drops, including HIPERs/PEs, are essential and required for about a year.

We strongly recommend planning for *four major* preventive service maintenance drops in the first year, based on the quarterly RSU. Then, you can move to two major and two minor preventive service maintenance drops

as the release passes through the early adopter curve. In between these drops, be vigilant and take advantage of the Enhanced HOLDDATA on a regular basis to find out which critical HIPERs/PEs are available.

One of the advantages of the CST/RSU process for recommended service maintenance, as opposed to the PUT route, is that it enables you to stay current on HIPERs/PEs that have gone through more testing but lets you stay further back on non-HIPERs/PEs maintenance. This capability provides some level of protection against PTFs in Error (PEs).

Finally, you have to be able to accept some level of risk and be able to handle some "bumps in the road" during the migration.

Security Considerations When Removing DDF Private Protocol

As previously mentioned, there is zero tolerance in DB2 10 for z/OS for DDF Private Protocol. Ahead of migrating to DB2 10 for z/OS, you need to plan for and work on eliminating all use of DDF Private Protocol and converting it to DRDA before you leave DB2 8 NFM or DB2 9 for z/OS NFM. There are fundamental differences in how authorization is performed, based on which distributed protocol you use and whether the protocols are used in combination.

Private Protocol is unique to the DB2 for z/OS requester and supports static SQL statements only. The plan owner must have authorization to execute all SQL requests executed on the DB2 for z/OS server. The plan owner is authenticated on the DB2 for z/OS requester and not at the DB2 for z/OS server.

Now, let us compare that with the DRDA Protocol. DRDA supports both static and dynamic SQL statements. The primary auth ID and associated secondary auth IDs *must* have authorization to execute both static SQL packages and dynamic SQL at the DB2 for z/OS server. The primary auth ID authenticated and secondary auth IDs are associated at the DB2 for z/OS server.

Until DB2 10 for z/OS, Private Protocol and DRDA protocols can be used by the same application within the same commit scope. You can "mix and match." Private Protocol security semantics are used due to possible inconsistent behavior, which is dependent on how the programs are coded and executed. That is a brief history of the differences between Private Protocol and DRDA Protocol.

Things have changed with APAR PM37300, which applies to DB2 8 and DB2 9 for z/OS. It provides control over the authorization checks performed when

migrating from Private Protocol to DRDA Protocol. In DB2 10 for z/OS, Private Protocol security semantics are no longer used because the default is to use DRDA Protocol for access from a DB2 for z/OS requester.

DB2 8 and DB2 9 for z/OS will now use DRDA authorization checks and will use the DB2 system parameter PRIVATE_PROTOCOL to determine what security checks should be performed. This system parameter was previously introduced for customers to prevent new use of Private Protocol after all the previous use was eliminated. To do this, a customer would set PRIVATE_PROTOCOL to NO.

So, before you disable Private Protocol by setting PRIVATE_PROTOCOL to NO, ensure that all the appropriate grants are in place by granting execute privileges to any user who plans to run a package or stored procedure package from a DB2 for z/OS requester at the DB2 for z/OS server. It will be treated like any other DRDA client application at the DB2 for z/OS server.

Clearly, this is a major change that could have a big impact. To help customers migrate to DRDA Protocol and the changes in security checking, both DB2 8 and DB2 9 for z/OS still provide the option to continue to prevent the introduction of new Private Protocol requests, but now provide the option to continue to use the Private Protocol authorization checks. This is achieved by changing the setting of the DB2 system parameter PRIVATE_PROTOCOL from NO to AUTH.

Save critical access paths and accounting data

BIND REPLACE and REBIND activity can cause unwanted access path changes. You should identify important queries, plans, and packages. Be sure that plan tables contain access paths and costs. ALTER current plan tables to add new DB2 10 for z/OS columns. REBIND may change access paths, so extract plans and run REBIND with EXPLAIN under a dummy collection or a different application or program name.

Keep accounting reports for crucial queries and applications. If you have a problem and send in accounting layout long reports and the plan table data, we will be able to troubleshoot the problems more *quickly*. If you *do not* have the reports and the data, then we must *guess*.

Items Planned for Post-GA Delivery

The first item to mention is APREUSE and APCOMPARE. These features are introduced with APAR PM25679. These options of BIND REPLACE and REBIND

provide a way to generate a new SQL runtime but, at the same time, ask DB2 10 for z/OS to give you the old access path wherever possible. So, if you have previously re-bound under DB2 9 for z/OS, this will mitigate the risk of access path change on the first BIND REPLACE or REBIND in DB2 10 for z/OS.

Additional items planned for post-GA delivery include the following:

- In DB2 10 for z/OS, you will be able to delete a data sharing member. This function was introduced by APAR PM31009. Deletion of a DB2 member will require a quiesce of the data sharing group.
- Inline LOBs will be introduced for SPT01 to gain the benefits of data compression and improve BIND/REBIND performance. This function is introduced with APAR PM27811.
- Enhancements for new DBA authorities are introduced with APAR PM28296:
  - Prevent privileged users from stopping audit traces
  - No implicit system privileges for DBADM
- Online REORG concurrency for materializing deferred ALTERs is introduced with APAR PM25648.
- Temporal enhancements:
  - TIMESTAMP WITH TIMEZONE support (APAR PM31314)
  - Enhancement for data replication (APAR PM31315)
  - ALTER ADD COLUMN, propagate to history table (APAR PM31313).
- New system profile filters based on "client info" fields is introduced with APAR PM28500:
  - Three new columns for userid, appname, and workstation
  - Wildcard support: if column is '*' then all threads pass that qualification.
- A new DB2 system parameter (ZPARM) to force deletion of coupling facility (CF) structures on group restart (APAR PM28925). This feature is aimed at disaster recovery. We want to avoid a situation during a disaster restart of using "stale" information in the CF structures. When the DB2 member starts and it is the first member to connect to the structure, it wipes out those structures and forces a group restart.
- Relief for the incompatible change in CHAR of decimal data by using APAR PM29124 to restore the previous behavior that existed prior to DB2 10 for z/OS.

- Real storage monitoring enhancements to be provided in APAR PM24723; this APAR also provides protection for over-commitment of available real storage.
- Hash LOAD performance (APAR PM31214)
- DSSIZE greater than 64GB (APAR yet to be announced).
- REORG REBALANCE SHRLEVEL CHANGE (APAR yet to be announced).

Note

RSM APAR OA35885 is a prerequisite to the enhanced storage monitoring capability provided by DB2 APAR PM24723. DB2 APAR PM24723 is strongly recommended for production use of DB2 10 for z/OS.

We strongly advise customers *not* to go into a major production environment *without* the proper monitoring of real and auxiliary storage usage as provided by this APAR and DB2 APAR PM24723. Together, these two APARs provide DB2 10 for z/OS with statistics on real and auxiliary storage use in relation to the 64-bit memory object allocated by DB2 for z/OS above the 2 GB bar.

DB2 10 for z/OS can request z/OS to provide information about real and auxiliary storage, based on a particular addressing range. It provides proper monitoring when you have multiple DB2 subsystems running on the same LPAR. It also provides some protection against the system paging, overcommitting real storage, or running out of AUX storage. DB2 can free unused memory back to the z/OS operating system.

When should you migrate to DB2 10 for z/OS?

A "normal" migration is moving one version at a time every three years. For customers with even earlier versions, the ability to skip a migration cycle will be attractive, but this ability is not "something for nothing." Customers need to consider the tradeoffs and challenges in a "skip version" migration. Most customers who migrate to a new version by three years after the general announcement (GA) of the respective new release are already on DB2 9 for z/OS.

The project for skipping a release is larger. While the testing and rollout are only a little greater than a single version migration, the education and remediation work is roughly double the size; most project plans estimate 150 percent. Consider the timing carefully. Improvements in DB2 9 for z/OS are

delayed with a "skip" release migration plan. You may need to have extended service on DB2 8.

You will find more details about the "when to migrate" decision in the IBM Technote at *http://www.ibm.com/support/docview.wss?uid=swg21006951*.

In summary:

- We recommend the regular application of preventive service maintenance. It should be a continual process.
- Testing should be performed over and above that performed by DB2 for z/OS Development.
- CST testing still does *not* replace customer regression/stress testing.
- You must be prepared to tolerate some "bumps in the road."
- Customers who are not prepared to take mitigating actions and have no tolerance for "bumps in the road" should not be early adopters and should migrate directly to DB2 9 for z/OS.

For customers who are still running DB2 V7, the option to skip from DB2 8 to DB2 10 for z/OS is very attractive and makes the current path clear. Customers who have just migrated to DB2 8 may like this alternative, for the short term.

DB2 10 for z/OS supports migration from DB2 9 for z/OS NFM or from DB2 8 NFM. Customers not yet running DB2 8 or DB2 9 for z/OS should plan to migrate first to DB2 for z/OS V8, as preparation for an eventual migration to DB2 10 for z/OS.

We estimate that about one in five customers migrated using a "skip version" technique from DB2 V5 to DB2 V7, and we expect to see a similar proportion this time. The savings for skipping a version migration are less than 50 percent, since the education and needed application and administration changes are about the same. Customers who do skip migration report that the project takes longer, about 50 percent longer than a normal migration path.

Changing from DB2 8 or earlier to DB2 10 for z/OS will require a cultural shift that some describe as "culture shock." If customers spend the bulk of their migration project time in testing, savings could be up to 40 percent. But most customer plans should expect 20 to 25 percent reduction, compared with two migrations.

The tradeoff for skipping is primarily the later delivery of DB2 9 for z/OS improvements, namely CPU savings, especially in utilities and disk savings via compression for indexes, improved insert and update rates, improved SQL, and pureXML for developer productivity, as well as better availability.

Summary

To summarize, DB2 10 for z/OS is a very good release in terms of the opportunities for price/performance and scalability improvements. There is significant DBM1 31-bit VSCR after rebind as soon as DB2 10 for z/OS CM. You can use 1 MB size real storage page frames on z10 and z196 processor. But the key is the use of long-term page fix on local buffer pools. There are also improvements in terms of reduced latch contention and latch management overhead.

Over and above the "out-of-the-box" performance improvements as a result of BIND/REBIND and the use of 1 MB size real storage page frames, there are opportunities for further price/performance improvements *provided you have enough real memory*. It is a classic tradeoff between increased real storage provision in order to reduce CPU resource consumption. This includes making more use of persistent threads both for legacy CICS and IMS/TM applications, as well as the use of high-performance DBATs for DDF workloads.

If you have enough real memory, you can make greater use of the BIND option RELEASE(DEALLOCATE) with these persistent threads. But you must recognize that increased use of the BIND option RELEASE(DEALLOCATE) is a tradeoff; it will lead to increased storage consumption, and you will need to plan for additional real memory over and above the required 10 percent to 30 percent increase just to stand still when migrating to DB2 10 for z/OS.

The use of the BIND option RELEASE(DEALLOCATE) with persistent threads can also reduce concurrency because BIND/REBIND and SQL DDL activity will not be able to break in to work.

DB2 10 for z/OS also provides opportunity for the greatly enhanced vertical scalability of an individual DB2 member in data sharing and the potential for LPAR/DB2 consolidation.

You must carefully plan, provision, and monitor real storage consumption. Early customer adopters of DB2 10 for z/OS, migrating from either DB2 8 or DB2 9 for z/OS, should make plans and take extra care to mitigate the risk of instability. Those steps include:

- Plan regular full "major" maintenance drops.
- Use CST/RSU recommended maintenance.
- Perform application regression and stress testing to keep problems away from production
- Be prepared to tolerate some "bumps in the road."

Gaining the Financial Benefits of DB2 10 for z/OS

Why, how, and where performance matters

by Cristian Molaro

Introduction

Today's business faces a highly competitive and demanding environment; good performance is, in many cases, becoming a prerequisite for business continuity.

For example, the concept of high availability nowadays means more than being available; it has evolved into being available *and* providing good response times. With more critical systems and an ever-increasing number of online users, database management systems, such as DB2 for z/OS, play a critical role in achieving high availability.

However, providing fast and consistent response times should not create a conflict with another of today's main concerns: reducing costs.

It is important for organizations to realize that good performance can address both requirements at the same time: fast response time by process optimization and cost reduction by reduced CPU utilization. This is why performance matters.

DB2 10 delivers business value out of the box by providing immediate CPU savings after migration. Organizations committed to good performance get even more value by exploring and implementing the new and improved features offered by DB2 10.

There is more than CPU reduction; DB2 10 provides unprecedented improvement in scalability, supporting from 5 to 10 more concurrent users per single DB2 subsystem. Many internal changes have synergy with System z, improving the throughput of today's most demanding applications.

DB2 10's potential to improve performance is creating a high level of expectations as to how it can provide cost savings. DB2 10 can realize budget savings for many areas in the organization, such as database administrators, architects, and decision makers.

Common questions are:

- How can DB2 10 performance realize cost savings in total cost of ownership (TCO)?
- How should the migration return on investment (ROI) be calculated?

The purpose of this paper is to teach you why, how, and where performance matters and to help you build the business case for migrating to DB2 10. We analyze how DB2 10's performance benefits can realize cost savings. In addition, we provide guidelines for building an effective costing scenario and show you the potential benefits to be gained when migrating to DB2 10 for z/OS.

Why Performance Matters

A recent worldwide study[1] collected today's primary IT business concerns among database administrators, managers, and decision makers. Part of this study was a survey that gathered more than 1,100 answers from active IT professionals involved with the DB2 family of products on all platforms, but mainly on z/OS.

Performance was one of the most frequently mentioned points of interest.

Figure 2.1 documents the distribution of answers to one of the questions, *"What are the main concerns of your IT business?"*

As one can see, nearly 70 percent of the participants indicated that improving performance was a major concern to their organizations. Only two aspects mattered more than performance in the results: *Availability & Reliability* and *Reduce Cost*.

