# Gaining the Financial Benefits of DB2 10 for z/OS

## *Why, how, and where performance matters*

*by Cristian Molaro*
*MConsulting*
*Belgium 2011*

# Table of Contents

# 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.

## About the author

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

He was recognized by IBM as an IBM Champion in 2009, 2010 and 2011.

He 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". Cristian is part of the IDUG EMEA Conference Planning Committee.

You can contact him at ***cristian@molaro.be***

## 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?"*

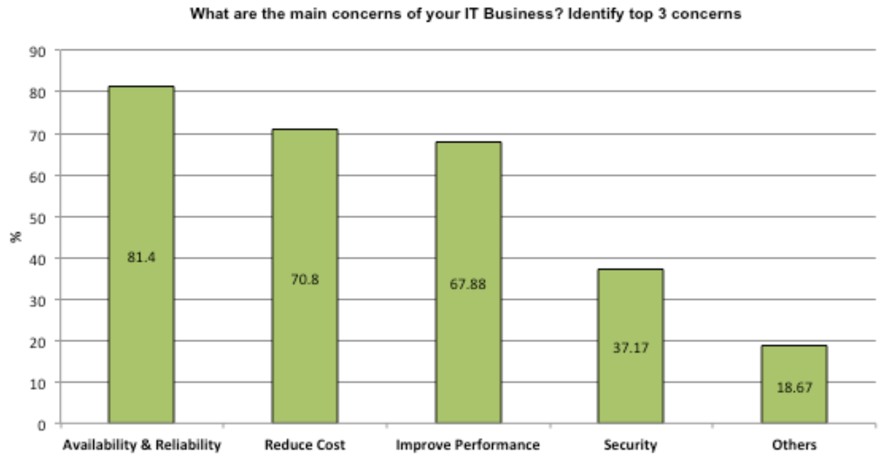


*Figure 2.1: The top concerns of today's IT business*

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*.

*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.

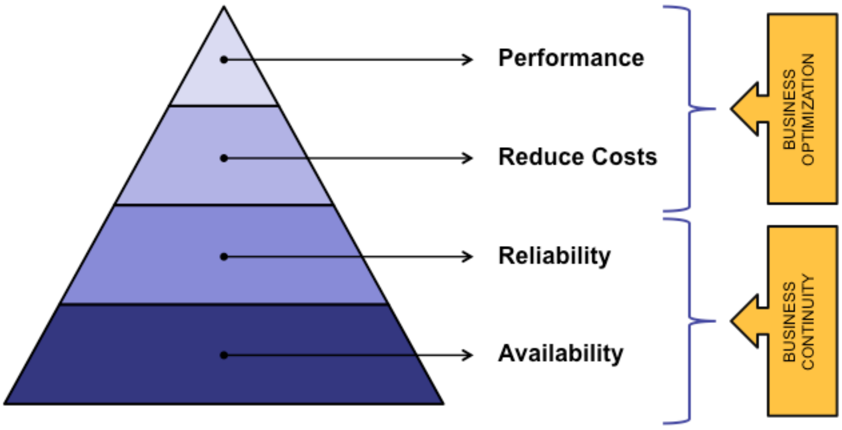


*Figure 2.2: The hierarchy of today's main IT business concerns*

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.

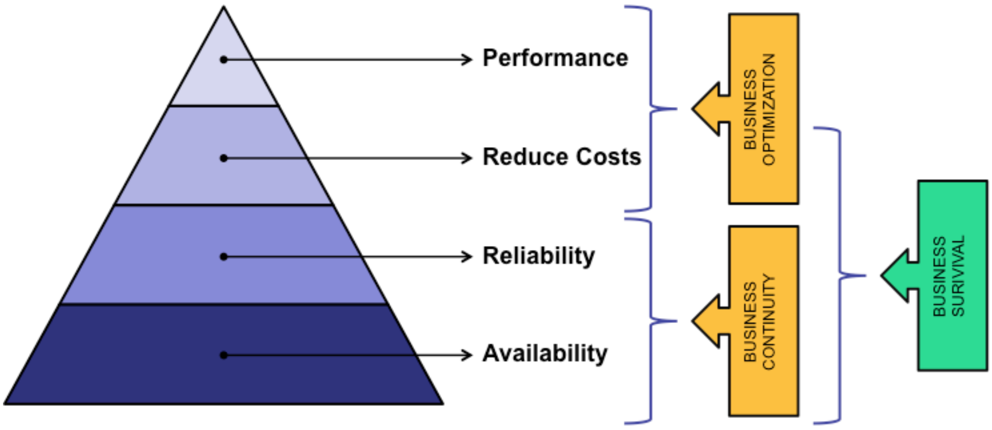


*Figure 2.3: IT business concerns and business survival*

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 offloaded 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.

