DB2 for z/OS Version 8.1
Driving Business Value

Julian Stuhler
Triton Consulting
August 2004

Table of Contents

Abstract

This paper will provide a high-level overview of the major new features of IBM DB2 UDB for z/OS Version 8.1 from an IT Executive's perspective, with the emphasis on the underlying business value that the new release can deliver. It will also examine the business case for the upgrade from Version 7, including a generic Return On Investment (ROI) model that can be easily adapted to most customer's specific circumstances.

N/B. All references to "Version 7" within this document refer to IBM DB2 UDB for OS/390 Version 7, and references to "Version 8" refer to IBM DB2 UDB for z/OS Version 8.1.

About The Author

Julian Stuhler is a Principal Consultant with Triton Consulting, a UK-based company specialising in the provision of DB2 consultancy, education and managed services to clients throughout Europe. He has over 17 years relational database experience, working for a number of clients within the insurance, telecommunications, banking, financial services and manufacturing sectors.

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

Julian is an IBM DB2 Gold Consultant, and a member of the IDUG Board of Directors. He can be contacted at **julian.stuhler@triton.co.uk**.

Executive Summary

DB2 for z/OS Version 8 is one of the most important releases in the lifetime of IBM's flagship relational database management system. Building upon the solid foundation of Version 7, the new product has been extensively enhanced and re-architected to directly address many of the critical challenges facing IBM's most demanding enterprise customers.

Business Benefit Summary

The following sections summarise the business benefits associated with the move to DB2 Version 8.

Scalability & zSeries Synergy

Feature	Business Benefit	Page
64-bit support	Version 8 allows customers who have invested in IBM's latest generation of hardware to fully exploit the immense storage capabilities of the 64-bit memory model, with no changes being required to their applications. This in turn allows DB2 systems to cache significantly more data and to vertically scale much higher than was previously possible, resulting in a reduction in system overheads and more efficient use of limited machine resources. In some cases, these enhancements may be sufficient to allow customers to defer the implementation of data sharing, if this was being considered purely for scalability reasons. Although customers can expect some increase in CPU requirements following the move to a 64-bit environment, these increases are expected to be offset by the significant performance and productivity enhancements within the new release.	15
Partitioning Enhancements	Version 8 provides some major enhancements to the partitioning options available to DB2 database designers, allowing them to create more efficient designs that are able to support higher transaction loads for a given amount of machine resource. These benefits can be implemented on existing database designs with minimal loss of availability, due to the new dynamic schema change capabilities.	18
Recovery Log Enhancements	Version 8 allows significantly more recovery log data to be held both online and offline, allowing a DB2 system to handle larger workloads while maintaining the recoverability of the data.	19
Other zSeries Synergy	Version 8 exploits specific features in z/OS and the zSeries hardware to deliver highly efficient character conversion and encryption services.	20

Data Availability

Feature	Business Benefit	Page
Dynamic Schema Change	Version 8 provides a good initial set of capabilities for changing database structures while retaining application access to the data. Many customers will be able to use these features to implement changes more frequently, allowing them to react more rapidly to changing market conditions or unexpected performance issues. Database administrator productivity will also be improved (and the risks associated with the change reduced), due to the significantly simpler change process.	21
Partitioning Enhancements	The new partitioning model introduced in Version 8 allows true partition independence for the first time. This reduces the availability impact of table reorganisations, allowing the administrator to keep the data better organised and therefore potentially improving the performance of the application.	23
Dynamic System Parameters	The additional dynamic system parameters in Version 8 decrease the likelihood of the entire DB2 system having to be made unavailable while configuration parameters are changed, which improves overall availability and allows system administrators to be more responsive to changing application demands.	23
System Wide Backup & Recovery	Provided you have the right DASD subsystem, the new system-wide backup and recovery capabilities can reduce the time that DB2 data has to be unavailable due to system backups. Perhaps more importantly, the ability to rapidly restore a system to a given point in time means that applications can be made available again in the minimum possible time in the event of a major problem.	24

Application Porting

Feature	Business Benefit	Page
SQL Enhancements	Together, these SQL enhancements reduce the effort, complexity, risk and cost of porting database definitions and application code to DB2 Version 8, making it an even more attractive platform for vendor application porting, infrastructure consolidation projects and development on Windows, Unix or Linux.	25
DB2 Universal Driver	The new Universal Driver removes yet another set of code changes needed when porting applications to the zSeries by delivering a single driver on all platforms (including DB2 for z/OS V7 and V8, while also providing useful performance and functionality enhancements.	26

Web Enablement

Feature	Business Benefit	Page
Java Enhancements	The new Java function within the DB2 Universal Driver provides some valuable productivity and performance enhancements, consolidating Java's position as a primary zSeries programming language.	27
XML Enhancements	Although the XML Extender will still be required to decompose and store XML data in DB2, Version 8's built-in XML publishing functions offer some welcome performance improvements and point the way for more extensive native XML support in the future.	28

Performance & Operating Costs

Feature	Business Benefit	Page
Multi-Row Fetch & Insert	Multi-row insert and fetch is a significant performance enhancement that can potentially benefit many types of application, with CPU savings of up to 50% for some operations. Distributed applications will see even bigger improvements due to the ability to replace multiple (and time consuming) trips across the network with a single call.	29
Optimisation Enhancements	DB2's industry-leading SQL optimiser has been further enhanced in Version 8, offering more efficient query execution and lower CPU costs for many common queries.	30
Materialised Query Tables (MQTs)	MQTs have already proven their worth in DB2 for Linux, Unix & Windows, and are a powerful means of dramatically improving query response times and reducing resource consumption without any effort on behalf of the query user. A very valuable feature for many decision support / data warehousing environments.	31
Other Performance Enhancements	Together, these performance enhancements provide considerable scope for reducing the CPU requirements and increasing the throughput of customer workloads.	31

Developer / DBA Productivity

Feature	Business Benefit	Page
Productivity Enhancements	Version 8 introduces a number of very significant new features that will allow both DB2 administrators and application developers to become more productive in their day-to-day roles. With staff costs consuming an increasingly significant proportion of today's IT budgets, these enhancements could be an important factor in any cost justification for the move to Version 8.	32

Global Integration & Deployment

Feature	Business Benefit	Page
Unicode Support	Version 8 builds upon the initial Unicode support implemented within the previous release, removing many of the restrictions that were preventing customers from fully exploiting Unicode's considerable benefits in a multi-national application environment.	33

Data Security

Feature	Business Benefit	Page
Row Level Security	Version 8's multi-level security features fundamentally change the way in which DB2 access privileges are managed. Implementing them will require careful planning and analysis, but the vastly increased flexibility will make the effort worthwhile for those customers who require an even greater level of control and capability in security management and access to data.	34

Return On Investment

The following graphs summarise the overall return on investment for three typical Version 8 customers: a traditional OLTP environment, an SAP environment and a Java environment. Please refer to “Justifying The Upgrade” on page 43 for more details.