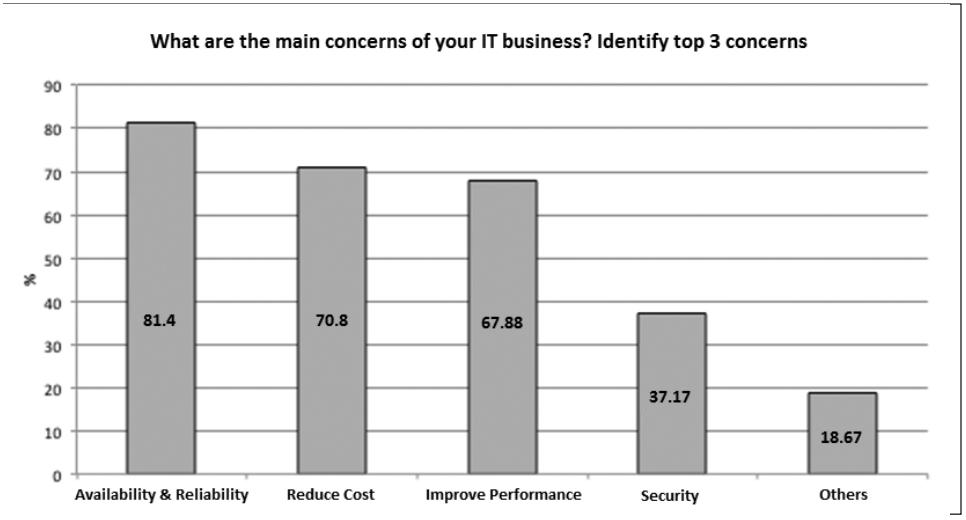

Figure 2.1: The top concerns of today's IT business

Availability & Reliability can be described as *the identified needs for assuring business continuity*. This aspect can be related to concerns regarding the guarantees of business survival. In today's highly competitive world, a relatively short IT outage can mean significant losses in both money and reputation. A major outage can result in being out of the business permanently. In the past few years, many organizations have invested huge amounts of resources to guarantee high availability and disaster recovery. Many businesses have invested in dual or triple datacenters and state-of-the-art high-availability solutions, such as IBM's Parallel Sysplex® technology[2] and DB2 for z/OS data sharing environments.

Reduce Cost is also related to an organization's survival. The ability to provide the same or better service at a reduced TCO has been, is, and will continue to be one of the main objectives for any business evolving in today's competitive market. This has been particularly true during the past few years, during which the business continuity of many organizations had to be assured by cost reduction campaigns. Almost every company in the world has been impacted to some

degree by the need to deliver business at a reduced cost. For some organizations, projects that focused on efficiency and optimization were the only projects allocated funding in the past several years. Even if the financial and economic worldwide climate is improving, TCO optimization will remain a main concern in the future as an indication of a company's willingness to take a leadership role in today's very competitive landscape.

Interestingly, the above-mentioned survey responses follow the structure proposed by "*Maslow's hierarchy of needs"*[3] as shown here in Figure 2.2.

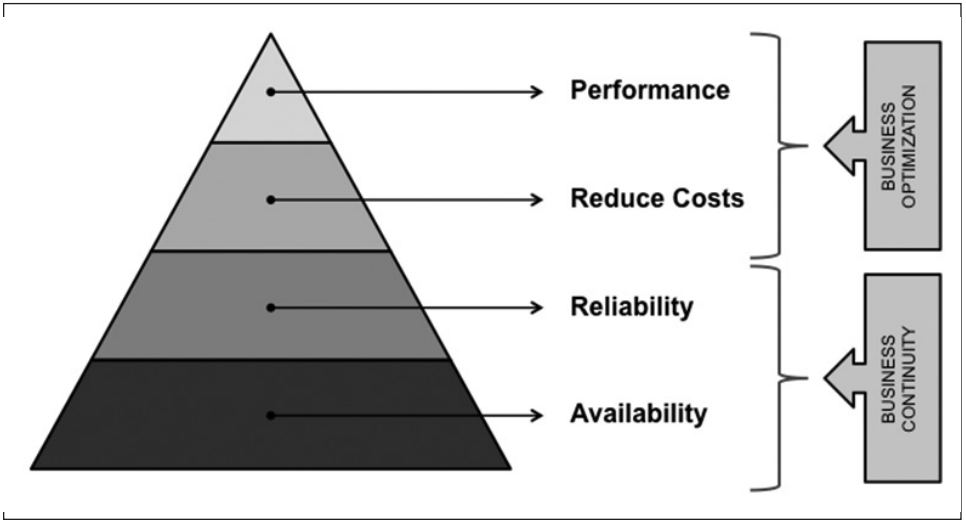

Figure 2.2: The hierarchy of today's main IT business concerns

This figure graphically and visually represents today's main IT concerns or needs. You need to read the pyramid from the bottom, where the most fundamental concerns or business needs are represented, to the top. Companies should concentrate efforts on the upper levels only after the needs of the more fundamental lower levels are satisfied. In fact, organizations that focus only on reducing cost or improving performance but lack a solid high-availability plan risk being exposed to a business disruption or being out of business in case of a major disaster.

In this particular representation, *Availability* and *Reliability* have been classified under *Business Continuity* because they are key concepts involved in guaranteeing the survival of the organization. *Reduce Costs* and *Performance* are

under *Business Optimization* because these concerns may become important only after business continuity has been ensured.

Considering the current worldwide economic and financial climate and the fact that we are evolving into a more and more competitive business environment, we could agree that *Availability*, *Reliability*, and *Cost Reduction* are necessary conditions to business survival. This idea is represented in Figure 2.3.

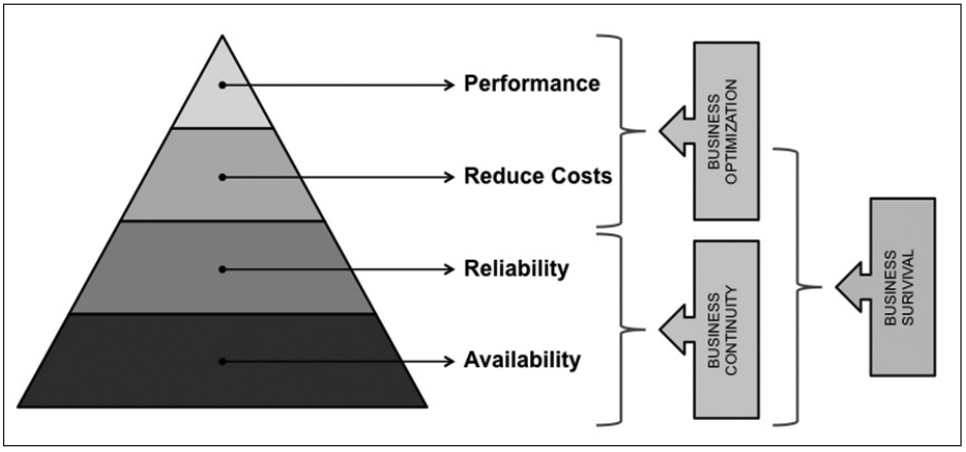

Figure 2.3: IT business concerns and business survival

In some organizations, performance is considered an important but secondary concern. However, today's business demands a higher priority when it comes to performance. Why? IT organizations today are aware of the fundamental impact that good performance has on the company's other main concerns, such as reputation and high availability.

For this reason, the past couple of years have seen the definition of high availability changing rapidly from simply *being available* to *being available and having excellent response time*. With many applications providing a Web interface, and thus having more online users, the organization now has a near-instantaneous response time expectation. A modern application must be constantly available *and* must provide fast and consistent response time. Excellent response times can be achieved by optimizing the IT infrastructure, applications, and middleware to provide fast response times to online users. Nowadays, many applications query large amounts of historic data, and the organization still expects to deliver a good response time. DB2 for z/OS is a

database management system that can handle this kind of workload and response requirement.

Providing fast and consistent response time is often achieved by increasing the available processing power. But this approach has a big downside because an increase in computational power creates a conflict with the other main concern: reducing costs.

It is important, then, to realize that DB2 10 can address both requirements at the same time: fast response time through process optimization *and* cost savings through reduced CPU utilization for a transaction. This is why performance matters!

Good performance benefits on response time are straightforward to visualize. The same is true for the financial benefits, although estimating the effective impact on TCO is not always easy. We discuss this point in detail later.

Good performance can also provide benefits that are difficult to measure but that can provide priceless competitive advantages, such as customer satisfaction or the capacity to absorb seasonal influences without service degradation or the need for additional resources.

Nowadays, good performance is a necessary condition for the survival of a company and its sustained growth.

IT organizations that realize the potential of good performance also discover that optimizing performance is a never-ending process. Many organizations invest time and resources on a permanent basis in the pursuit of coping with the basic needs of *Availability* and *Cost Reduction* through good *Performance*.

However, in very rare cases, we have the opportunity to improve performance *out of the box* by simply installing a new release of our database management system.

DB2 10 for z/OS is the industry state-of-the-art database solution for businesses with the most demanding high availability and reliability requirements. By providing improved performance with savings out of the box, DB2 10 effectively addresses one of today's main IT business concerns.

DB2 10 for z/OS Performance and Cost Savings

DB2 10 for z/OS provides performance benefits that have the potential to result in a reduction of the total cost of ownership.

Executive summary

Most DB2 installations achieve immediate CPU savings in the range of 5 percent to 10 percent by migrating to DB2 10 for z/OS. Some specific but commonly found workloads can even receive a benefit of up to 20 percent CPU reduction.

Improvements can be realized immediately after migration. This is known as *out-of-the-box savings*. Customers who decide to invest in getting the most of the performance improvements of DB2 10 may further increase these savings by exploiting the many new DB2 10 features.

On top of CPU improvements, DB2 10 provides unprecedented savings in the storage areas that previously limited the volume of concurrent workloads, providing a dramatic scalability improvement. Customers can achieve additional TCO savings possibilities by exploiting scalability benefits such as consolidating DB2 system and DB2 members in a data sharing setup.

DB2 10 also provides functional and SQL enhancements targeted to improve performance, simplify current and new applications, and reduce the development cycle time to market. These features have the potential to reduce financial costs by optimizing the way in which we use, develop, and manage applications that use DB2 as their database management system.

Detailed analysis

DB2 10 for z/OS provides performance improvements in many ways. On top of important CPU savings, often achievable out of the box, it delivers other TCO reduction opportunities, which are discussed later in this paper.

DB2 10 performance enhancements and savings go beyond only CPU savings. If for the same installed capacity you will see a throughput improvement, this means you are effectively increasing your potential to deliver business.

A CPU reduction at peak times can immediately reduce your TCO by reducing software costs related to CPU utilization. It also can provide you savings by delaying, or even avoiding, an increase in installed capacity or a server upgrade.

CPU is often the center of the discussion when it comes to performance analysis and tuning, but many other variables need to be considered and are just as important.

The DB2 10 performance improvements can be classified in three main categories:

- **CPU reductions:** DB2 10 enhancements provide better performance with less CPU utilization compared with DB2 9. Many of these changes are available without a need for code or parameter changes. Further savings can be achieved after infrastructure, database, or application changes. In addition, DB2 10 extends the range of the CPU usage that can be off-loaded to a specialty engine, contributing to further reduce the billable CPU.
- **Improved scalability:** Impressive reductions in memory requirements, which used to limit the concurrent database activity, provide 5 to 10 times more concurrent workload support per DB2 subsystem. DB2 logging and the internal DB2 serialization mechanisms (latches) have been improved, giving an overall increased throughput per DB2 where they might have been the limiting factor in previous versions. These changes open new consolidation opportunities where users can consider reducing the required number of DB2 subsystems for the same workload with the potential of obtaining further performance improvements and TCO savings.
- **Productivity enhancements:** DB2 10 provides enhanced and new features that have the potential to accelerate productivity and reduce administrative requirements. New features, such as temporal data support, provide the opportunity to push into the database logic that is implemented in application logic today in coding and database objects. This capability provides an effective reduction of resource utilization with an added simplification of maintenance and development.

DB2 10 for z/OS introduces the concept of **out-of-the-box savings**: no database or application changes are required to gain immediate benefits.

To better understand the performance saving potential involved in migrating to DB2 10, we could consider putting the DB2 10 performance objectives in perspective. Figure 2.4 compares the goal CPU changes, in percentage, through the history of DB2.

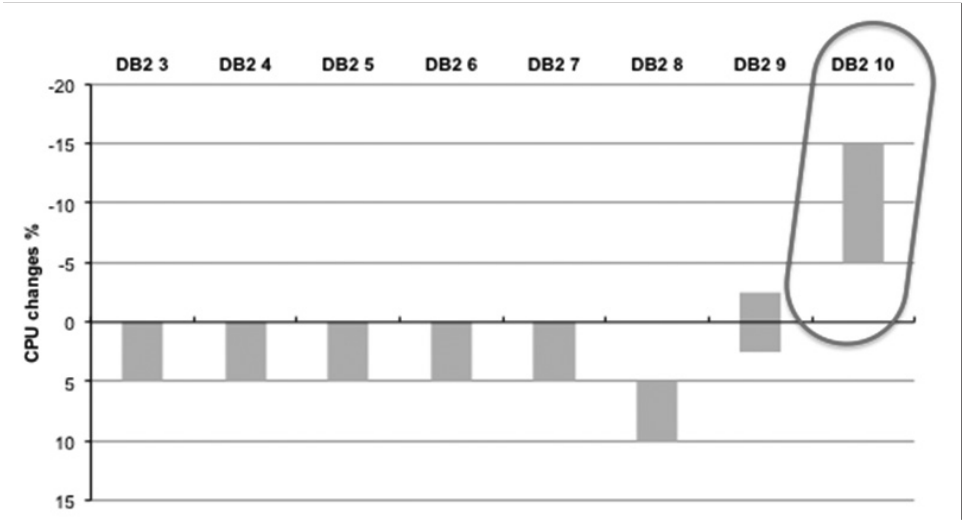

Figure 2.4: DB2 10 performance objectives

Each new version has provided new functionalities and features, extending the DB2 value in many ways. A reasonable CPU regression was always expected and accepted in exchange for a better product—in other words, a more powerful database. And a rational exploitation of the increased business value should more than compensate for a contained overall CPU increase.

The expectations have been stable since the beginning of DB2 until DB2 V7, with the objective of a CPU regression in the 0 percent to 5 percent range.

DB2 8 provided 64-bit addressability and the ability to manage huge amounts of real store at the price of a more important CPU regression. In many cases, the increased CPU could be compensated for with other resources (e.g., by larger buffer pools).