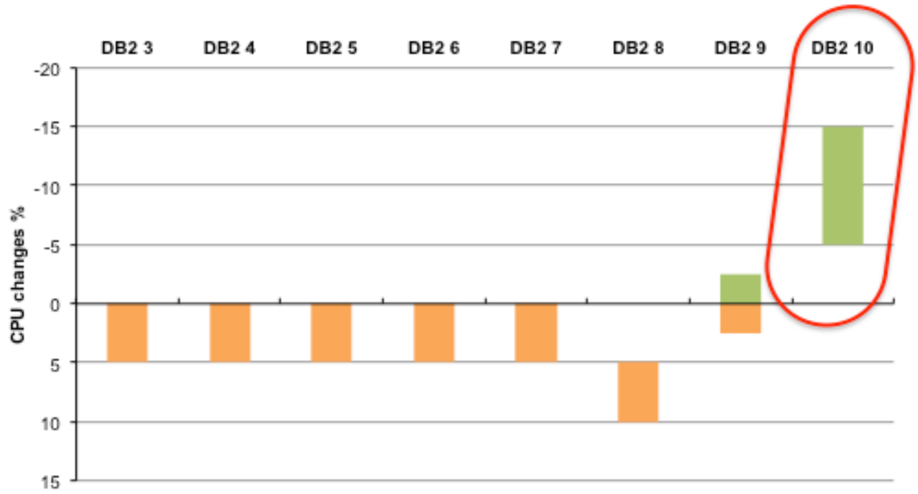


*Figure 2.4: DB2 10 performance objectives*

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.

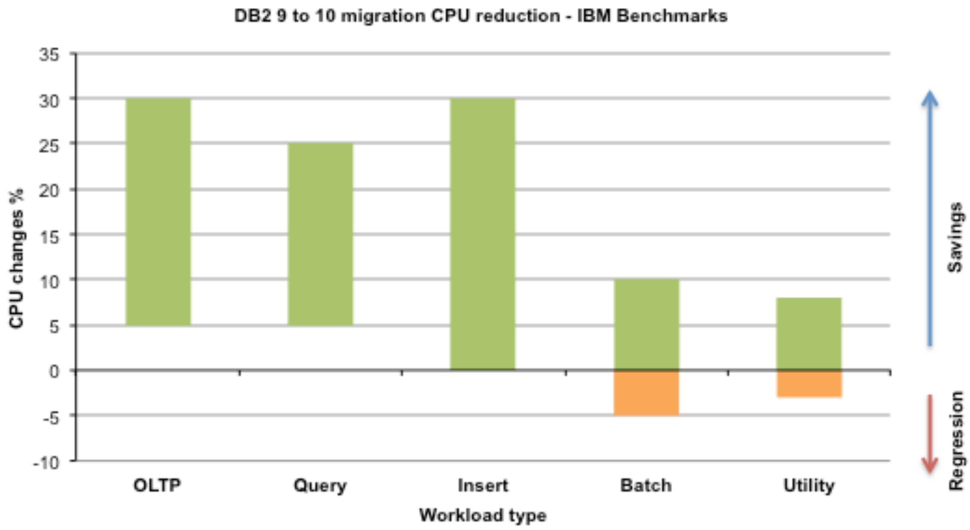


*Figure 2.5: DB2 9 to DB2 10: observed CPU changes*

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.

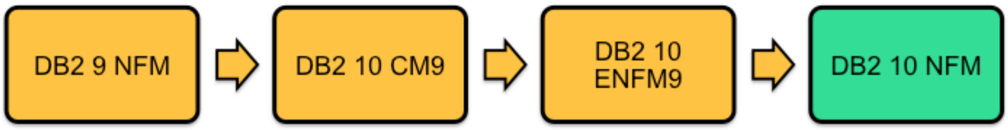


*Figure 2.6: Migration to DB2 10 from DB2 9*

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.

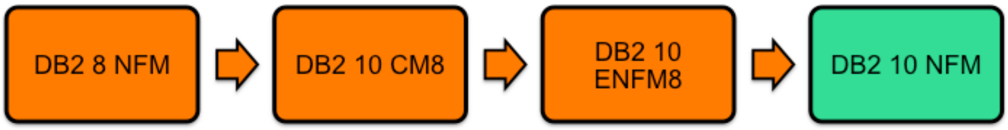


*Figure 2.7: Migrating to DB2 10 from DB2 8, Skip release*

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.

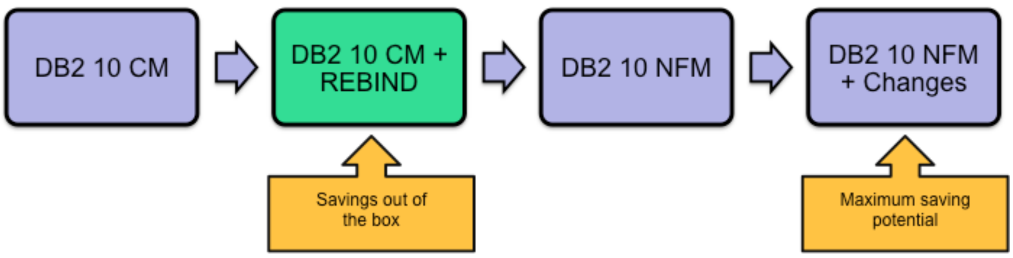


*Figure 2.8: The DB2 10 performance migration path*

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.