It should be noted that while these scenarios use real hardware and software costs, they are by necessity extreme generalisations and are based on a number of factors and assumptions that will vary significantly from customer to customer. However, the models presented should be of use to many customers in providing an outline of the potential financial position.

OLTP Customer

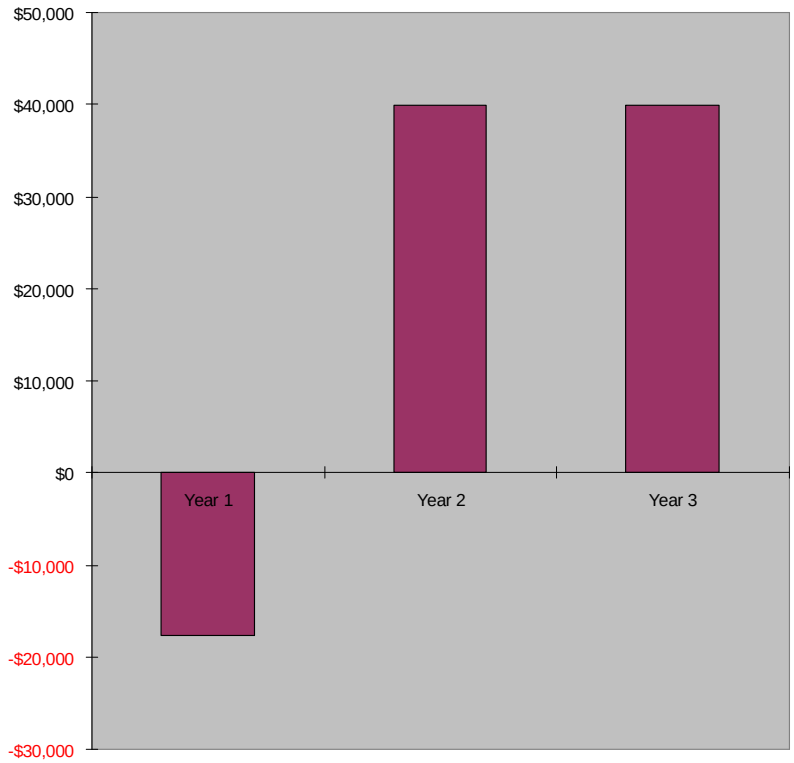

Figure 1 – OLTP Customer Overall Savings

SAP Customer

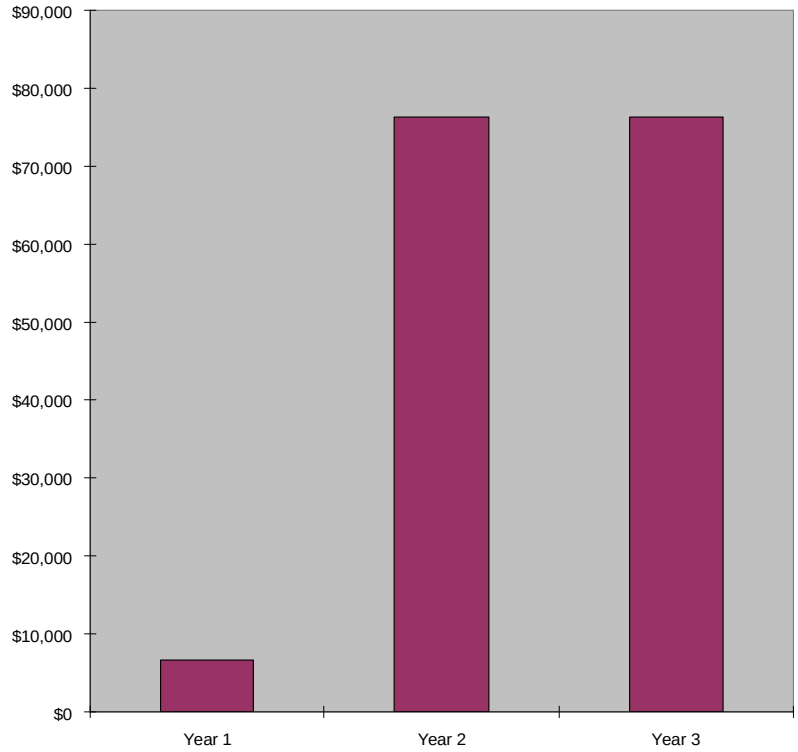

Figure 2 – SAP Customer Overall Savings

Java Customer

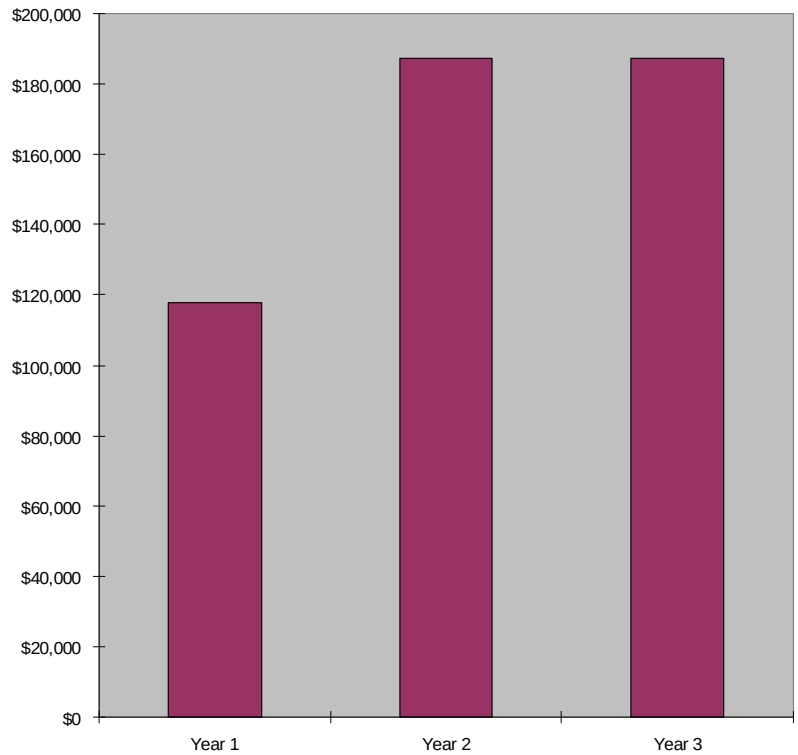

Figure 3 – Java Customer Overall Savings

Conclusion

DB2 for z/OS Version 8.1 promises to be one of the most significant releases ever for IBM's flagship database management product. It is the first middleware product to break the bonds of 31-bit computing and begin to exploit the enormous scalability within IBM's 64-bit zSeries machines, allowing customers to drive greater workloads through their systems and "do more with less".