DB2 9 set the trend toward a more contained CPU increase after migration. DB2 9 also improved memory utilization below the 2 GB bar.

As it is clear from the figure, DB2 10 is the first release that has been assigned the goal of providing CPU savings in the range of 5 percent to 10 percent. Not only do you get a rich DB2 version with new and improved features; you also get CPU savings at the same time. In addition, DB2 10 provides big improvements in storage requirements below the 2 GB bar, dramatically increasing the scalability of DB2. It has the potential to support from 5 to 10 more concurrent database accesses. However, there is a bigger requirement for total storage in DB2 10.

You need to plan and possibly provide more real storage in order to avoid system paging, which has a negative impact on performance.

A key concept is that CPU savings are workload- and environment-dependent, and results will vary.

Bearing this point in mind, we can examine the results obtained by IBM under controlled and repeatable workload conditions. Figure 2.5 provides an overview of the several workloads tested by IBM.

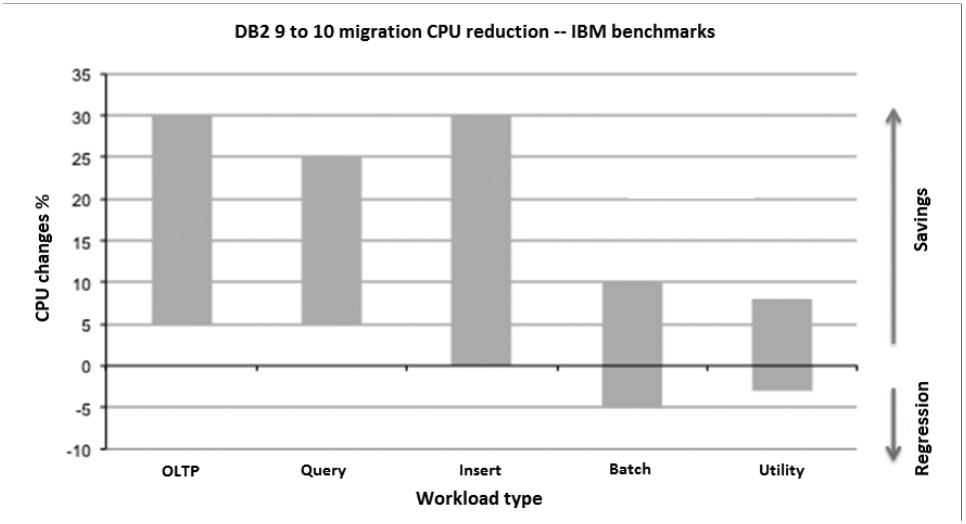

Figure 2.5: DB2 9 to DB2 10: observed CPU changes

The results were grouped by workload category. Positive changes, expressed as percentages, indicate CPU savings. Negative values indicate CPU regression. In most cases, the benchmarks were based on customer data, and these results were almost always confirmed by the customer running DB2 10.

The tests were done under controlled conditions and are repeatable; this is a requirement for quality statistical analysis. The range of values for each type of workload is then not a side effect of the test conditions but the summary of many different test scenarios and conditions for workloads of the same type. The benchmark results enforce the concept of variance in the expected savings; as mentioned before, CPU improvement is workload-dependent, and your results will also vary.

Setting the correct level of expectations

Because each organization, each application, and each environment is unique, it is not practical to design a single standard methodology that is able to forecast how much resource utilization improvements, or savings, you can get from DB2 10. The potential benefits are workload- and environment-dependent, and results will vary; keep in mind this key concept while reading this document and while preparing your own DB2 10 for z/OS business case.

> Almost certainly, any organization will get, in one way or another, performance benefits from DB2 10. Certainly, however, results will vary depending on many conditions.

But what could you expect? For most workloads, you may obtain up to 10 percent CPU reduction for static SQL after you REBIND packages. You may observe an even higher improvement with workloads that had scalability issues in previous versions of DB2 or in distributed applications targeting DB2 for z/OS and exploiting DRDA and dynamic SQL.

Some workloads are particularly susceptible to showing big savings; for example, those exploiting native SQL/PL procedures can have up to 20 percent CPU reduction. Query workloads will also show significant improvements as a result of many positive access path changes. For more detailed discussions on this subject, refer to the IBM Redbook *DB2 10 for z/OS Performance Topics* (SG24-7942).

The variance in savings is not the only challenge: Answering the question about how these improvements will impact your total cost of ownership may be an even more demanding exercise. Nevertheless, the available information, the user experiences, and the documented performance improvements, combined with a good knowledge of your business cycle, make it possible to model the financial benefits of migrating to DB2 10 for z/OS with an acceptable degree of approximation.

As is natural in any modeling exercise, the quality of the input data influences the quality of the output data. A good knowledge of your environment is of capital importance.

During the construction of a business case, you often do not control, or do not have, all the variables of the scenario. Such is the case when estimating the

CPU savings that DB2 10 for z/OS can provide to your organization and, more important, what its impact will be on the TCO.

Some cases are easier to model than others, and the degree of incertitude varies. It is fundamental to understand that even if there are very high chances that your organization will achieve CPU savings when migrating to DB2 10 for z/OS, the degree of those savings depends in large part on the nature of your workload.

Incertitude is a component of most of the business cases, and analysts need to make assumptions. Taking assumptions on the CPU savings that you will get from DB2 for z/OS can be part of the business model exercise. As for any business case, make sure that your assumptions are mentioned as such in the report.

The many paths to get the DB2 10 savings

DB2 10 obtained General Availability (GA) status on October 22, 2010. At the time of this writing, DB2 10 has reached a level of maturity that is considered to be stable and reasonably safe by many non–early-adopter organizations. Early observations and migration experiences are being shared within the DB2 community, and a lot of documentation support is available.

During the early stages of your DB2 10 migration planning, you should consult these references:

- Information APARs: II14477 and II14474
- Toleration of fallback and data sharing coexistence APAR: PK56922
- Pre-migration checkout APAR: PM04968

A very simplified DB2 9 to DB2 10 migration path is documented in Figure 2.6.

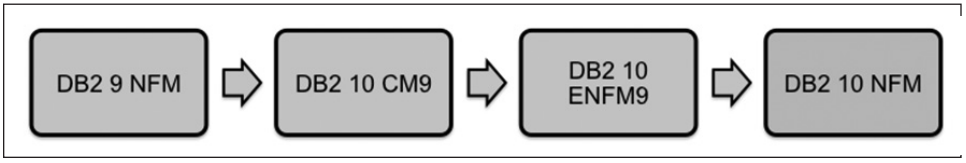

Figure 2.6: Migration to DB2 10 from DB2 9

When migrating from DB2 9, you start from DB2 9 New Function Mode (NFM). The first stage in DB2 10 is named DB2 10 Conversion Mode (CM9);

from here, you can fall back to DB2 9, if required. The next steps are DB2 V10 Enabling New Function Mode (ENFM9) and finally DB2 10 New Function Mode (NFM).

The maximum performance potential is available in DB2 10 NFM. However, because DB2 10 delivers many of its performance benefits in Conversion Mode, the first migration step is also interesting.

Some of the most relevant DB2 10 performance features grouped by migration effort are listed here below:

- **DB2 10 Conversion Mode**
  - Improved performance of SQL at runtime
  - Faster single-row retrievals via open-fetch-close chaining
  - Parallel index update at insert
  - Query parallelism improvements
  - Workfile in-memory enhancements
  - Insert improvement for universal table spaces (UTS)
  - Index list prefetch
  - Memory changes exploiting more 64 bit, some after REBIND
  - Increased DDF performance (high-performance database access threads, or Hiper-DBATs)
  - Buffer pool enhancements
- **DB2 10 New Function Mode**
  - Efficient caching of dynamic SQL statements with literals
  - Faster fetch and insert
  - SQL Procedure Language performance improvements
  - MEMBER CLUSTER for UTS
  - Utility enhancements
- **DB2 10 New Function Mode features requiring changes**
  - Hash access path
  - Index include columns
  - Inline large objects (LOBs)

Functional details about these and other features are available in the Redbook *DB2 10 for z/OS Technical Overview* (SG24-7892). Performance considerations and observations are reported in the Redbook *DB2 10 for z/OS Performance Topics* (SG24-7942).

Because of the catalog restructure, DB2 10 provides a higher level of concurrency to operations involving DB2 Catalog access, such as BIND and

REBIND operations. This change effectively increases the scalability of these workloads, which were often single-threaded, with the potential to reduce the time required for some maintenance job streams.

Because of constraints related to providing fallback capabilities from Conversion Mode to the originating DB2 version, DB2 8 or DB2 9, there is some performance degradation in this mode for operations involving access to the DB2 Catalog (CM mode does not have the restructured catalog).

These constraints are relaxed further down in the migration path, and you may consider planning a quicker move forward to the next stages of the migration process than for previous DB2 migrations.

By doing so, you not only improve the performance of catalog-related operations but also give your organization the opportunity to exploit the DB2 10 performance benefits available in New Function Mode.

DB2 10 provides the *skip release* migration path; you can migrate to DB2 10 from DB2 8. This opportunity enables organizations to accelerate the migration process and to get into the DB2 10 benefits in a shorter time frame. Skipping DB2 9 in the process can also reduce migration costs. Figure 2.7 provides a basic representation of the steps involved.

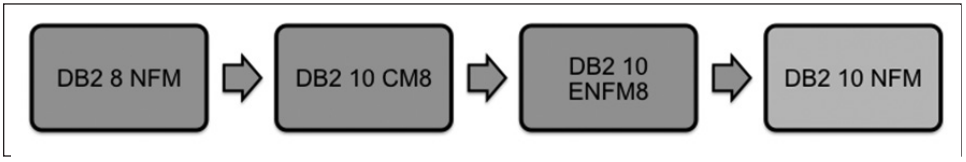

Figure 2.7: Migrating to DB2 10 from DB2 8, Skip release

Nevertheless, as the technology change is bigger in this scenario than in the normal process, you need to take special considerations to balance the potential migration saving benefits with the eventual stability risks related to the need to absorb the technology gap between DB2 8 and DB2 10.

From a management point of view, a risk assessment analysis is needed to make a choice between a normal migration scenario and the skip release scenario. Common risk-mitigating considerations include the need for longer planning, a more careful testing, and more investment in education.

Also consider that skip migration is a one-direction process only: once a DB2 subsystem has been in DB2 V10 CM8—the first step and thus the start of the skip migration—it becomes impossible to migrate to DB2 9.

Savings in the migration process per se should not be your only driver toward a skip release migration. A certain amount of risk is involved using this migration path.

On top of considering the skip release migration path as a quicker way to get the DB2 10 benefits and functionalities, you need to carefully weigh what is involved in absorbing two DB2 releases in a single step.

The performance "migration path"

There is a process involved in gaining the performance benefits of DB2 10. Knowing how you get the expected advantages can help you to better plan your migration strategy.

In some cases, you can get performance benefits from DB2 10 even before installing the product. For example, the preparation for the migration process includes the removal of the DB2 Private Protocol (PP). Private Protocol is not supported in DB2 10, and applications that attempt to execute PP will fail. PP is used for distributed access to DB2 for z/OS, typically for communications between DB2 subsystems using three-part names. PP has been deprecated in favor of the Distributed Relational Database Architecture (DRDA). DRDA is the industry standard embraced by IBM for distributed access to any member of the DB2 family and beyond. DRDA is a state-of-the-art protocol for distributed access, and this is in a strong contrast with PP, which has not been enhanced since DB2 V5.

By migrating from PP to DRDA, you get features that improve performance, such as support of packages and stored procedures, on top of an optimized protocol that has the potential to reduce CPU and elapsed time on distributed access to DB2.

Preparing to migrate to DB2 10 is also a great opportunity to review the DB2 Client, Driver, and DB2 Connect versions in use in your organization. To fully exploit the DB2 10 enhancements for distributed access, you need at least to deploy Version 9.7 Fix Pack 3a. The potential to obtain performance benefits exists as a consequence of exploiting some of the latest developments on the communications protocol. For example, there is a dramatic reduction in the total elapsed time involved in the communications between a distributed application and DB2.

As part of the migration preparation, you need to review the real storage (memory) utilization in your logical partitions (LPARs). DB2 10 allows big savings in memory utilization below the 2 GB bar, providing a huge step forward in scalability. However, there is a net increase in the total amount of real storage required for the same settings. In other words, there is an increase in memory requirements by the DB2 subsystem.

The eventual provision of more storage and, more important, the review of the system paging activity could help to improve the overall health of a z/OS image in case of paging.

Packages bound in DB2 V5 and earlier versions of DB2 are not supported in DB2 10. Also, plans that contain database request modules (DBRMs) are not supported by DB2 10. Although DB2 10 can convert DBRMs "in-flight," it is best if you convert them to packages beforehand. Because DB2 10 might force you to rebind these older packages and plans, you will almost certainly obtain performance improvements by exploiting new DB2 features that are available only through rebind.

Paralleling the stages involved in the migration process, we could build a picture representing what could be called a *performance migration path*. In other words, how do you get the DB2 10 performance benefits through the migration process? Figure 2.8 depicts the DB2 10 performance migration path.

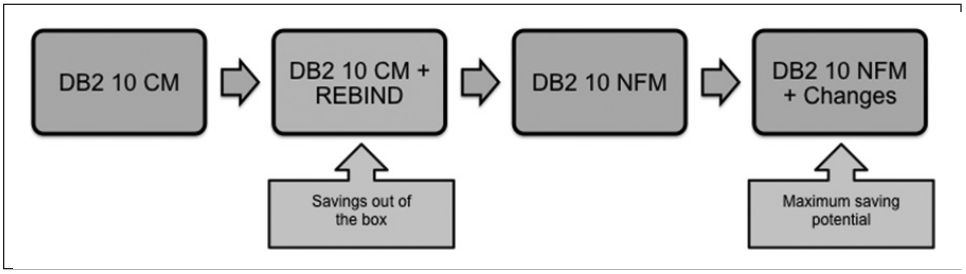

Figure 2.8: The DB2 10 performance migration path

Of course, the first stage is to migrate to DB2 10 Conversion Mode. As mentioned, a distinctive characteristic of DB2 10 is that many performance improvements are available already in DB2 10 CM.