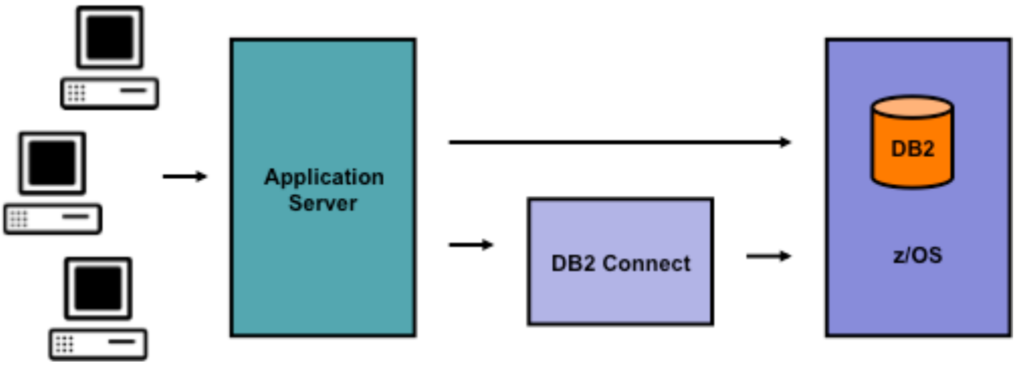


*Figure 2.9: Distributed applications*

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.

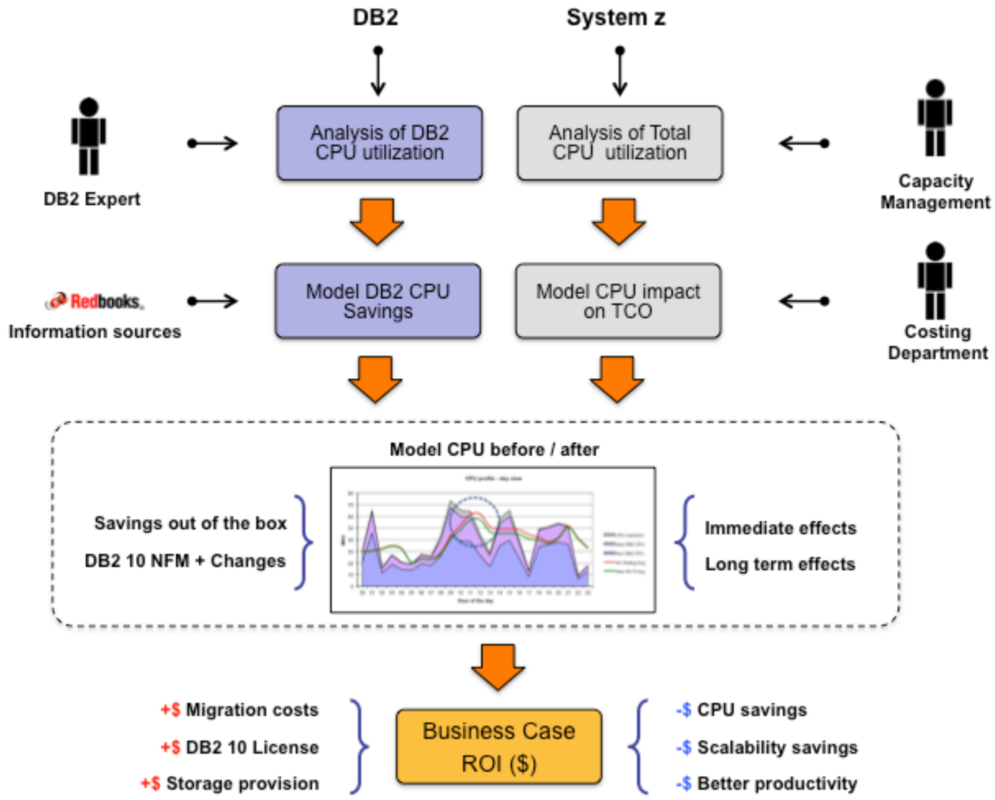


*Figure 2.10: The process of building the DB2 business case*

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 red 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].

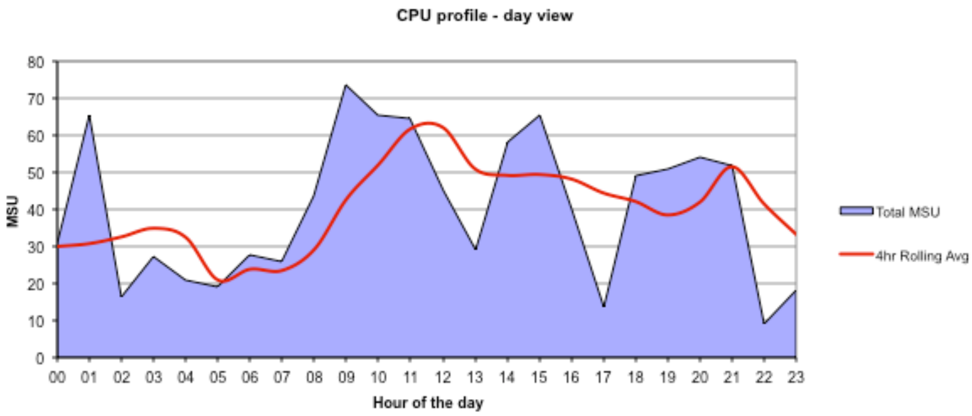


*Figure 2.11: Daily CPU profile and 4-hour rolling average*

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.