Major advances have also been made in improving data availability, further reducing planned downtime and allowing today's demanding 24/7 applications to stay active for longer. Many customers will be able to use these features to implement changes to their databases more frequently, allowing them to react more rapidly to changing market conditions or unexpected performance issues.

Although the move to a 64-bit environment introduces its own CPU demands, these are likely to be more than offset by the significant performance enhancements in Version 8 – many of which can be realised with little or no application changes being required. The performance enhancements drive down the total ownership cost for DB2 applications, making each transaction cheaper to execute and allowing greater workloads to be handled by a given server.

The new release also removes many of the barriers preventing non-DB2 applications from being ported to the zSeries platform. As shown in the case study on page 40, this can open the way for some compelling cost cases to be built around server consolidation projects.

Finally, significant productivity savings are possible due to the increased automation and intelligence built into the product. This allows skilled DB2 administrators and developers to spend less time performing lower-level routine tasks and more time adding value to the organisation by producing better written, more efficient applications.

In combination, these new features represent a significant advancement in DB2's capabilities, and most customers are expected to see significant financial and technical benefits as a result of migrating to the new release.

Introduction

Data has always been the lifeblood of any organisation, but as today's e-business applications demand ever-greater availability and performance the choice of enterprise database is becoming more and more critical.

DB2 for z/OS Version 8 is one of the most important releases in the lifetime of IBM's flagship relational database management system. Building upon the solid foundation of Version 7, the new product has been extensively enhanced and re-architected to directly address many of the critical challenges facing IBM's most demanding enterprise customers.

A wealth of material exists on the technical changes within Version 8, but finding descriptions of how those new features will improve your business results can be a challenge. This paper will provide a high-level overview of the major new features from an IT Executive's perspective, with the emphasis on the underlying business value that DB2 V8 can deliver. It will also examine the business case for the upgrade from Version 7, including a generic Return On Investment (ROI) model that can be easily adapted to most customer's specific circumstances.

DB2 UDB for OS/390 Version 7 – The Foundation for Today's Enterprise

Today, DB2 is universally accepted as the premier database system for IBM's zSeries mainframe architecture. Although other products do exist for this platform[1], DB2 sits at the heart of most of the business-critical mainframe IT applications that have been written during the last 20 years.

DB2 Version 7 has been generally available since early 2001, and more than two-thirds of IBM's DB2 user base are at that release. The scalability and reliability of IBM's zSeries platform makes it a very attractive choice for customer's high-volume, mission-critical applications.

It should also be noted that Version 7 is a vital stepping-stone for those customers on previous releases of DB2 that wish to take advantage of the new features in Version 8 – migration from releases prior to Version 7 is not supported by IBM.

In this section, we'll briefly review DB2 UDB for OS/390 Version 7 and look at some of the ways in which it helps to deliver competitive edge.

Performance

Improving performance and therefore reducing the resources required to support a given workload is a key objective in any new release. Version 7 delivered a number of important enhancements in this area, including:

- **SQL Enhancements**. A number of performance-related SQL enhancements help to reduce resource consumption and reduce workload cost. These include improvements to DB2's ability to handle very high volumes of insert processing, limiting result sets to a set number of rows and avoiding unnecessary sort operations.

- **Scrollable cursors**. Many applications allow the user to page backwards and forwards though a set of displayed data. This often requires separate SQL statements to be executed each time the user moves forwards or backwards. Version 7 introduced a new scrollable cursor feature, which allowed the application developer to navigate backwards and forwards through a result set while retrieving the data just once from the table.

Availability

Reducing planned down-time and increasing availability helps DB2 to better support demanding e-business applications. Version 7 delivered some important new items in this area.

- **Online subsystem parameters**. Version 7 saw the initial implementation of this important facility, which allows system administrators to change many of the vital system configuration parameters without having to stop and restart the DB2 system. This facility has been further enhanced within Version 8 (please see Dynamic System Parameters on page 23).

[1] Including IBM's own DB2 for Linux, Unix and Windows, running in a Linux for zSeries partition.

- **Restart Light**. This enhancement can reduce the amount of time that DB2 data is unavailable in the rare event of a DB2 member in a data sharing group failing.

Web Enablement

Version 7 marked another important step forward in DB2's ability to support demanding e-business applications. Web-enablement features delivered in this release included the following items:

- **Unicode support.** Version 7 introduced initial support for Unicode[2] storage, by allowing user data in DB2 tables to be stored in Unicode format. This enhancement makes it considerably easier for multi-national applications to correctly store and display data regardless of the user's local language. Version 8 further expands DB2's Unicode capabilities considerably, as described in the section on Global Integration and Deployment on page 33.
- **Network monitoring.** One of the key challenges introduced by distributed applications is how to break down the overall response time if performance needs to be improved. Version 7 introduced a new feature that allows DB2 to pass useful performance information back to the calling client. This makes it quicker and easier for database administrators to diagnose and resolve distributed application performance issues.
- **Support for JDBC 2.0.** JDBC is the industry standard for connectivity between Java applications and relational databases. DB2 Version 7 introduced support for the JDBC 2.0 standard. This incorporated a large number of functionality and performance improvements, including new data types, support for scrollable cursors, and connection pooling.

Application Development

Improving developer productivity has an immediate and obvious impact on the overall cost of developing and maintaining an application. Version 7 introduced several enhancements in this area, as described below:

- **Java Stored Procedures.** Stored procedures promote code re-use and offer considerable potential for improving the response time for many applications. Version 7 introduced support for high-performance interpreted Java stored procedures. Previous versions required the developer to perform a complex compilation process before the Java code could execute, but this is no longer necessary in Version 7.
- **SQL Stored Procedures.** Version 7 significantly enhanced support3 for a new type of stored procedure, written entirely in ISO-standard SQL rather than in a traditional programming language. These stored procedures are more easily portable between platforms, and very easy to write and implement compared to their more traditional counterparts.
- **New extenders.** Version 7 shipped a XML and extender, built upon the object relational features introduced previously in Version 6.. The extender allows the user to both generate XML documents from DB2 data and to decompose XML documents and store the constituent items in DB2. These facilities have been extended with the new native XML function support in Version 8, as described in XML Enhancements on page 28.