With the exception of packages bound in DB2 V5 or previous DB2 releases, REBIND is not a migration requirement. However, a BIND or REBIND is required to fully get the maximum benefits of CPU and memory savings provided by

DB2 10. In addition, a REBIND in DB2 10 will use the improved DB2 optimizer, and improvements in access path could be the result. For instance, some SQL predicates could use indexes that were not possible in previous versions. As always, the normal consideration using the REORG–RUNSTATS–REBIND cycle applies to this scenario.

So, the second stage in your performance path is to REBIND. And you should plan to do so as soon as practically possible in order to get performance benefits and to prevent an eventual performance regression.

Many organizations are very cautious about the risks of performance degradation related to an access path change involved in a REBIND. DB2 10 extends the advantages provided in DB2 9 by the Access Path Stability feature, giving an even more robust solution to manage and control these risks. In some scenarios, you can exploit this framework during the migration process to gain a degree of protection against unexpected changes in access paths. DB2 10 exploits, by default, the EXTENDED Plan Management mode, keeping up to three copies of a package for access path fallback purposes. If you are not exploiting Access Path Stability in DB2 9, or if you are migrating from DB2 8, you should consider exploiting this feature. Access Path Stability has already proven value in Version 9 but has been further improved in DB2 10.

REBIND is needed only once per package for the migration; there is no need to perform a REBIND again during your path through the migration process.

Dynamic SQL will get immediate benefits as a consequence of the prepare stage involved in the preparation of the statement for execution.

The next stage is to move to DB2 10 New Function Mode (NFM); this is the third box in Figure 2.8. The full DB2 10 potential is released in NFM. New objects, new SQL, and new functionalities are now available. They have the potential to improve even further the performance of DB2 and applications accessing DB2.

To exploit the new functionalities, there is often some investment involved. Database changes, new SQL syntax, or development techniques will be required. The ROI can be substantial, with improvements that are beyond resource savings; new DB2 10 features such as MEMBER CLUSTER support for, and easy conversion to, universal table spaces, XML enhancements, in-line LOBs, hash access, and temporal support can not only improve performance but also reduce development time requirements. The last box of Figure 2.8 represents this stage.

Organizations exploring and implementing the new features made available in NFM (organizations that we could qualify as *committed to good performance*) are in the best position to exploit the full potential of DB2 10 for z/OS.

Performance benefits for distributed applications

A very common architecture today consists of Java applications running on an application server, such as IBM WebSphere® Application Server, connecting to DB2 for z/OS. This kind of access to DB2 is also known as *distributed access* to DB2 for z/OS. In a high-availability configuration, these applications may connect to a data sharing DB2 group running in a Sysplex environment.

In such an environment, the DB2 drivers, the TCP/IP components of z/OS, z/OS Workload Manager (WLM), and the DB2 data sharing group work together to provide the applications with a transparent high availability and workload balancing when accessing data residing in DB2 for z/OS.

This configuration provides the state-of-the-art distributed access to DB2 and is also referred to as DB2 Sysplex Support. Figure 2.9 shows a schematic representation of this configuration.

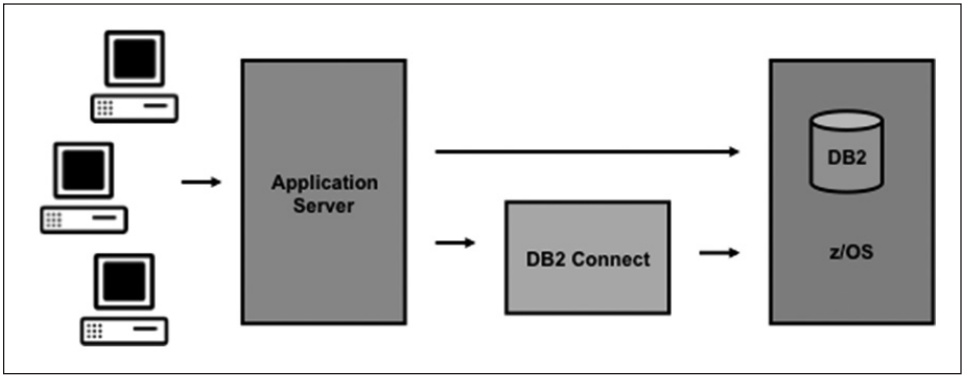

Figure 2.9: Distributed applications

DB2 10 for z/OS provides important performance improvements of special interest for distributed applications, such as:

- High-performance DBATs
- Improved return to client result sets
- Enhanced support for native SQL/PL procedures
- Extended correlation token
- Virtual and real storage improvements
- LOBs and XML materialization avoidance

Again, refer to the Redbooks *DB2 10 for z/OS Technical Overview* (SG24-7892) and *DB2 10 for z/OS Performance Topics* (SG24-7942) for more details.

DB2 Connect, DB2 Clients, and DB2 Drivers 9.7 Fixpack 3a or later fully exploit the DB2 10 capabilities for remote applications. In many cases, a DB2 Connect server is not required to exploit the full capabilities of the distributed access to DB2. Removing a DB2 Connect Server from the infrastructure provides a simpler configuration and will improve performance by reducing the code path between the application and DB2.

Several tests[4] with distributed applications were executed to document the performance benefits of DB2 10. Some of the results of many tests, designed to explore the performance benefits of DB2 10 for distributed applications, are documented in Table 2.1, showing the savings in CPU.

Workload	DB2 9	DB2 10	Savings
SQL ODBC/CLI (dynamic SQL)	2114	1997	5.5%
JDBC: Dynamic SQL	2152	2017	6.3%
SQLJ: Static SQL	1899	1761	11.9%
Stored procedures in SQLJ with static SQL	1768	1642	6.7%

Table 2.1: Out-of-the-box savings for distributed applications (CPU mSecs)

These tests focused on the most common connectivity options; Java drivers were JCC Type 4. The only change between scenarios was the migration to DB2 10. The savings are expressed as a percentage (%) of total CPU, including the DB2 address spaces. These are improvements *out of the box*.

The RELEASE option of the BIND and REBIND commands determines when to release resources that a package uses, either at each commit point or when the program terminates. RELEASE(DEALLOCATE) can increase the package or

plan memory requirements at execution time because additional items become resident. In compensation, it has the potential to reduce CPU by minimizing the allocation and deallocation processing.

For many DB2 versions, the RELEASE option of distributed applications has not been honored. Regardless of the RELEASE bind option specified for the package, the behavior has always been RELEASE(COMMIT).

DB2 10 provides a very important reduction in storage requirements below the bar of 2 GB. This relief allows the reconsideration of BIND options that were not practical in previous DB2 versions due to storage constraints, including RELEASE(DEALLOCATE).

The new PKGREL option of the -MODIFY DDF command specifies whether DB2 ignores the bind options of packages that are used for remote client processing. PKGREL can be modified by two options:

- BNDOPT: The rules of the RELEASE bind option that were specified when the package was bound are applied. This is the default option in DB2 10.
- COMMIT: The rules of the RELEASE(COMMIT) bind option are applied. This is the DB2 9 (and earlier) behavior.

Running distributed applications with RELEASE(DEALLOCATE) improves performance in many cases. However, it could become difficult to execute some operations, such as DDL, on objects allocated by the workload. PKGREL(COMMIT) allows the package to dynamically revert to the DB2 9 behavior when needed.

The results in Table 2.2 were collected using the DB2 10 PKREL(COMMIT) option—that is, running in the same conditions as in DB2 9. These observations can be extended to show the benefits of RELEASE(DEALLOCATE), as shown in the table[5].

Workload	DB2 9	DB2 10	Savings	DB2 10 DEALLOC	Savings
SQL ODBC/CLI (dynamic SQL)	2114	1997	5.5%	1918	9.3%
JDBC: Dynamic SQL	2152	2017	6.3%	1855	13.8%
SQLJ: Static SQL	1899	1761	11.9%	1668	16.6%
Stored procedures in SQLJ with static SQL	1768	1642	6.7%	1550	11.9%

Table 2.2: Out-of-the-box savings for distributed applications with RELEASE(DEALLOCATE)

Applications running with AutoCommit can achieve further benefits. Applications running packages bound with the RELEASE(DEALLOCATE) option in combination with PKGREL=BNDOPT have shown up to 37 percent CPU savings with AutoCommit ON.

Building the DB2 10 for z/OS Business Case

DB2 10 for z/OS provides many performance enhancements with the potential to reduce TCO. Building a good business case and a valid ROI is a multidisciplinary process involving many participants. Organizations investing in understanding how their applications work and how DB2 10 can bring benefits to them are in the best position to realize why, how, and where performance matters.

Executive summary

The DB2 10 potential to effectively reduce costs by a CPU reduction is related to the costing model in effect in your organization. You must have a clear vision of how CPU utilization is reflected in your TCO.

A basic and fundamental requirement is to understand your environment. An adequate understanding of the way in which CPU is used will help you to select the best cost-saving scenario.

> Building a model for the TCO impact of DB2 10's CPU reduction is a multidisciplinary challenge. It involves DB2 administration, capacity management, purchase management, and performance skills. The level of accuracy of a benefit projection is related to the quality of the data and the knowledge of the environment.

This section covers the basics needed to guide you through the process of building a DB2 10 for z/OS business case.

The business case building process

If your goal is to estimate the impact of the DB2 10 savings on your organization's TCO, you need to combine the knowledge about your applications, the estimation of the DB2 10 savings for your workload, and the understanding of the components of your TCO.

As discussed in this paper, there is more saving potential in DB2 10 than just CPU improvements. Many other saving opportunities exist, and in some cases estimating the actual financial added value can be challenging. Be sure to explore all the DB2 10 possibilities, keeping in mind that the business value of DB2 10 certainly has the potential to improve performance and increase productivity while saving costs.

Building the business case for DB2 10 is a process. Figure 2.10 represents its main steps.

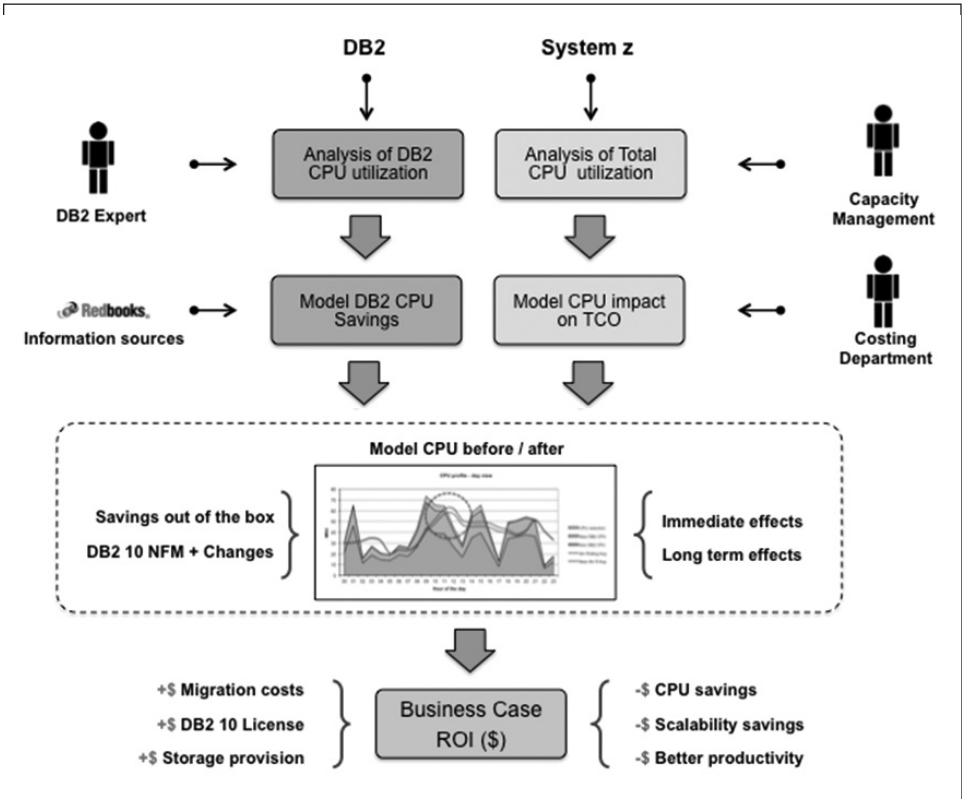

Figure 2.10: The process of building the DB2 business case

One of the purposes of this figure is to emphasize that the process is a multidisciplinary effort. The diagram shows a DB2 information path at the left and a System z path at the right.

The DB2 side of the information process starts with an analysis of your DB2 CPU utilization. A DB2 specialist collects the DB2 accounting information to describe and document how DB2 is used in your company. For the purposes of this paper, the information about the time frame when DB2 is used is also important. The next step is to estimate the CPU potential savings from DB2 10. There are many sources of information to support you in this process. You should document separately which benefits are to be obtained out of the box and which ones will require you to invest in system or application changes.

The System z side starts with the analysis of the total CPU utilization. In building the business case, you need to evaluate how the DB2 10 savings could affect the global CPU; this includes processes that are not related to DB2. The next step is to understand how CPU is related to TCO. Because you are modeling how CPU savings can provide financial benefits, you need to understand how a potential CPU change can reduce TCO.

Both DB2 and System z information can be used together in a CPU modeling study. You need to do this before and after migrating to DB2 10 and study the differences. We discuss the details involved in the following sections.

Now, you have the elements that are required to build the DB2 10 business case and to calculate the migration ROI. Here, you merge the information about the models and the CPU impact on TCO. Also consider other cost factors, such as migration and licensing. Your business case may need to include a provision to increase real storage (memory) requirements of DB2 10.

During the process, keep in mind that the financial benefits of the DB2 10 CPU savings are closely related to software pricing. This is the reason why understanding *why, how, and where performance matters* in your organization is so important in this process.

DB2 10 savings and software pricing

How the DB2 10 CPU reduction will impact your TCO is closely related to the IBM System z software pricing model in use for your organization. Software pricing is a complex topic; do not hesitate to ask for expert advice. The best approach is to consult your IBM representative on pricing matters. Immediately available details can also be consulted on the IBM page “IBM System z Software Pricing” at *http://www.ibm.com/systems/z/resources/swprice*.