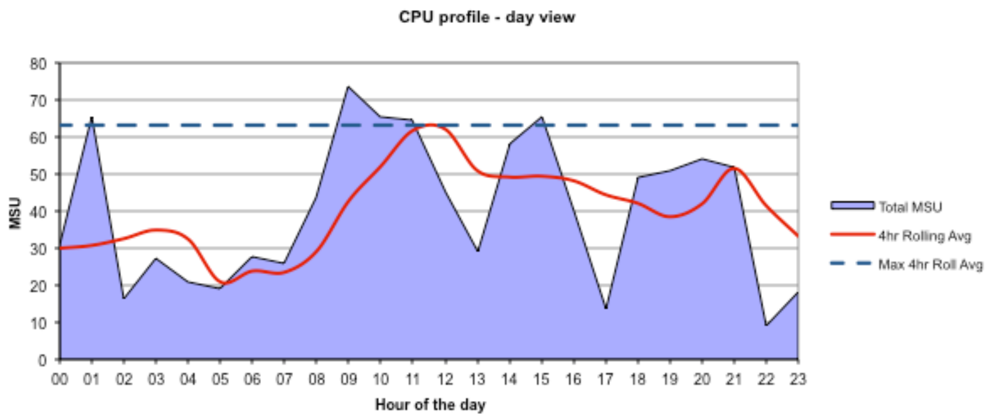


*Figure 2.12: Daily CPU profile and peak 4-hour rolling average*

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.

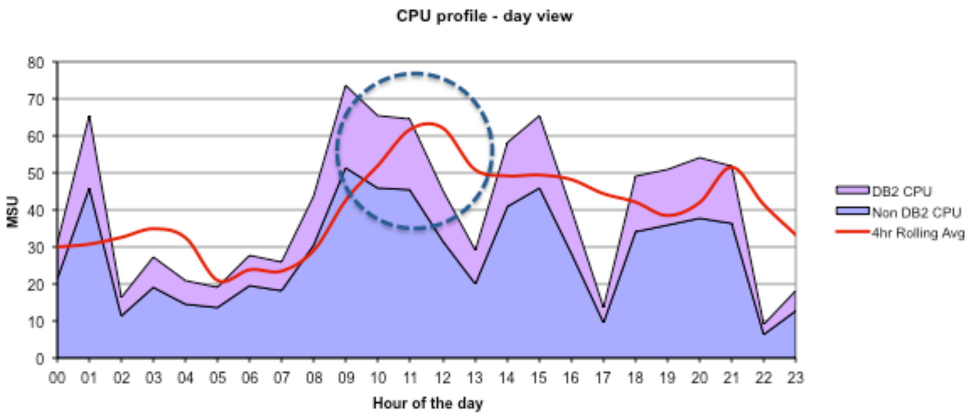


*Figure 2.13: Daily CPU profile and DB2-related CPU*

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.

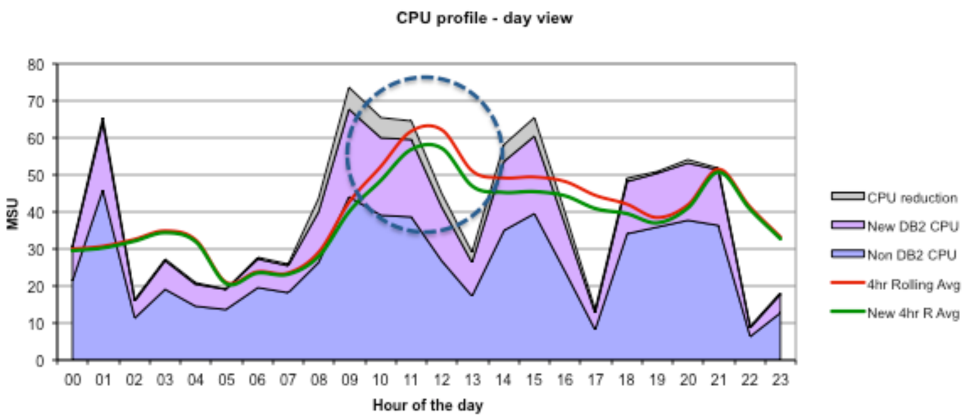


*Figure 2.14: Daily CPU profile and DB2 10 CPU savings*

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.

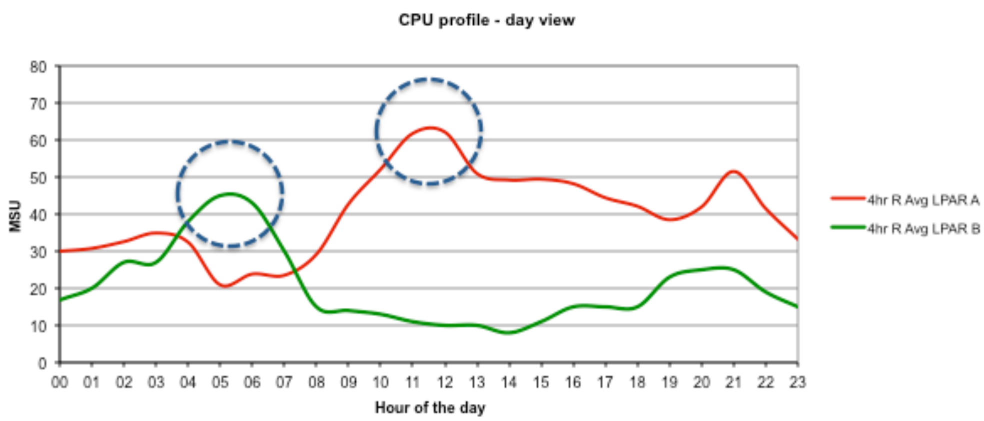


*Figure 2.15: CPU average for multiple LPARs*

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.