[2] Unicode – the global character encoding standard designed to allow applications to correctly store and display characters regardless of the platform, language or program.

[3] The SQL stored procedures functionality is also available for DB2 Version 5 and Version 6 as a separately installable feature.

DB2 for z/OS V8 – Driving Business Value

In this section, we'll take a more detailed look at the major features of Version 8, and how many of IBM's largest enterprise customers are intending to use them to deliver an enhanced IT service to the business.

Scalability and zSeries Synergy

Scalability – the ability to easily increase the capacity of a system to cope with increased demand – has never been more important. Today's IT systems are typically highly integrated with an organisation's partners, suppliers and customers, and must be able to quickly respond to customer demands, new market opportunities or competitive threats. IBM has termed such applications "On Demand", and they routinely have to deal with peaks in transaction throughput that were unheard of even a few years ago.

In this sort of environment, the ability to be able to expand the processing capabilities of core infrastructure components such as the database system in a rapid and cost-effective manner is critical.

64-Bit Support

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

Today, storage constraints are probably the single biggest factor in limiting the scalability of a single DB2 Version 7 system. Each process that runs concurrently within that system requires some storage, so the more workload a given system is asked to handle, the higher the storage requirements. Historically, the storage requirements of a single DB2 system have been growing at 20-30% per year to support the increased workload demands being placed upon it.

Since Data Sharing became available in DB2 Version 4 back in the mid-nineties, it has been possible to scale horizontally by adding more DB2 systems into a single "data sharing group", with all members of the group sharing access to the same common set of data and coordinated via a Coupling Facility (see Figure 4 - DB2 Data Sharing Architecture).

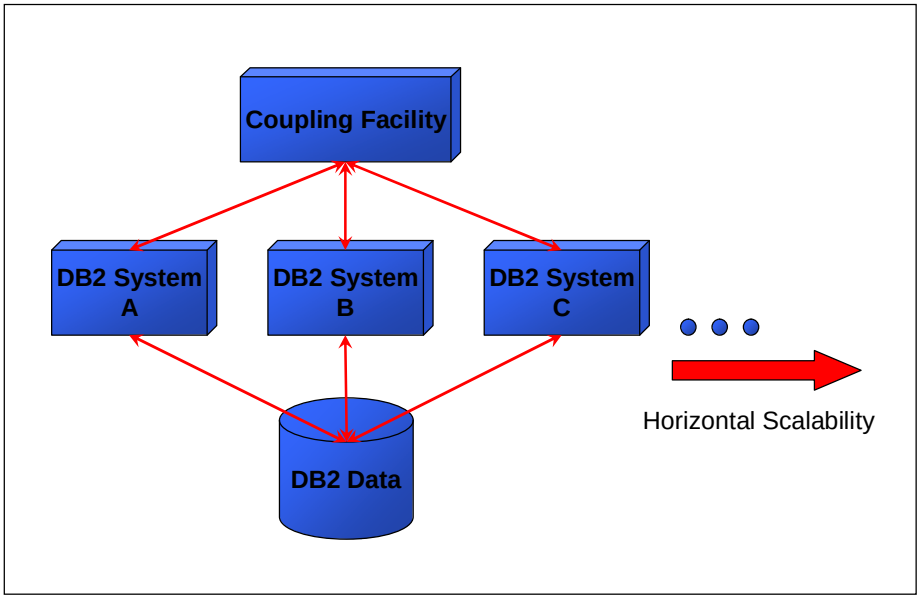

Figure 4 - DB2 Data Sharing Architecture

This data sharing approach provides excellent scalability and resilience. However, the decision to move to a data sharing environment should not be made lightly as there can be implications to performance and the complexity of the operating environment. Financial considerations, such as purchase of the coupling facility and software licensing costs, must also be evaluated.

For these reasons, most DB2 customers try to minimise the number of subsystems within a data sharing group. Some customers with less demanding resilience requirements may even avoid the use of data sharing altogether, preferring to vertically scale up individual DB2 systems.

So, what are the storage constraints preventing this vertical scalability?

Well, its certainly not down to the price of the storage itself. zSeries incremental storage costs have been falling steadily, and at around $10,000 per GB they are approximately a quarter of their value at the time the z990 was announced in mid 2003.

So, the cost of storage itself is no longer the issue, its more a matter of DB2's ability to fully use it. Those customers who are not using the latest zSeries hardware[4] in 64-bit mode are limited to 31-bit real storage addressability (see Figure 5 – z/OS 64-Bit Memory Map on page 17). Practically speaking, this means that any application (including DB2) can only directly access up to 2GB of "central" storage. Any additional storage (and with those rapidly falling storage prices it is common for 8GB or more to be installed) has to be treated as separate "expanded" storage, and although DB2 can use this to store read-only data, it is less efficient (and therefore more expensive) to access than "central" storage. Although it is still usually more efficient for DB2 to fetch data from expanded storage than it is to get it from disk, other limitations in the use of expanded storage make this a less than ideal solution.

If you are fortunate enough to be running a recent zSeries machine in 64-bit mode and the correct version of OS/390 or z/OS[5], these limitations are removed and up to 16 Exabytes (an incredible 2^{64} bytes) can theoretically be addressed[6]. This completely

[4] Limited to IBM eServer models z800, z890, z900 and z990 at the time of writing.
[5] OS/390 V2R10 or above.
[6] However, at the time of writing the maximum storage available on any hardware configuration is 256 GB (on an IBM eServer Z990).

removes the concept of “expanded” storage, and allows much more of the memory[7] to be directly used by DB2 for caching data, which in turn allows data to be accessed in a more efficient manner.