In case you are running non-IBM software, you need to contact your independent software vendors (ISVs) to obtain information about how ISV

software charges relate to the CPU utilization for your organization, as well as to obtain information about a possible mandatory upgrade of the ISV software before moving to DB2 10.

DB2 10's potential to impact your total cost of ownership is related to your pricing model.

The System z Software Pricing is the frame that defines the pricing and the licensing terms and conditions for IBM software that runs in a mainframe environment.

An IBM Customer Agreement (ICA) contract is the frame for the *Monthly License Charge (MLC)*, which includes license fees and support costs that apply to IBM software products such as z/OS, OS/390®, DB2, CICS, IMS, and WebSphere MQ.

Software-related costs are measured by MLC pricing metrics such as:

- Advanced Workload License Charges (AWLC)
- Advanced Entry Workload License Charges (AWLC)
- Workload License Charges (WLC)
- Entry Workload License Charges (EWLC)
- Midrange Workload License Charges (MWLC)
- System z New Application License Charges (zNALC)
- zSeries® Entry License Charges (zELC)
- Parallel Sysplex License Charges (PSLC)

The MLC pricing metric is based on customer choice and/or the mainframe environment. For the purpose of this paper, it is important to focus on understanding what MLC pricing metric is of application in your organization to be able to effectively model what would be the impact of a given CPU reduction in your Monthly License Charge.

Understanding how the type of MLC metric works will enable you to model the Monthly License charges applicable to MLC products, such as z/OS, z/TPF, z/VSE®, middleware, compilers, and selected systems management tools and utilities. Not all the IBM software running on your mainframe is necessarily an MLC product.

The case of the Advanced Workload License Charges

Let's consider the *Advanced Workload License Charges (AWLC)* monthly license pricing metric. This metric can be applied to a standalone environment or in a Parallel Sysplex environment, IBM eEnterprise 196 (z196) servers. Among other benefits, the AWLC pricing metric enables organizations to *manage software cost by managing workload utilization.*

Two types of charges are involved when working on AWLC:

- **Advanced Workload License Charges (AWLC):** These are variable charges based on server capacity and/or utilization. This metric applies to products such as z/OS, DB2, IMS, CICS, WebSphere MQ, and Lotus® Domino®.
- **Flat Workload License Charges (FWLC)**: This is a flat charge per server that applies to legacy products such as less current compilers and older MVS™/VM/VSE utilities.

Organizations working with Monthly License Charges metrics based on CPU utilization can benefit from immediate monthly license charges reductions after migrating to DB2 10 as a consequence of the DB2 10 CPU reduction.

The benefits of DB2 10 CPU reductions will also apply to other eligible IBM products such as z/OS, IMS, CICS, WebSphere MQ, and Lotus Domino.

In consequence, it is important to consider the license software impact on all software whose license is related to CPU utilization.

Your organization can implement AWLC in one of two ways:

- **Sub-Capacity AWLC:** In this case, charges are based on the utilization of the LPAR[6] or LPARs where an AWLC product executes.
- **Full-Capacity AWLC:** Under this option, charges are based on the full z196 server capacity where each AWLC product executes.

DB2 10 performance improvements can help to reduce the TCO on both Sub-Capacity and Full-Capacity pricing models.

How DB2 10 performance improvements help to reduce TCO on Sub-Capacity pricing models is discussed in the following sections.

Estimating the DB2 10 financial benefits

Under Sub-Capacity workload license metrics, such as AWLC or WLC, the software charges are calculated based on the 4-hour rolling average CPU utilization per z/OS LPAR observed within a one-month reporting period[7]. This information is obtained by the IBM supplied Sub-Capacity Reporting Tool (SCRT) after processing of the related System Management Facilities (SMF) records[8].

The SCRT tool determines which Sub-Capacity eligible products are executing in each LPAR and the rolling 4-hour average utilization of each one on an hourly basis.

The CPU utilization and the effective 4-hour rolling average CPU utilization per interval are stored in or can be derived from the SMF records using common capacity management tools. If your company is working on workload-based licensed metrics, this information might already be available for your analysis.

To illustrate the relation between the CPU utilization and the 4-hour rolling average, let's consider an example derived from a real customer. Figure 2.11 is a representation of the CPU utilization per hour in a typical business day for a financial institution. The black line in the figure illustrates the 4-hour rolling average for this z/OS logical partition as stored in the SMF records. CPU utilization is measured in millions of service units, or MSUs[9].

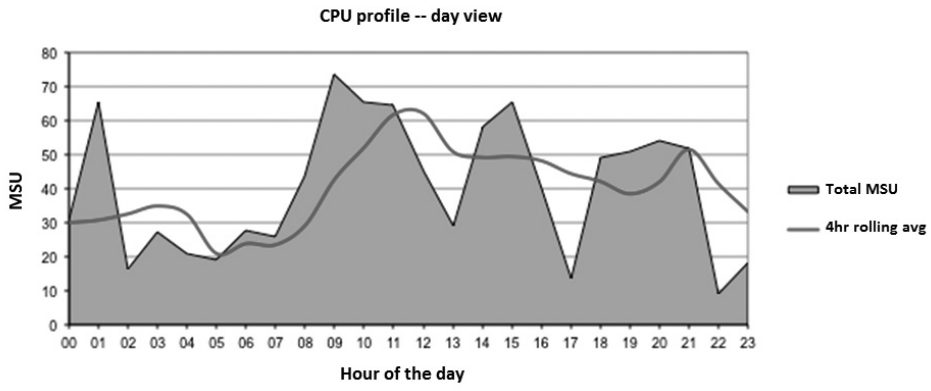

Figure 2.11: Daily CPU profile and 4-hour rolling average

The workload represented in the figure follows the common pattern of many other businesses, and we can identify two distinct workload patterns in the graph:

- During the business hours, typically from 8 a.m. to 5 p.m., most of the workload is executed via transactions. This type of workload is commonly referred to as online transaction processing, or OLTP for short. Many short transactions or business units of work that are typically CPU-intensive characterize an OLTP workload. This period of time is also commonly referred to as *prime shift.*
- Out of the business hours—in this example, from 5 p.m. to 8 a.m.—most of the housekeeping jobs are executed. These jobs or *batches* can be long-running processes designed to process or maintain large amounts of data. This period of time is also commonly referred to as *night shift* or *batch window*.

Figure 2.12 represents the same period, putting emphasis on the peak 4-hour rolling average for this day. The dashed line indicates this peak.

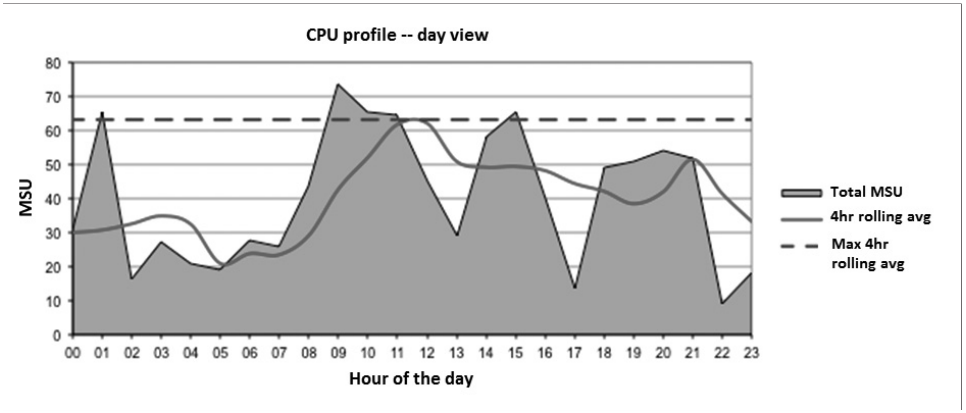

Figure 2.12: Daily CPU profile and peak 4-hour rolling average

Assuming that this figure represents the day that showed the highest 4-hour rolling average of the month, the dashed line is the MSU value to be used for calculating the Sub-Capacity software charges. As discussed, this is true not only for DB2 but also for all the other IBM software where Monthly License Charges are applicable.

In this particular example, the peak MSU occurred at 11 a.m. with an approximate value of 63 MSU.

In a Sub-Capacity license charges model, a reduction in the monthly peak 4-hour rolling average means a reduction of software-related costs.

DB2 10 provides out-of-the-box CPU savings for OLTP and batch workloads. By moving to New Function Mode and investing some efforts in exploiting all the new DB2 10 possibilities, you can achieve further savings.

DB2 10 has the potential to provide immediate (on a monthly basis) license-related cost savings by reducing the monthly peak 4-hour rolling average.

The challenge, then, is to accurately model the DB2 10 performance benefits effect on the peak 4-hour rolling average.

By combining the SMF and the DB2 Accounting and Statistics[10] records, it is possible to represent in a graph what percentage of the total CPU utilization is consumed by DB2-related processes. During this exercise, it is important to also consider the CPU used by the DB2 address spaces themselves.

The DB2 CPU used by applications can be obtained from the DB2 Accounting records. The DB2 CPU used by the DB2 address spaces can be extracted from the DB2 Statistics records. The total DB2 utilization is then calculated as follows:

$$DB2_{CPU} \approx DB2_{CPU\_APPLICATIONS} + DB2_{CPU\_ADDRESS\_SPACES}$$

Once you have calculated the total DB2 CPU, you can calculate the non-DB2 CPU using subtraction, as follows:

$$Non\_DB2_{CPU} \approx Total_{CPU} - DB2_{CPU}$$

The availability of specialty engines in the System z server in the LPAR requires special considerations that we discuss later.

Figure 2.13 shows an example of the DB2 CPU and non-DB2 CPU in a graph. The graph illustrates how and when DB2 applications are using CPU.

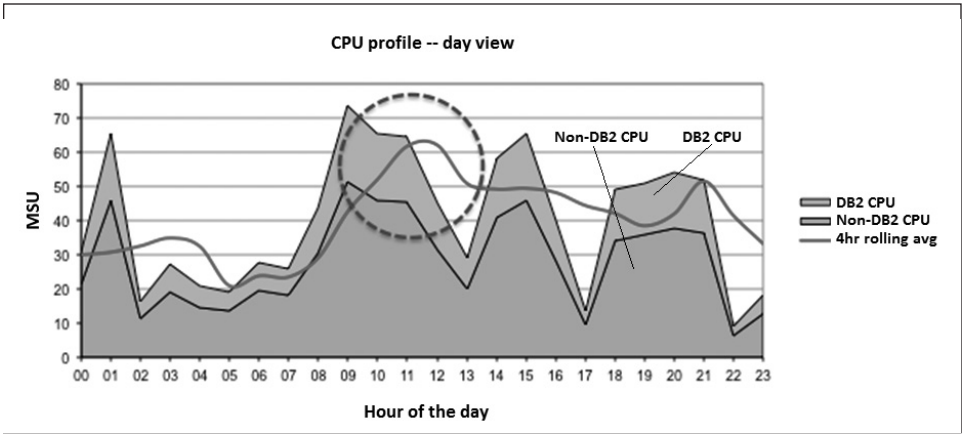

Figure 2.13: Daily CPU profile and DB2-related CPU

You must concentrate on the period that contributes the most to the peak 4-hour rolling average. In the figure, this zone is indicated with a dashed circle. In this particular case, the area is located during the prime shift period, and most of the DB2 activity is due to OLTP transactions. The graph also shows that there is a DB2 CPU-intensive period between 6 p.m. and 9 p.m., but this area is not a contributor to the peak 4-hour rolling average.

As previously discussed, DB2 10 provides CPU improvements for OLTP workloads ranging from 5 percent to 25 percent. Changes for batch processes range from a 5 percent CPU regression to 10 percent improvement.

As an example, after detailed analysis of the OLTP and batch workload, you may consider it conservative to assume a 20 percent CPU reduction for the OLTP workload and a 5 percent CPU improvement for the batch processes.

To model the CPU changes, you can calculate the DB2 CPU after the DB2 10 savings as follows:

$$New\_DB2_{CPU} \approx (DB2_{CPU\_APPLICATIONS} + DB2_{CPU\_ADDRESS\_SPACES}) * (1 - DB2\_10_{SAVINGS})$$

This calculation needs to be done per period and bearing in mind that the DB2 10 savings will change depending on the workload. In this particular case, savings are computed as 0.2 (or 20 percent) for the OLTP period and 0.05 (or 5 percent) for the batch zone.

CPU savings due to DB2 10 can be calculated as:

$$DB2_{CPU\_REDUCTION} \approx (DB2_{CPU\_APPLICATIONS} + DB2_{CPU\_ADDRESS\_SPACES}) * (DB2\_10_{SAVINGS})$$

Because DB2 10 savings apply only to DB2 CPU, the non-DB2 CPU remains the same.

However, the total CPU utilization in the LPAR will be reduced. The new values can be calculated as follows:

$$Total_{CPU} \approx Non\_DB2_{CPU} + New\_DB2_{CPU}$$

Once you have the new total CPU per period, you can just calculate the resulting 4-hour rolling average based on this new LPAR CPU utilization. To be accurate, the 4-hour rolling average must be calculated using four single-hour periods.

Figure 2.14 provides a graphical representation of this modeling exercise.

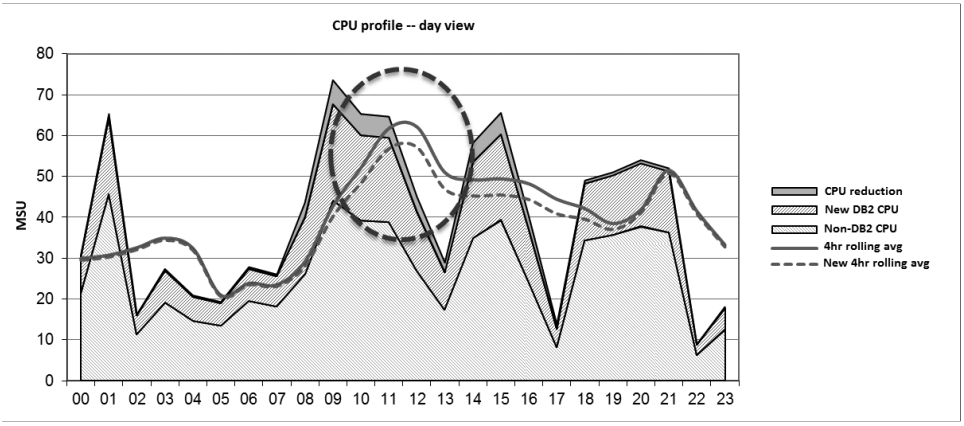

Figure 2.14: Daily CPU profile and DB2 10 CPU savings

For our case, in which only 30 percent of the total CPU in the LPAR was involved in DB2, a CPU reduction of 20 percent due to DB2 10 leads to a reduction of *8 percent of the peak 4-hour rolling average*.