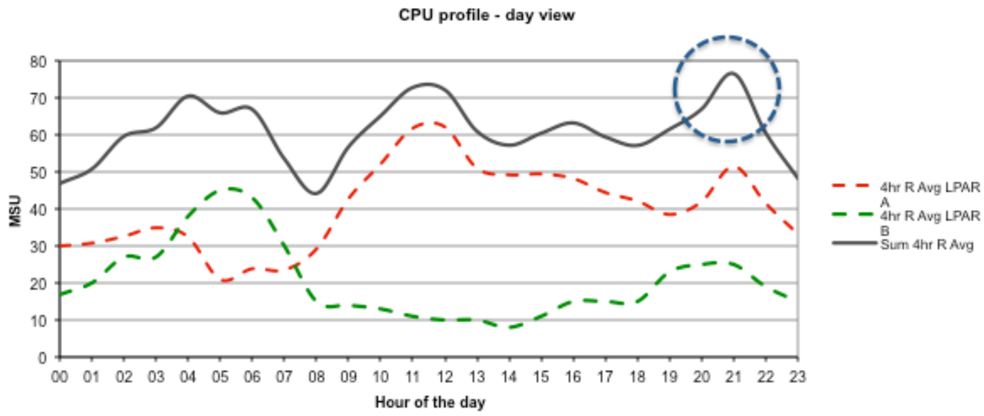


*Figure 2.16: Multiple LPARs combined average CPU utilization*

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.

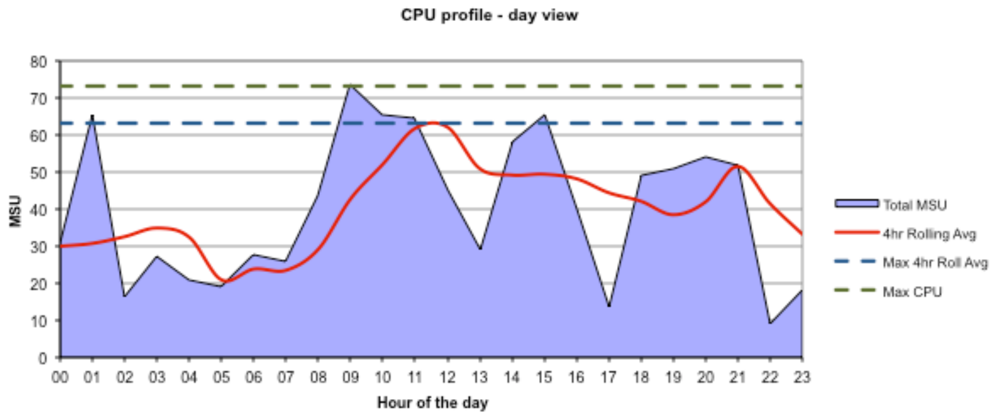


*Figure 2.17: Four-hour rolling average and peak CPU utilization*

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.

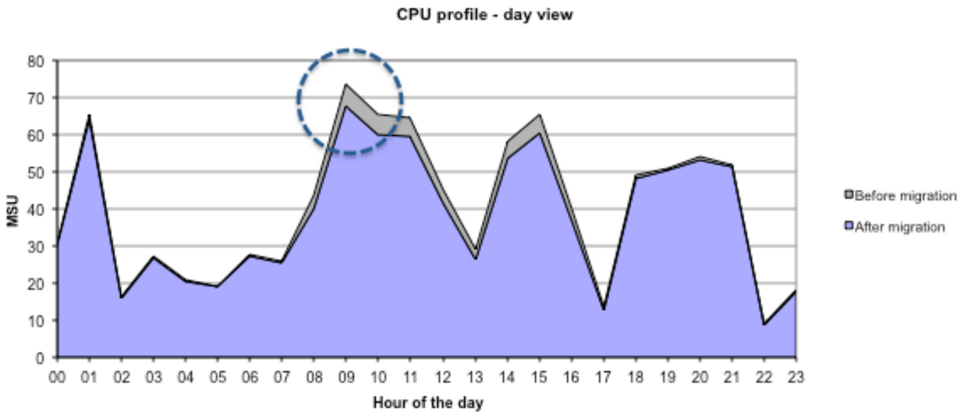


*Figure 2.18: Peak CPU requirement before and after migration to DB2 10*

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.