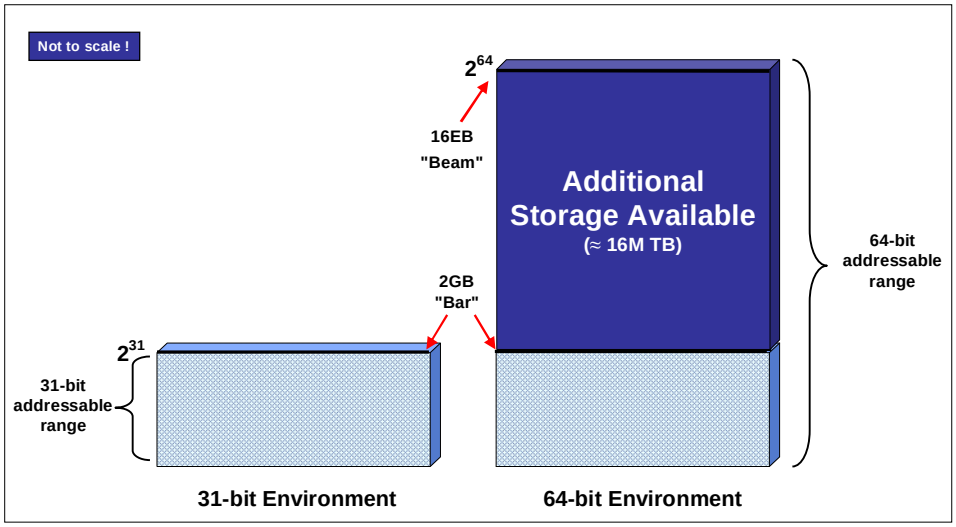

Figure 5 – z/OS 64-Bit Memory Map

However, despite the ability to cache more data, internally DB2 Version 7 is not able to fully exploit the new 64-bit memory model. Critical internal storage areas are still resident below the 2 GB memory “bar”, limiting the number of concurrent transactions that a single DB2 system is able to support. Therefore, even on systems with very large amounts of storage available, vertical scalability is limited and customers may have to continue to scale horizontally. The 32 GB maximum limit on data caching is also an issue for the largest eServer models, which can be configured with up to 256 GB of storage.

Version 8 has been comprehensively reworked to address this issue, and fully exploits the latest 64-bit hardware and operating system advances. One of the overall themes for Version 8 is “breaking the limits” and nowhere is this more true than with its storage exploitation - DB2 is now able to directly address all of the available storage within a given zSeries machine.

As processor speeds continue to increase at a faster rate than disk subsystems, the relative cost of performing I/O is increasing, and minimising I/O has become a major objective in improving performance. Version 8 allows the amount of data DB2 is able to cache in storage to be increased (storage permitting) from the previous limit of 32 GB to a massive 1 Tb[8]. This could dramatically decrease the amount of I/O required to support a given workload, allowing greater throughput with no CPU upgrade being required. In some cases, this has been shown to reduce the elapsed time of I/O intensive processes such as batch programs by an order of magnitude. This change also simplifies DB2’s use of storage, consolidating all of the data cache areas into a single z/OS address space and thereby simplifying DB2 systems management and operations.

In addition to this increased data caching capability, DB2’s internal code and storage usage has been enhanced to allow it to fully exploit the increased headroom available in a 64-bit computing environment. As a result, many of the limits imposed on the

[7] Up to 32 GB

[8] Note that this is the Version 8 architectural limit. In practice the data cache should be fully supported by storage in the zSeries server, so the current limit on a fully configured IBM eServer z990 is 256GB, or 0.25 TB

amount of work that can be handled by an individual subsystem have been significantly increased (for example, the maximum number of datasets that can be open at any one time has been increased from 32,000 to 100,000).

In combination, these 64-bit enhancements will allow many customers to reduce the overall number of DB2 systems required to support a given workload, with a corresponding decrease in system overheads. No application changes are required in order to realise these benefits – they are available immediately following the upgrade to Version 8.

However, it is important to note that the move to a 64-bit environment is not completely without costs. DB2 is required to perform additional work to operate in this environment, and some customers may see CPU increase by around 5-10% as a result. The good news is that this increase should be more than offset by the significant performance and productivity enhancements described elsewhere in this document. It is important that all of these costs and benefits are considered as part of the migration justification (see Justifying The Upgrade on page 43).

***The bottom line**: Version 8 allows customers who have invested in IBM's latest generation of hardware to fully exploit the immense storage capabilities of the 64-bit memory model, with no changes being required to their applications. This in turn allows DB2 systems to cache significantly more data and to vertically scale much higher than was previously possible, resulting in a reduction in system overheads and more efficient use of limited machine resources. In some cases, these enhancements may be sufficient to allow customers to defer the implementation of data sharing, if this was being considered purely for scalability reasons. Although customers can expect some increase in CPU requirements following the move to a 64-bit environment, these increases are expected to be offset by the significant performance and productivity enhancements within the new release.*

Partitioning Enhancements

For many years, DB2 has allowed designers to split large tables into a number of smaller pieces (called partitions) to allow them to be more easily managed. Throughout that time, the maximum number of partitions has been limited to 254.

This limit has hampered database designers, who often had requirements for many more partitions to be supported (for example, a common requirement for time-series data is to be able to store 1 day's worth of data in each partition, and keep 7 year's worth of data online – a total of 2,548 partitions). In order to work around this limitation, the designer often had to compromise the database design, which usually resulted in sub-optimal query performance. In some cases, this may also have led to more intrusive housekeeping activities, affecting data availability

DB2 Version 8 increases the 254 partition limit, allowing a maximum of 4,096 partitions for a single table. This gives database designers much more flexibility in choosing the most efficient partitioning scheme for a given table.

Another major Version 8 partitioning enhancement relates to the way in which data is physically ordered in tables. In previous releases, it was necessary to physically order the data using exactly the same attributes as those used to partition the data. In the example shown in Figure 6 below, the table holding customer data is partitioned according to customer number, so the data within each partition also has to be physically ordered by customer number. Again, this situation could force database designers to make compromises in the design, as the optimal partitioning attribute and the optimal physical ordering attribute are not always one and the same.

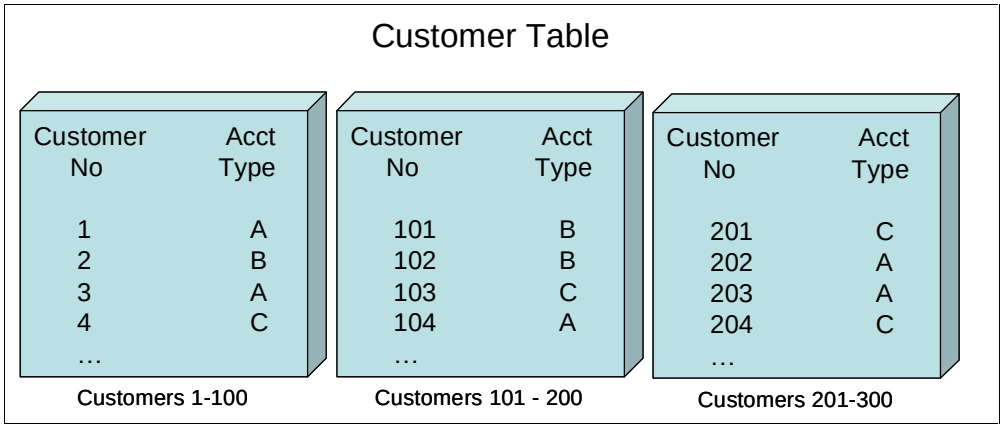

Figure 6 – Version 7 Partitioning

For the first time, Version 8 allows these two decisions to be divorced, so that the database designer is able to pick the most efficient attributes for each based on their own merits. In the example shown in Figure 7 below, the table is partitioned by customer number but the data in each partition is physically ordered by account type. This enhancement will allow many queries to operate more efficiently, thereby reducing the machine resources required for a given workload.