This last reduction will reflect in the monthly license charges. The calculated percentage may or may not be linearly applicable to the eligible software licenses: the unitary MSU price is grouped by cumulative monthly pricing levels as defined in the *Advanced Workload License Charges Structure*. At a glance, the

more MSUs you consume, the cheaper they are. To accurately evaluate the net reduction in costs, you need to apply this structure to your particular case.

Note that the focus must be the peak periods, and you need to concentrate on the DB2 10 performance changes impact on the rolling average. For example, consider a worst-case scenario where batch processes will show a CPU regression of 5 percent when moving to DB2 10; were this to happen in the batch window far from the peak hours, the effect on monthly license charges would be none, provided that the 4-hour rolling average is not impacted.

Multiple Logical Partitions

The previous discussion focused on an environment with a single z/OS LPAR. The common configuration having multiple LPARs requires special considerations when you are modeling software costs.

For example, consider Figure 2.15. The figure results from the observations of a z196 server configured with two LPARs, LPAR A and LPAR B, running on AWLC. It shows the 4-hour rolling average for both LPARs.

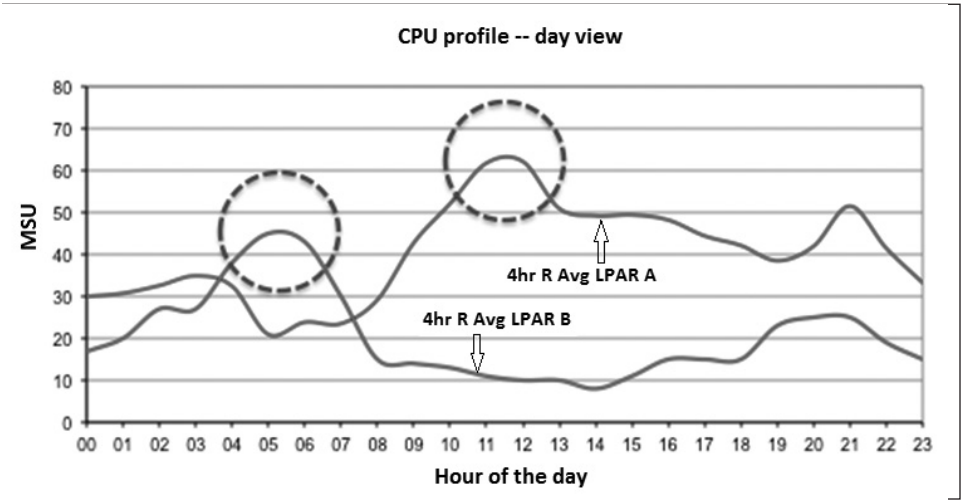

Figure 2.15: CPU average for multiple LPARs

The peak average does not occur at the same time for both LPARs. Table 2.3 summarizes the peak values and the interval at which they are measured.

LPAR name	Max 4-hr rolling average	Interval (hour)
LPAR A	61 MSU	12:00 p.m.
LPAR B	45 MSU	05:00 a.m.

Table 2.3: Peak 4-hour rolling average per LPAR, multiple LPAR example

Based on these values, workload-dependent licensed software running *only* in LPAR A would be charged at 61 MSU. For the same conditions, software running *only* in LPAR B would be charged at 45 MSU.

Pricing for software running in *both* LPARs, such as z/OS and probably DB2, requires the analysis of the combined 4-hour rolling average. This value is obtained by adding the actual averages per hour interval and not the sum of peak averages of each LPAR. Figure 2.16 shows the combined average CPU utilization for this example.

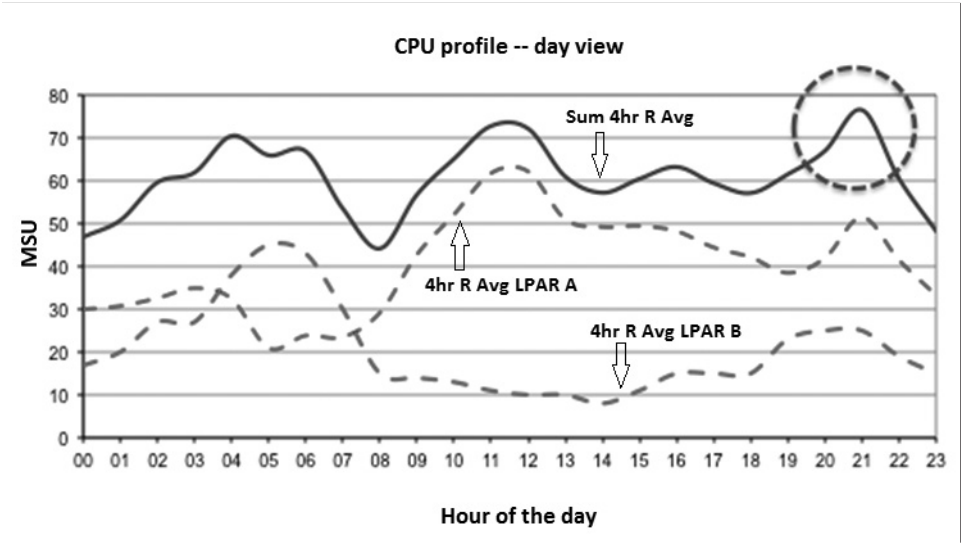

Figure 2.16: Multiple LPARs combined average CPU utilization

The sum of the individual peaks would be the result of adding the peak 61 MSU for LPAR A to the peak 45 MSU for LPAR B, which is 106 MSU. However, the combined average CPU utilization is 76 MSU and happens at the 9 p.m. interval. This result is documented in Table 2.4.

LPAR name	Max 4-hr rolling average	Interval (hour)
Combined LPARS A & B	76 MSU	09:00 pm

Table 2.4: Combined peak 4-hour rolling average after migration

This particular interaction is of interest when modeling the effects of DB2 10 in your CPU utilization. You should model each LPAR change and compute the combined effect to correctly model the pricing effects on software running in both LPARs.

Estimating long-term DB2 10 financial benefits

The previous paragraphs discussed how the DB2 10 performance benefits can reduce the 4-hour rolling average CPU utilization with a potential to reduce TCO. The 4-hour rolling average has the particularity of smoothing peaks and valleys of CPU utilization. It is not a representation of the maximum amount of CPU used. In fact, the CPU used by a workload can go far beyond the 4-hour rolling average at peak time. The maximum CPU consumed is defined by the workload characteristics and by the available CPU capacity in the LPAR. Consider the example in Figure 2.17.

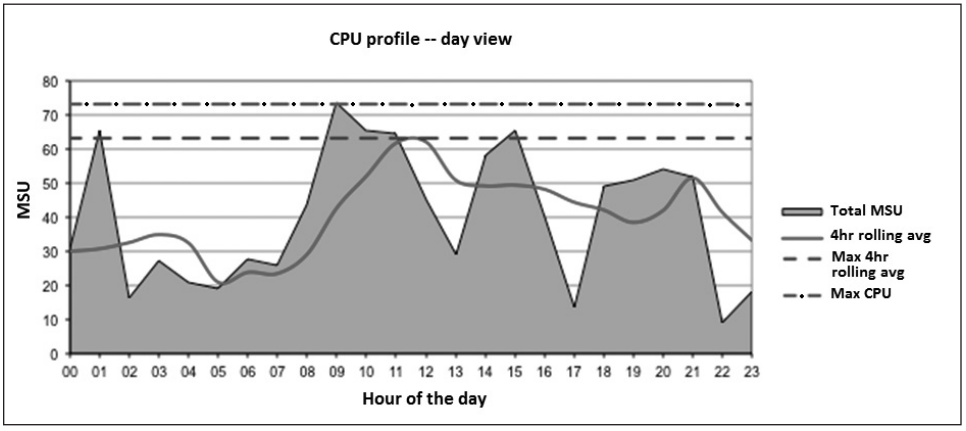

Figure 2.17: Four-hour rolling average and peak CPU utilization

This chart gives an example of CPU profile for a typical day in an OLTP environment. The peak 4-hour rolling average is 62.14 MSU; the peak CPU utilization is 73.57 MSU. The total available capacity in this LPAR is 80 MSU. Table 2.5 documents these values for clarity.

Capacity indicator	Value
Maximum 4-hour rolling average	62.14 MSU
Maximum CPU utilization	73.57 MSU
LPAR capacity	80 MSU

Table 2.5: Key capacity indicators

To determine the degree of utilization of the CPU resources at peak time, the peak CPU is divided by the capacity of the LPAR:

$$Utilization_{PEAK} \approx Peak_{CPU} / Capacity_{CPU}$$

For this particular example, 73.57 MSU / 80 MSU is approximately 90 percent; in other words, the CPU utilization of this LPAR is 90 percent at peak time.

The importance of this performance and capacity indicator resides in the fact that a high level of utilization increases the waits or queuing for a resource. Transactions that execute with a very good response time at lower overall CPU utilization will show degradation in response time when the machine is too busy.

System z and z/OS are designed to work well on servers with a high percentage of CPU utilization. In contrast to other technologies, the mainframe can provide a very good level of service with an LPAR running at high utilization.

The tolerable percentage of CPU utilization is dependent on the workload characteristics. For example, read-only environments can perform very well with CPU utilization up to 98 percent. Workloads involving database changes will suffer at lower utilizations because of elongated locking delays due to wait for CPU conditions.

z/OS Workload Manager (WLM) provides the capacity to distribute resources depending on priorities and goals achievement. This allows organizations to assign a process's business importance into CPU dispatching priorities. The important work gets CPU first.

However, all the tasks need to be executed sooner or later, and if the entire workload is important, there is nowhere from which to take CPU. Even if WLM is a state-of-the-art technology with capacities unknown in other platforms, you cannot solve a capacity problem by solely relying on WLM.

An LPAR having a capacity problem means that the available CPU is not enough to ensure overall good performance. Under capacity constraints, important workload starts to show degraded unacceptable time or insufficient throughput.

The 4-hour rolling average defines pricing. The peak CPU utilization that an application is allowed to reach defines its capacity to deliver an acceptable response time.

What is an acceptable response time? It depends on the application Service Level Agreement (SLA). The SLA defines the performance expectations for an application and should be the basis for the definition of good or bad performance. It can be measured like response time, in seconds or number of transaction per unit of time.

In many cases where an SLA cannot be respected, and after optimization of the involved processes, the only alternative is to increase the CPU capacity. This can be achieved in several ways, including an upgrade of the System z CPU model or buying a newer more powerful server.

In most of the cases, an increase in capacity has a direct or indirect impact on the TCO.

The performance benefits of DB2 10 can delay or avoid the need for an increase in CPU capacity or mainframe model upgrade by reducing the CPU requirements at peak time.

Figure 2.18 gives an example of CPU profile for a typical day, highlighting the peak CPU utilization. It also highlights the CPU savings expected after migrating to DB2 10.

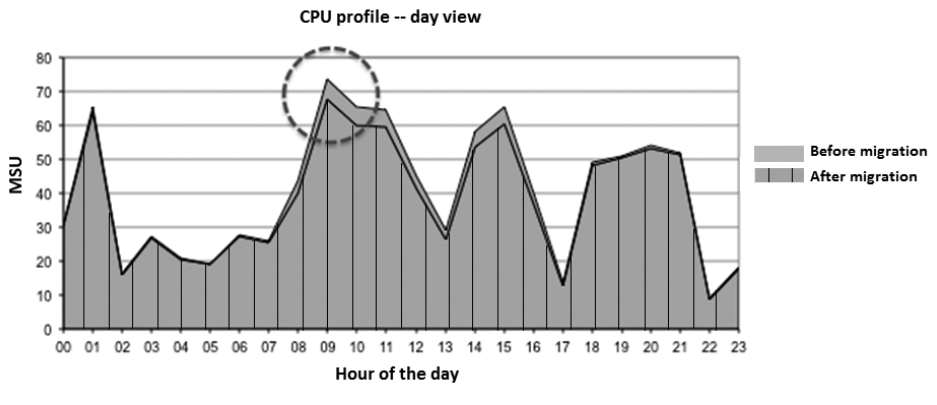

Figure 2.18: Peak CPU requirement before and after migration to DB2 10

The overall estimated LPAR savings is of about 8 percent. Table 2.6 shows the estimation in numbers.

Scenario	Max CPU utilization	Interval (hour)
Before migration	73.57 MSU	09:00 p.m.
After migration	67.69 MSU	09:00 a.m.
Savings:	5.88 MSU (≈ 8%)	

Table 2.6: Estimated LPAR overall CPU savings after migration to DB2 10

Immediately after migration to DB2 10, the maximum CPU utilization drops from 91 percent to 84 percent. Table 2.7 shows the details.

Scenario	Max CPU utilization	Utilization at peak
Before migration	73.57 MSU	≈91%
After migration	67.69 MSU	≈84%
Savings:	5.88 MSU (≈ 8%)	≈7%

Table 2.7: Estimated LPAR CPU utilization at peak after migration to DB2 10

The reduction in utilization at peak time could be considered as a virtual increase of capacity. Because of the many performance improvements, workloads can improve in CPU and also use less clock time. This can also be true for any task in the system, including those that are not using DB2 at all because the DB2 workload claims less CPU.

Another interesting consequence is the potential to delay the need for an increase in capacity or upgrade. For example, assuming that the natural business growth requires a 4 percent yearly increase on CPU at LPAR level, *the estimated 8 percent overall CPU reduction will delay an upgrade for two years*.

Consider the information in Figure 2.19.