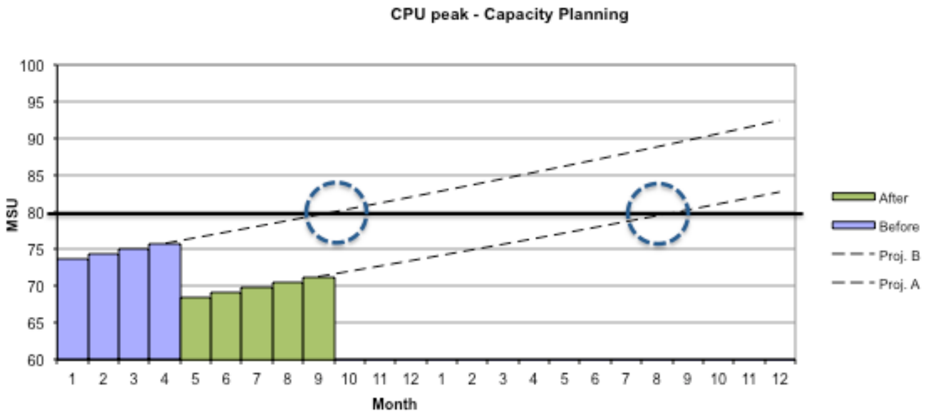


*Figure 2.19: CPU projection before and after migrating to DB2 10*

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.

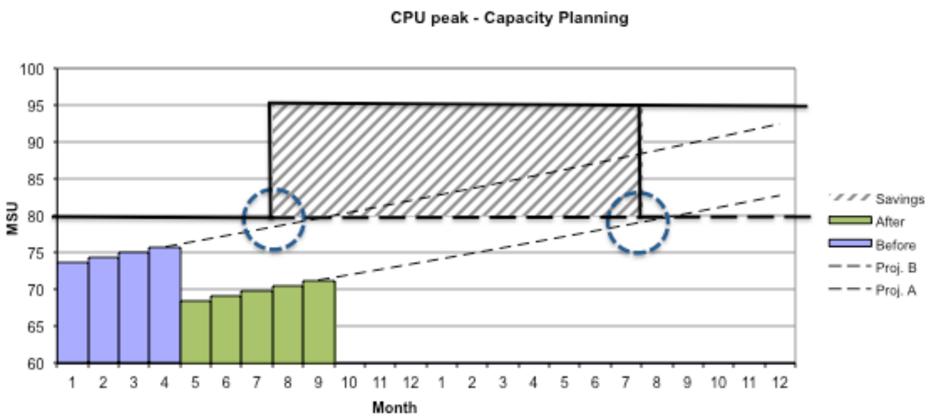


*Figure 2.20: CPU savings by delayed upgrade*

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.

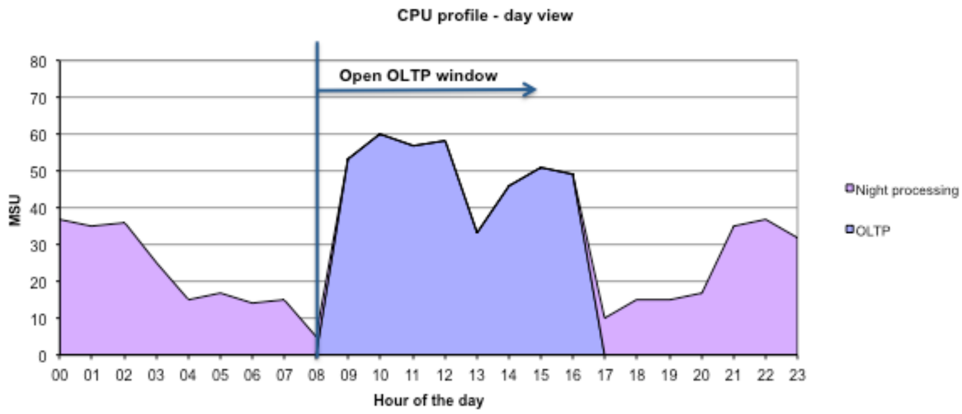


*Figure 2.21: Open OLTP window after end of batch*

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.

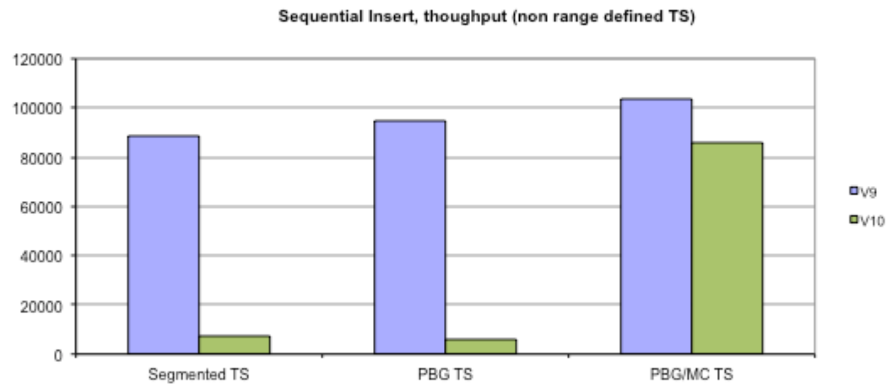


*Figure 2.22: Sequential insert improvements*

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.