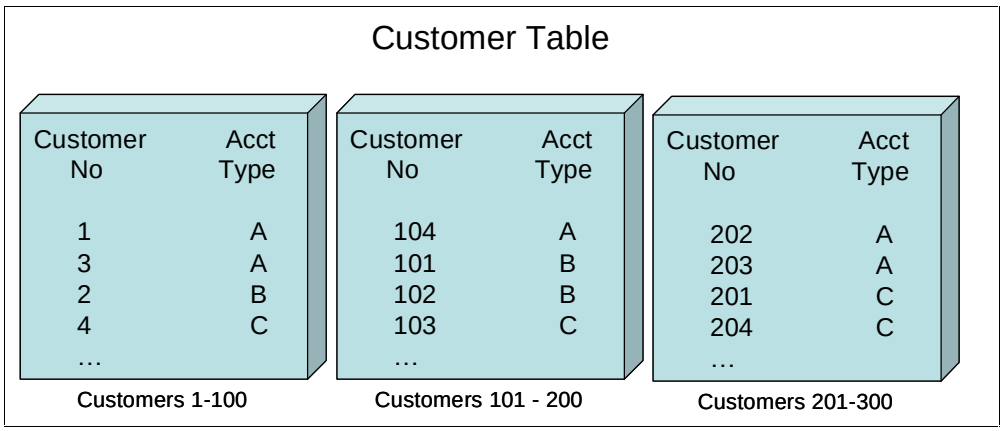

Figure 7 – Version 8 Partitioning

Both partitioning enhancements can be implemented on existing tables using Version 8's new dynamic schema change capabilities, significantly reducing the risk, effort and data availability impact (see page 21 for more details).

***The bottom line**: Version 8 provides some major enhancements to the partitioning options available to DB2 database designers, allowing them to create more efficient designs that are able to support higher transaction loads for a given amount of machine resource. These benefits can be implemented on existing database designs with minimal loss of availability, due to the new dynamic schema change capabilities.*

Recovery Log Enhancements

As the throughput of a single DB2 subsystem increases, the amount of recovery log data produced by that system will also tend to increase. Keeping this recovery log data available online is important, as it dramatically speeds up recovery operations, maximising data availability,

DB2 Version 7 allows a maximum number of 31 online recovery log datasets[9]. Version 8 triples this limit to 93 log datasets[10], allowing DB2 to keep recovery log data online for longer periods despite the increased throughput that the scalability enhancements described above are likely to generate.

A related enhancement increases the number of offline recovery logs that DB2 Version 8 is able to use from 1,000 to 10,000. This increases the scope of log data that DB2 is able to access[11], making it possible to take backups of some data less often and therefore saving CPU costs and reducing routine housekeeping effort.

***The bottom line**: Version 8 allows significantly more recovery log data to be held both online and offline, allowing a DB2 system to handle larger workloads while maintaining the recoverability of the data.*

Other zSeries & z/OS Synergy Items

Other examples of the synergy between zSeries hardware, the z/OS operating system and DB2 Version 8 include:

- **Unicode conversion services**. Certain zSeries machines[12] have specific hardware instructions to assist in the conversion to and from Unicode, and these are used by associated enhancements within z/OS V1.4 to dramatically improve the performance of character conversions. Version 8 is also able to perform some conversion internally, streamlining the conversion process still further. (The advantages of the move to Unicode are covered in more detail in Global Integration and Deployment on page 33).

- **Cryptographic support**. Version 8 is able to make use of the z/OS Integrated Cryptographic Service Facility to increase security for sensitive information transmitted to distributed clients.

***The bottom line**: Version 8 exploits specific features in z/OS and the zSeries hardware to deliver highly efficient character conversion and encryption services.*

Data Availability

The resilience and reliability offered by the zSeries platform is a major reason for its use in so many mission-critical applications today. Unfortunately, although this allows *unplanned* outages to be minimised, there are still situations where a *planned* outage is necessary to perform essential system maintenance. These planned outages are becoming increasingly painful – batch windows are shrinking or disappearing altogether, and maintenance windows are a precious commodity that many customers simply cannot afford. Many global On Demand[13] applications have availability requirements that amount to the data having to be online 24 hours a day, 7 days a week, 365 days a year. In this environment, any form of planned outage becomes a major business issue.

In database terms, three of the most common reasons for data being made unavailable during a planned outage are:

[9] Equivalent to approximately 124GB of recovery log information, if the logs are defined at their maximum size.
[10] Equivalent to approximately 372GB of recovery log information, if the logs are defined at their maximum size.
[11] Increased from 4Tb to 40Tb.
[12] These instructions were initially introduced on the IBM eServer z900, and further enhanced on the IBM eServer Z990.
[13] On Demand is IBM's term for IT systems that are highly integrated with an organisation's partners, suppliers and customers, and must be able to quickly respond to customer demands, new market opportunities or competitive threats.

- **Data housekeeping**. This involves running special utility programs against the database in order to take backups, collect database statistics, reorganise the data for efficient access, etc.

- **System configuration change**. DB2 is a highly configurable database management system, with a large number of system parameters (known as DSNZPARMS) that govern the way in which it operates. Changes to many of these parameters require the system to be stopped and restarted to take effect, potentially resulting in a system-wide outage that could affect a number of different applications.

- **Schema change**. As new versions of an application are put into the production environment, an associated change often has to be made to the structure of the database. Resolving some performance issues may also require database structure changes. Depending upon the nature of the change, this can result in significant periods of unavailability.

IBM has made great progress in addressing the first issue, by consistently enhancing the DB2 utilities in the last few releases to allow more and more processing to take place while the data remains fully available for applications to read and update. Although this problem has not yet completely disappeared, its impact can often be reduced to manageable proportions in a properly designed application. A notable exception to this is the situation surrounding partitioned tables[14]. One of the major reasons for partitioning a table is to reduce the impact of routine data housekeeping operations, as it is possible to conduct these operations on a single partition at a time, thereby allowing application access to the remaining partitions and improving data availability. However, this "partition independence" does not extend to most of the indexes defined on the table, so during data maintenance activities access to the entire table can be restricted even if the maintenance is being performed on only a single partition.