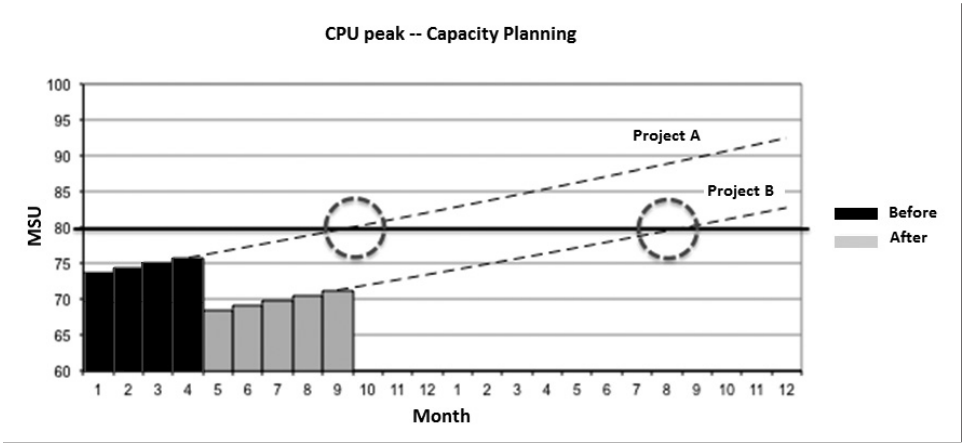

Figure 2.19: CPU projection before and after migrating to DB2 10

This chart shows the peak CPU utilization per month before and after migrating to DB2 10. The chart also includes the increase in CPU requirements due to normal business growth. The capacity of this LPAR is 80 MSU, as represented by the bold line. The dashed lines are projections of the CPU requirements for both scenarios. The intersection of these lines with the 80 MSU line indicates when the maximum capacity for the LPAR will be reached and a capacity upgrade will be necessary to avoid workload performance degradation.

This example illustrates the potential of DB2 10 CPU savings: by migrating to DB2 10 in month 5, the need for upgrade is delayed by 10 months. In this particular case, the upgrade would be from 80 MSU to 95 MSU. Figure 2.20 illustrates the savings.

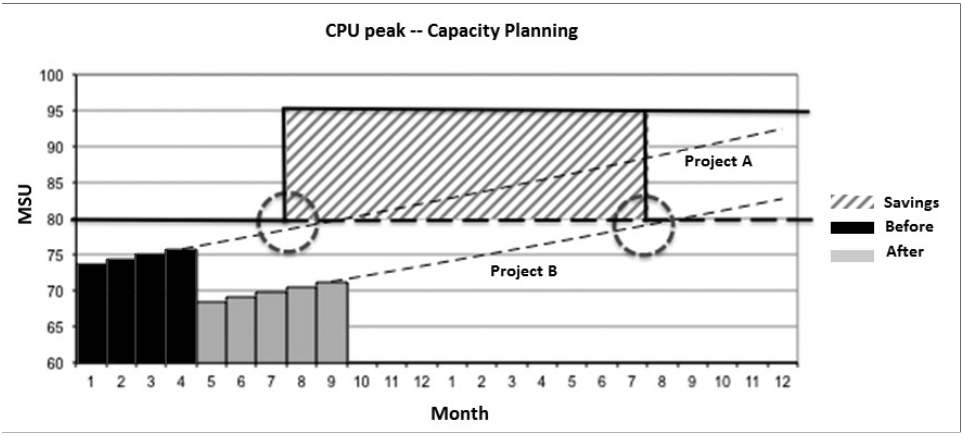

Figure 2.20: CPU savings by delayed upgrade

The savings thanks to a 10-month upgrade delay can be very important depending on how the TCO is related to the LPAR capacity in your organization. For example, many software products are licensed based on the total CPU capacity; if the upgrade implies an update of the System z server, the financial savings can be substantial.

Some Invaluable Performance Benefits of DB2 10

The DB2 10 performance enhancements provide ways to reduce TCO that go far beyond CPU savings only.

Executive summary

DB2 10 provides substantial CPU and performance benefits. You obtain them out of the box or by investing on getting the most of its new and enhanced features.

Many of the changes in DB2 10 provide performance benefits that are not directly or easily quantifiable but have the potential to reduce TCO in the short, medium, or long term. This section covers some of the benefits of DB2 10 that are difficult to quantify but that could bring invaluable benefits to your organization.

While building the DB2 10 business case, be sure to explore all the savings potential applicable to your organization's configuration.

More throughput

When data becomes very volatile, there can be performance inhibitors other than CPU. This situation is often observed during nightly batch jobs. Many organizations are constrained in that they must wait for critical nightly batch processing to end before being able to begin their business day processing (OLTP). Figure 2.21 represents this scenario using our example data.

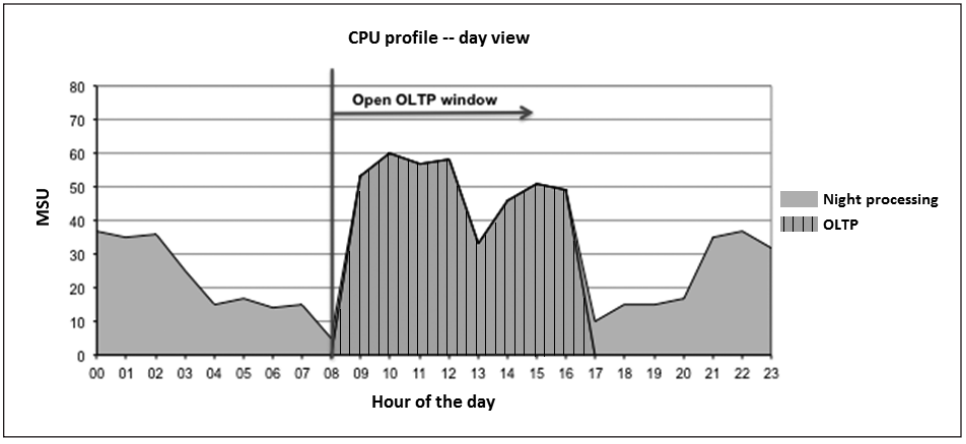

Figure 2.21: Open OLTP window after end of batch

If the end of the batch process is delayed, it means a late start to the daytime activity. Depending on the company and its business activity, this service degradation can have a huge financial impact.

DB2 10 can improve the throughput of data-intensive processes. Changes such as logging enhancements, latching contention relief, dynamic prefetch enhancement, and I/O parallelism for index updates provide better performance, resulting in an improved throughput without the need to change existing applications.

From a performance point of view, mass insert applications need special attention. The DB2 10 insert performance improvements vary, and some of the biggest improvements were observed for high-volume concurrent insert processes in data sharing environments.

Some of the out-of-the-box insert performance-related enhancements include:

- Space search improvement
- Index I/O parallelism
- Log latch contention reduction and faster commit process
- Support for MEMBER CLUSTER in universal table spaces
- LRSN spin loop avoidance

MEMBER CLUSTER support for UTS is available in DB2 10 NFM. This feature can help to dramatically improve insert performance in data sharing environments.

As an example, Figure 2.22 illustrates some of the insert performance results observed on a high-insert benchmark workload in a two-way data-sharing environment.

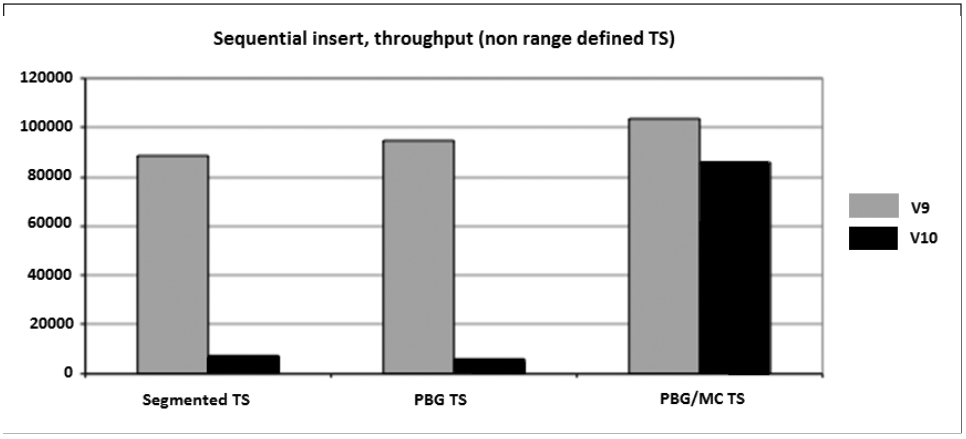

Figure 2.22: Sequential insert improvements

All of these improvements have the potential to reduce the elapsed time required by batch processing. Batch jobs might end sooner after migration to DB2 10. Figure 2.23 shows the time saved after migration for our example.

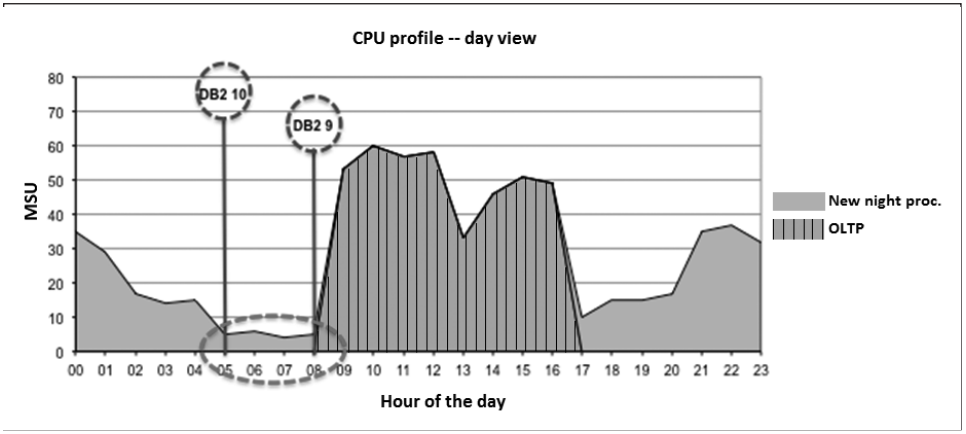

Figure 2.23: Increase of end of batch – start OLTP buffer

Having the overnight processing end sooner provides an additional buffer before the opening of the OLTP window. This extra buffer can sometimes compensate for unexpected peaks in batch processing, seasonal treatments, or operational problems during night processing.

The financial value of this extra buffer is difficult to measure in numbers. Nevertheless, it can help to avoid service disruptions and delays, which are quantifiable depending on the activity. Also, the extra buffer provides a better Quality of Service, increasing customer satisfaction (something that is priceless).

DB2 10 storage and savings by consolidation

With a high level of simplification, the storage used in DB2 could be divided into below and above the 2 GB bar. The memory used below the 2 GB bar (BTB) imposes a limit on the number of concurrent threads, depending on the workload characteristics, to a few hundred. Although the last versions of DB2 had improvements in this area, virtual storage BTB remains the most common constraint. This constraint is commonly known as *Virtual Storage Constraint (VSC)*.

DB2 data sharing is used to provide high availability and concurrent access to shared data without application changes. Also, it is frequently used to solve the scalability problems associated with virtual storage constraints. Many organizations run large data sharing environments to support a high number of concurrent users. Several DB2 data sharing members are often running in the same LPAR. Figure 2.24 shows an example of this configuration.

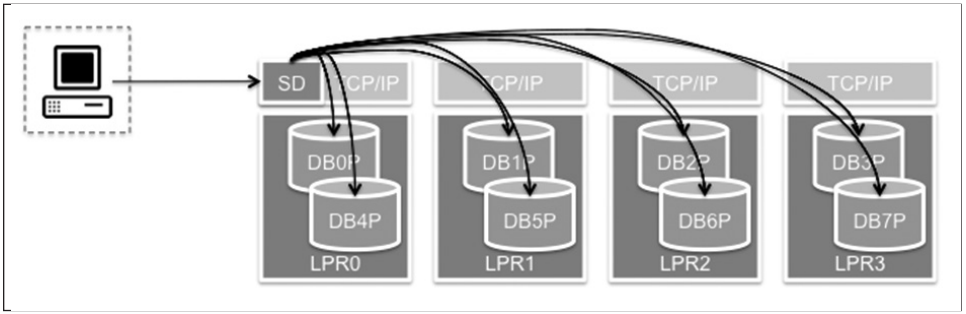

Figure 2.24: DB2 data sharing member co-location

DB2 10 for z/OS provides a quantum leap forward in scalability by moving a large portion—from 50 percent to 90 percent—of the BTB storage to 64-bit virtual storage. These benefits are available to dynamic SQL immediately after migrating to DB2 10. Static SQL using packages will benefit after a REBIND.

Figure 2.25 shows an example of improvements for a distributed benchmark application using dynamic SQL. It highlights the results obtained for 500 users. This is a typically considered safe value for the maximum number of threads allowed to execute simultaneously in many installations running DB2 versions previous to 10.

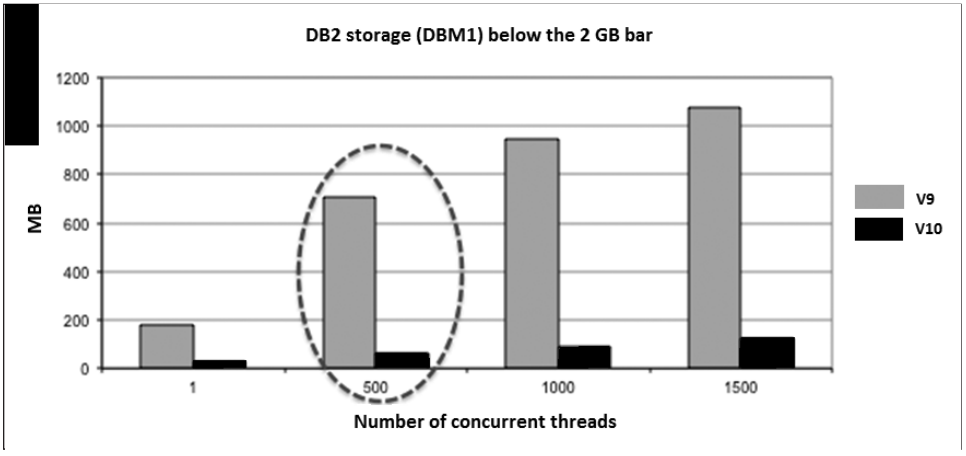

Figure 2.25: DB2 10 storage relief for dynamic SQL

The DB2 10 memory changes substantially increase the number of concurrent threads that can be supported by a single DB2 subsystem. DB2 10 has the potential to support five to 10 times more concurrent users and up to 20,000 concurrent users in a single DB2 subsystem. Organizations running DB2 9 and experiencing virtual storage constraints can consider consolidating DB2 subsystems after moving to DB2 10. DB2 10 also improves in other areas, making possible running more work in a single DB2. These changes include enhancements in the logging subsystem and latching contention relief.