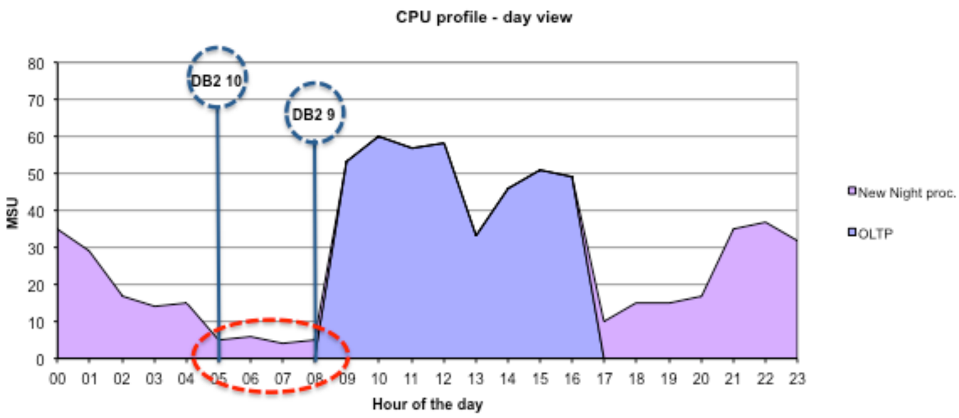


*Figure 2.23: Increase of end of batch – start OLTP buffer*

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.

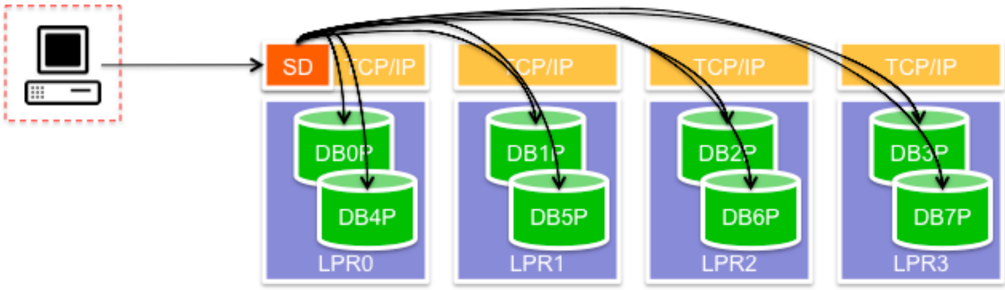


*Figure 2.24: DB2 data sharing member co-location*

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.

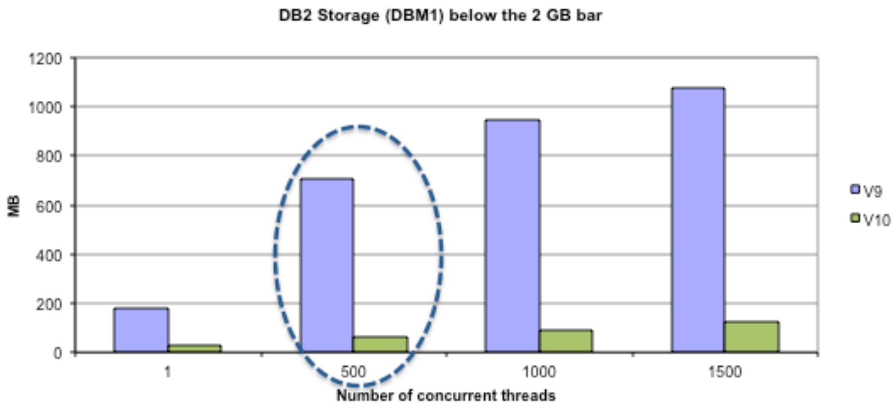


*Figure 2.25: DB2 10 storage relief for dynamic SQL*

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.

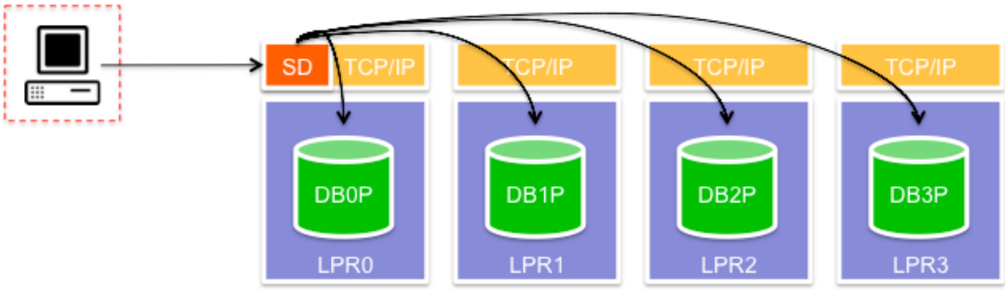


*Figure 2.26: DB2 data sharing member consolidation*

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.