The problems associated with system configuration changes have largely been addressed though a combination of the Version 7 enhancements allowing many of them to be dynamically altered, and data sharing (where individual members of a data sharing group can be reconfigured while applications running in the remaining members continue to access the date).

Unfortunately, the outages caused by the need to make schema changes continue to be a major source of disruption and data unavailability for many customers.

Dynamic schema change

One of the defining characteristics of many of today's mission-critical On Demand applications is a short application development cycle. Businesses need to react rapidly to changing marketing conditions and competitor initiatives, and that often means amending an application and implementing a new version in the production environment. These changes can often require changes to the structure of the database to be made at the same time.

Even if the application itself is relatively stable, database administrators often have to make changes to the database definition for operational reasons. Common examples include changes made to improve the performance of a critical business process, or amending the distribution of data within a partitioned tablespace to accommodate unexpected growth patterns.

[14] Tables (usually large) which have been split into multiple parts for performance, availability or manageability reasons.

Prior to Version 8, DB2 allowed very few changes to be made to existing structures without making the data unavailable. New tables and indexes could be added to an existing database, and new columns could be added to the end an existing table. Some other storage-related parameters could also be changed, but anything more fundamental (such as removing a column, or converting a column from one data type to another) required the entire table to be dropped and re-created. Of course, the data within the table would have to be unloaded prior to this, and reloaded again afterwards. This process (depicted in Figure 8 below) could take several hours or even days for large tables, during which time the data would be completely unavailable to any application that used it.

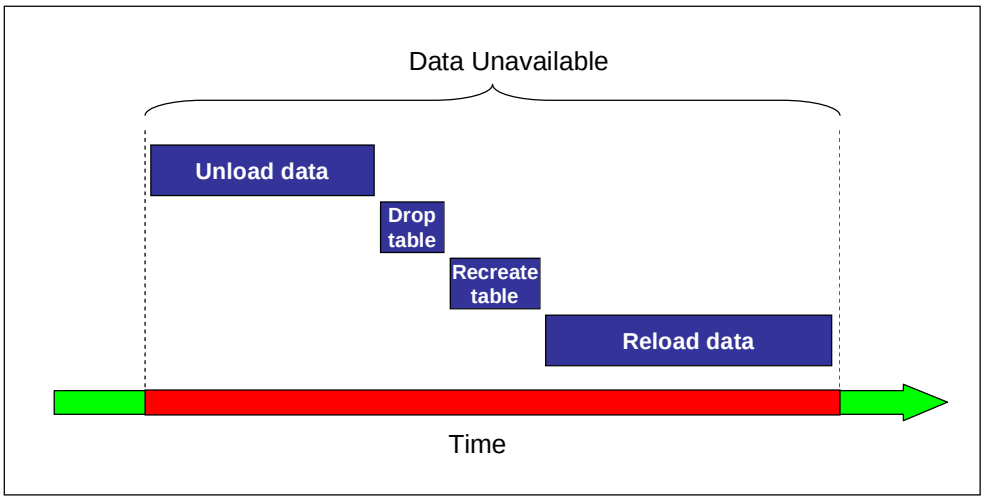

Figure 8 – Data Unavailability During Schema Change

Version 8 introduces the first implementation of IBM's solution to this problem, which they have dubbed "dynamic schema change". The concept is simple: allow changes to be made to the database structure dynamically, while applications retain full access to the data.

IBM has identified the most common changes that currently require data to be made unavailable, and targeted these within the Version 8 enhancements. These new capabilities include:

- Changing column data types and lengths. This is a very common requirement (for example a numeric column representing a customer number may originally have been defined as having 8 digits, but due to unexpectedly high volumes of business it must be increased to 10 digits.

- Adding columns to indexes. This is a common requirement for performance tuning purposes, and prevents the administrator from having to drop and recreate the index.

- Changing the physical order in which the table data is stored - another change that is frequently implemented for performance reasons.

- Adding a new partition to an existing table, and rebalancing the data within an existing partitioned table. This is often required for operational reasons, as the data grows in ways that were not originally envisaged when the database was designed.

These new features will go a long way to reducing the amount of time data has to be unavailable due to dynamic schema changes, allowing data to remain accessible for longer. Some challenges remain (there are other schema changes not supported in

the initial set of capabilities, and a short term performance penalty to be paid following an online change), but this should be viewed as a significant initial step in the right direction by IBM, and these features will certainly be augmented and expanded in future releases.

"Availability enhancements in every new release of DB2 for z/OS are always important to GAD and the latest introduction, Online Schema Evolution has met with praise from the DBA group who await further enhancements in future releases of the product."

Claus Weidenfeller, Lead DBA and DB2 Early Program Project Manager, GAD

"We have tested partition rotation, partition addition and rebalancing and these would increase our availability and reduce our risk due to change."

Version 8 Beta Customer

The bottom line*: Version 8 provides a good initial set of capabilities for changing database structures while retaining application access to the data. Many customers will be able to use these features to implement changes more frequently, allowing them to react more rapidly to changing market conditions or unexpected performance issues. Database administrator productivity will also be improved (and the risks associated with the change reduced), due to the significantly simpler change process.*

Partitioning Enhancements

As already discussed in the section on "Scalability" on page 15, Version 8 introduces some significant partitioning enhancements. As well as improving overall scalability, these enhancements also offer some very useful availability advantages.

When using the new partitioning approach introduced in Version 8, each partition can be made truly independent of the others. This allows data maintenance to be conducted on a given partition while allowing full application access to the remaining table partitions.

In addition, the ability to create many more partitions in Version 8 allows each partition to be smaller, thereby limiting the impact of any maintenance activities even further.

The bottom line*: The new partitioning model introduced in Version 8 allows true partition independence for the first time. This reduces the availability impact of table reorganisations, allowing the administrator to keep the data better organised and therefore potentially improving the performance of the application.*

Dynamic System Parameters

IBM has followed through on the work begun in Version 7 in allowing many of DB2's system configuration parameters (DSNZPARMS) to be changed while DB2 is active. Version 7 allows approximately 60 parameters to be changed in this way, and Version 8 adds more than 40 to that total (consisting of all of the new parameters introduced in Version 8, together with more of the older ones).

The bottom line*: The additional dynamic system parameters in Version 8 decrease the likelihood of the entire DB2 system having to be made unavailable while configuration parameters are changed, which improves overall availability and allows system administrators to be more responsive to changing application demands.*

System-Wide Backup & Recovery

Ensuring that DB2 data is properly backed up so that it can be recovered in the event of data loss or corruption is a major objective for many database administrators. Depending on the technique used to backup the data, it can also be responsible for a significant reduction in data availability.