The infrastructure in Figure 2.24 can be simplified as shown in Figure 2.26.

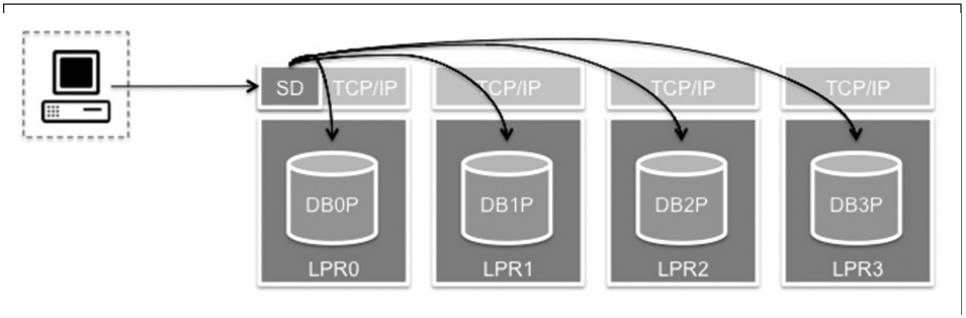

Figure 2.26: DB2 data sharing member consolidation

Data sharing member consolidation provides a simpler infrastructure and has the potential to reduce administration and maintenance costs. In addition, organizations may observe 0.5 percent CPU savings for each member removed from the group, and more saving on real storage.

In complex installations including multiple LPARs, there is a potential for even further consolidation. Consider, for example, the configuration in Figure 2.27. It shows a four-LPAR, eight-member DB2 data sharing group that consolidated from a 16-member data sharing group after migrating to DB2 10.

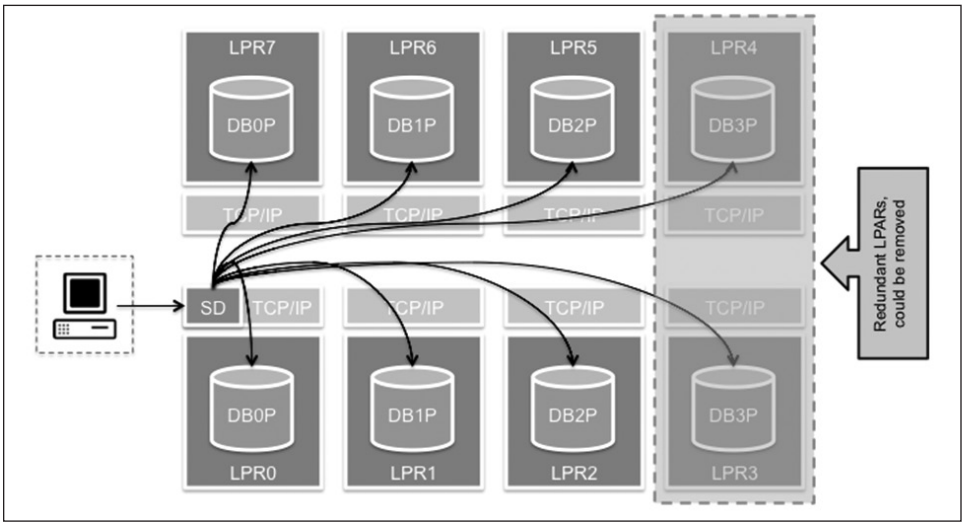

Figure 2.27: Logical partition consolidation

The combination of the CPU reduction and memory enhancements provided by DB2 10 makes it possible to consider an LPAR consolidation scenario. In the preceding example, the last two LPARs could be removed and their workload transferred to the remaining partitions. This scenario has the potential to further reduce TCO.

During a consolidation analysis, remember that DB2 data sharing is a high-availability solution. Multiple DB2 members, in combination with DB2 Sysplex Support, provide applications with seamless high availability and workload balancing. From an availability point of view, the ideal minimum number of DB2 data sharing members is four, with two at a remote location.

Specialty engines

A System z specialty engine is a special incarnation of a z processor designed to execute selected eligible workloads and to help reduce the TCO of the z platform. There is an acquisition cost per specialty engine; however, the CPU executed in them is not accounted for in license charges.

At the moment of this writing, four specialty engines are currently available:

- System z Integrated Information Processor (zIIP)
- System z Application Assist Processor (zAAP)
- Integrated Facility for Linux® (IFL)
- System Assist Processor (SAP)

The most interesting one from a DB2 point of view is certainly the zIIP processor. This specialty engine lets eligible DB2 workload be offloaded and helps reduce TCO. DB2 zIIP support was introduced with DB2 8.

DB2 10 extends the scope of the zIIP-eligible workload, increasing the potential to reduce TCO.

Some DB2 10 enhancements in this area are:

- The ability to offload 100 percent of prefetch and deferred write engines. This enhancement is significant for index compression and insert index I/O parallelism.
- The ability to offload 99 percent of RUNSTATS CPU (with no additional parameters).
- In z/OS V1R11, DFSORT allows additional zIIP redirect for DB2 utilities.
- Parsing process of XML schema validation:
- 100 percent of the new validation parser is eligible.
- Offload to zIIP, zAAP, or zAAP on zIIP.
- zIIP eligibility for DRDA workloads is increased from 53 percent to 60 percent.
- Certain DBM1 processes.
- Prefetch I/Os (reported as DBM1 SRB).
- Deferred write I/Os (reported as DBM1 SRB).
- Stored procedures written in SQL/PL.

A zIIP processor is not required to evaluate its potential benefits. By exploitation of the projected usage function activated by the z/OS parameter PROJECTCPU, you can identify which portion of the CPU would be executed on a zIIP processor. For example, consider the daily CPU profile for a data warehouse LPAR shown in Figure 2.28. This LPAR does not have zIIP engines.

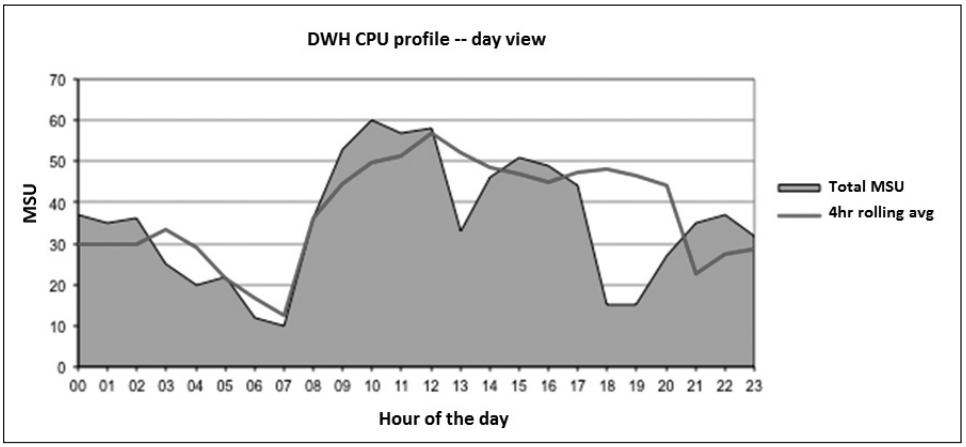

Figure 2.28: Data warehouse daily CPU profile

This chart is built using the SMF Type 72 records (RMF workload activity), and it shows the CPU used by the entire workload, DB2 and non-DB2, running in the LPAR. After activation of the PROJECTCPU z/OS parameter, the same records will contain the details of the specialty engine eligible CPU actually executed in a general-purpose processor.

By subtracting the projected CPU from the total CPU used, you can model the general-purpose CPU utilization if zIIPs with enough capacity were available in the LPAR. This information can be used to calculate a new four-hour rolling average because zIIP CPU does not impact license costs.

Figure 2.29 documents the resulting profile.

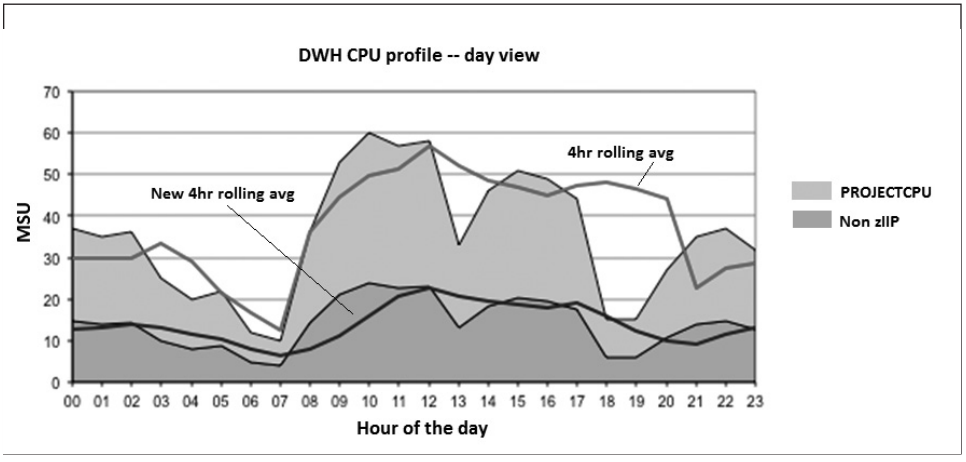

Figure 2.29: Projecting the zIIP-eligible CPU

The potential to reduce TCO is quite evident in this example. Consider that a DB2 data warehouse partition with distributed access to DB2 is most probably the best example for getting benefits from one or more zIIP engines.

z/OS 1.11 added the possibility to execute zAAP-eligible workload on a zIIP processor. The combination of zAAP and zIIP workload on a single specialty engine type makes it possible for organizations to reconsider use of a specialty engine for cases in which the eligible workload was not enough for a financially interesting ROI. The combined workload on a single engine type increases the ratio of benefits to acquisition cost. For example, with the zAAP-on-zIIP capability, all z/OS XML System Services parsing can be executed on available zIIP(s).

Notes

[1] IDUG Market Research Survey. May 2011.

[2] An excellent reference on this technology is the IBM Redbook *System z Parallel Sysplex Best Practices* (SG24-7817).

[3] A. H. Maslow. "A Theory of Human Motivation." Originally published in *Psychological Review*, 50, 370–396, 1943.

[4] IBM Redbook *DB2 10 for z/OS Performance Topics* (SG24-7942).

[5] Ibid.

[6] Logical partition.

[7] You can view a per-interval utilization (in MSUs) and a per-interval rolling 4-hour average utilization (in MSUs) by using the Sub-Capacity Planning Tool, an SMF70 post-processor available for download at *http://www-03.ibm.com/systems/z/resources/swprice/subcap/scpt/index.html*.

[8] System Management Facilities (SMF) collects and records system- and job-related information from a z/OS logical partition (LPAR) for later use.

[9] Million service units, or MSU, is a measurement of the amount of processing work that an IBM System z server is able to deliver in a given period. This unit is commonly used for mainframe-related capacity management and performance engineering. It has the advantage of being hardware-independent.

[10] The DB2 Accounting and Statistics records provide detailed information about the use of DB2 and system resources by applications and by DB2 itself.

Acknowledgements

Thanks a lot to the following people for their invaluable contributions to this paper:

Jeffrey Berger
DB2 Performance,
Senior Software Engineer,
IBM Information Management
USA

Dave Beulke
IBM Gold Consultant & IBM
Champion
USA

Klaas Brant
IBM Gold Consultant and IBM
Champion
The Netherlands

Paolo Bruni
DB2 Information Management Project
Leader at IBM ITSO
USA

Linda K. Martin
Principal, Marlin Consulting
Houston, Texas, USA

Katrin Noack
Belgium Thanks a lot to the
following people for their invaluable
contributions to this paper:

Surekha Parekh
IBM World-wide Marketing
Program Director – DB2 for z/OS
UK

Mark S. Rader
IBM Advanced Technical Skills (ATS)
Americas
USA

Julian Stuhler
IBM Gold Consultant and IBM
Champion
UK

Dino Tonelli
IBM System z Performance Analysis
USA

DB2 10 FOR z/OS

The Smarter, Faster Way to Upgrade

In the current economic climate, businesses are under significant pressure to control costs and increase efficiency to improve their bottom line. DB2 for z/OS customers around the world are still trying to gain competitive advantage by doing more with less: more business insight, more performance, more operational efficiency, more functionality, more productivity with less cost, quicker time to market, and a lower TCO. **With support for DB2 Version 8 scheduled to end in April 2012, there has never been a better time to start planning your DB2 10 upgrade**. Here are the top 10 reasons to start planning today:

1. Improved performance, with reduced software license costs
2. Increased number of concurrent users, by a factor of 10
3. Reduced contention in database administration
4. More administrative capabilities while database is online
5. Improved security and auditing
6. Ability to maintain "snapshots" of changing data – Temporal Data
7. Improved portability via enhanced SQL
8. Enhanced pureXML performance and usability
9. Improved productivity for database/systems administrators and application programmers
10. Better online transaction processing performance – Hash Access

John Campbell is an IBM Distinguished Engineer reporting to the Director for z/OS Development at the IBM Silicon Valley Lab.

Cristian Molaro is an independent DB2 specialist and an IBM Gold Consultant focused on DB2 for z/OS administration and performance.

Surekha Parekh is IBM's World-wide Marketing Program Director – DB2 for z/OS.

MC PRESS

MC Press Online, LLC
P.O. Box 4886
Ketchum, ID 83340-4886

Price: $16.95 US/$18.95 CN

ISBN 978-1-58347-345-0

EAN 9 781583 473450 51695 >