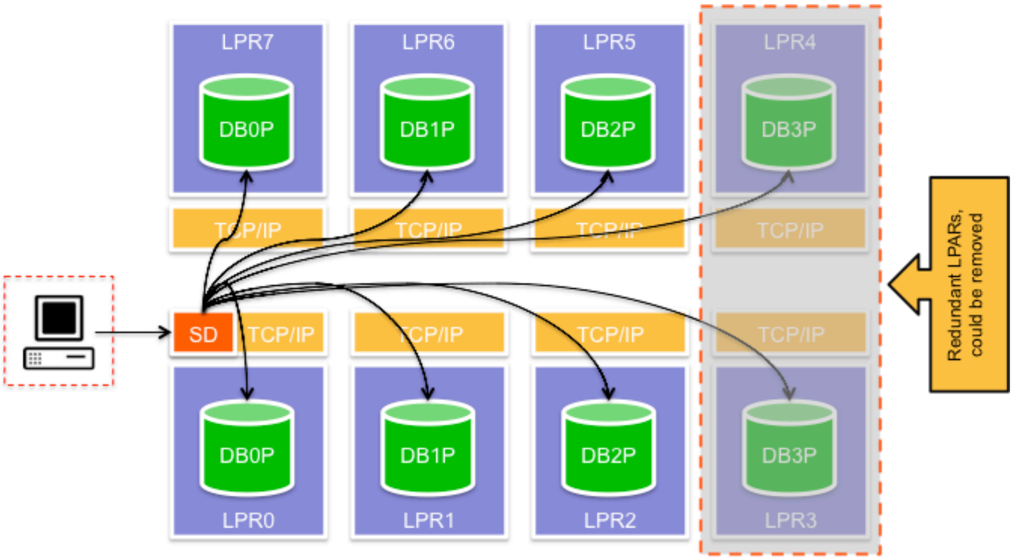


*Figure 2.27: Logical partition consolidation*

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:
  - o 100 percent of the new validation parser is eligible.
  - o 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.

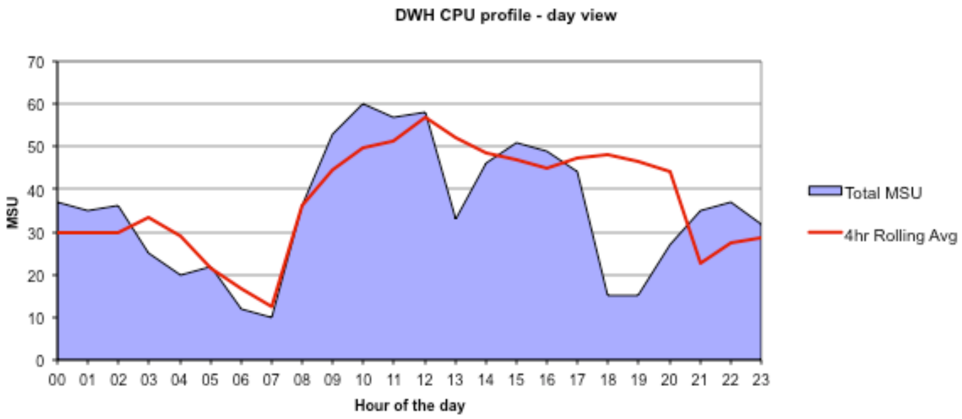


*Figure 2.28: Data warehouse daily CPU profile*

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.

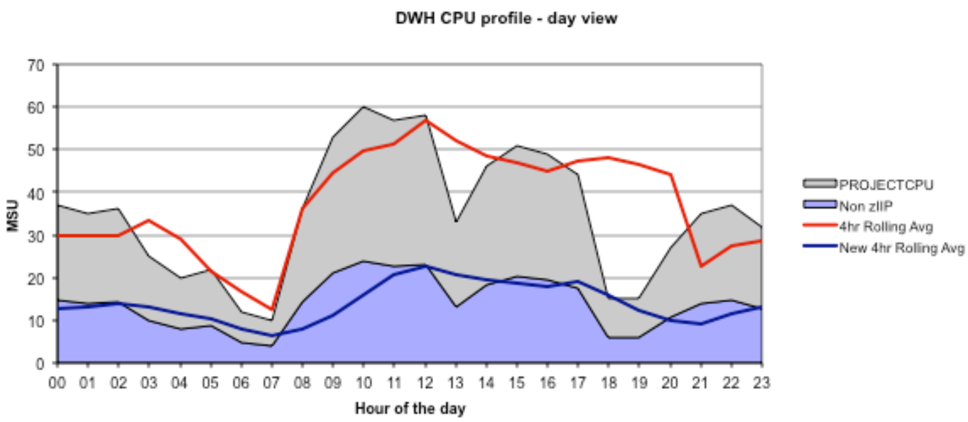

*Figure 2.29: Projecting the zIIP-eligible CPU*

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] Measurements in mSecs.

[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

**Surekha Parekh**
IBM Word-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

# Appendix: DB2 10 and reducing costs: user feedback

This section contains some of DB2 10 user feedback that is relevant to this white paper. For more information, refer to the IBM webpage **Hear what our Customers are Saying about DB2 10 for z/OS** at http://www.ibm.com/software/data/db2/zos/testimonials.html.

*"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 implement 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 to 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% CPU improvement and significant elapse time reduction in direct comparison to DB2 9 for z/OS. For all of our critical tested selects 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 thrilled about 'Temporal Data' feature – this feature has the potential to significantly reduce overheads. We have estimated that 80% 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 importantly make applications easier to understand so improve business efficiency and effectiveness"*

**Frank Petersen – Bankdata System Programmer**