Version 8 introduces the capability to backup an entire DB2 system (including all of the application data) with a single utility. Another allows the entire system to be restored to a user-defined point in time. This feature uses new facilities within z/OS to invoke ESS Flashcopy, which is able to very rapidly take volume-level dumps of all of the data in the DB2 system. Although it is rare to have to restore an entire DB2 system, this facility will be of great benefit in DB2 systems hosting CRM / ERP vendor applications (which typically reside in a subsystem of their own).

***The bottom line**: Provided you have the right DASD subsystem, the new system-wide backup and recovery capabilities can reduce the time that DB2 data has to be unavailable due to system backups. Perhaps more importantly, the ability to rapidly restore a system to a given point in time means that applications can be made available again in the minimum possible time in the event of a major problem.*

Application Porting

As the hardware, software and environmental costs associated with the mainframe continue to decrease, the zSeries is becoming an increasingly attractive place to host applications originally designed and developed for other platforms. For many customers, there is a compelling case to be made for porting existing applications from other platforms onto zSeries, so that they can exploit its industry-leading scalability and resilience. Other customers enjoy significant developer productivity gains from developing new applications in a distributed environment such as Windows, and porting to the mainframe for the final stages of testing and implementation. ERP and CRM vendors are also interested in the potential for porting their existing application code to run against DB2 on the mainframe platform.

Regardless of the reasons, a number of issues exist that hinder the application porting process and make it more complex, risky and expensive. These include:

- **DB2 for LUW database incompatibilities**. Differences exist between the implementations of DB2 on zSeries and the distributed platforms, and these differences can require changes to be made to application code and/or the database structure definitions as part of the porting exercise.

- **ISV database incompatibilities.** Differences also exist between DB2 and other databases such as Oracle and SQL Server, but they tend to be more major and therefore require a correspondingly higher degree of effort to overcome.

IBM has steadily reduced the number of internal incompatibilities that are relevant from an application perspective during the last few development cycles of the mainframe and distributed DB2 products. However, some significant differences remain and, depending upon the size and complexity of the application, those differences may require a large amount of effort to address. A typical example is the maximum length of a table name. DB2 on the distributed platforms allows table names of up to 128 characters in length, whereas DB2 for OS/390 & z/OS Version 7 restricts tables names to 18 characters. This restriction can require significant changes to both the database definition and the code when porting an application to the mainframe.

IBM has also been consistently adding function to DB2 that (directly or indirectly) provides similar functionality to that present in competitive ISV database products. Each time one of these items appears, the porting process becomes a little easier but with Version 7 there are still significant challenges to be overcome when considering a database migration.

Version 8 removes most of these portability barriers, opening up many possibilities such as large-scale server consolidation exercises that would previously have been less economically compelling. For a real-world demonstration of the financial benefits that such an approach can deliver, please refer to “A DB2 for z/OS Version 8 Customer Case Study: Univar” on page 40.

SQL enhancements

Many other database systems (including the current version DB2 for Linux, Unix and Windows) support much longer table and column names than DB2 Version 7 does. This is one of the major barriers to application portability, as these longer names have to be abbreviated in both the database definition and in each SQL statement within the application code – a time-consuming process. Version 8 increases table names from 18 to 128 characters, and column names from 18 to 30 characters (the same limits as those within Version 8 of DB2 for Linux, Unix and Windows), removing the need to amend table and column names when porting from most other database systems.

The maximum size of a single SQL statement has also been increased, from 32K in Version 7 to 2MB in Version 8. As well as accommodating the inevitable increase in SQL statement length caused by the longer names outlined above, this change makes it possible to port long SQL statements written for other platforms without having to suffer the cost and delay needed to manually re-code them.

Version 8 allows up to 225 tables to be referenced in a join operation (up from just 15 in Version 7). Again, this allows many SQL statements that would previously have required extensive re-coding to execute with little or no change on DB2 for z/OS. This enhancement is of particular value to ERP/CRM vendor applications, which make extensive use of large table joins in their applications.

One of the more subtle issues that can cause problems when porting an application to the zSeries platform is the difference in sort order (also known as the “collating sequence”). Due to the different encoding schemes used for mainframe and non-mainframe characters, an SQL statement executed on DB2 for z/OS can return data in a different order to an identical statement executed on one of the distributed platforms. For example, a typical mainframe query will return lower-case characters first, followed by upper-case character and then numbers. On the distributed platforms, the same query would return numerals first, followed by upper-case characters and then lower-case characters. This can cause unexpected results in some applications, and require additional coding effort to resolve. As described in detail on page 33, Version 8 makes extensive use of Unicode, with all SQL statements being processed internally in Unicode format. As most distributed applications will either use Unicode natively or use ASCII (which shares the same sorting sequence as Unicode) the sort sequence should be the same on both platforms, making portability easier.

A large number of other SQL enhancements within Version 8 will assist with application portability, in addition to improving developer productivity for new applications. These include:

- A number of new SQL functions have been introduced that are direct equivalents to (and in many cases syntactically identical to) function offered in DB2 for LUW and other vendors' database products.
- The maximum length of an index key has been significantly increased[15]. As for the SQL name length increases mentioned above, this removes another category of changes previously required to the database design when porting to DB2 for z/OS.
- Many non-DB2 applications use objects known as "sequences" to automatically generate unique numbers. Previous versions of DB2 for z/OS supported a similar concept known as identity columns, but these were less flexible and required special handling. Version 8 introduces specific support for sequences, increasing compatibility with other vendor's databases.
- DB2 for Linux, Unix & Windows has supported recursive SQL (also known as "bill of materials" queries) for some time. These queries are now also supported in Version 8, making it unnecessary to undertake the complex task of recoding the queries when porting an application to zSeries.

"GAD invested time in successfully testing the DB2 family integration support in Version 8, by porting an existing UNIX DB2 environment to the DB2 for z/OS platform without any problems, other than the creation of some indexes which were not created automatically."

Claus Weidenfeller, Lead DBA and DB2 Early Program Project Manager , GAD:

The bottom line*: Together, these SQL enhancements reduce the effort, complexity, risk and cost of porting database definitions and application code to DB2 Version 8, making it an even more attractive platform for vendor application porting infrastructure consolidation projects and development on Windows, Unix or Linux.*

DB2 Universal Driver

Java client applications need the services of a "driver" to allow them to access database resources on the server. Prior to Version 8, three distinct families of DB2 clients provided these drivers (the DB2 run-time client, the DB2 application development client and DB2 Connect), with each being targeted at a specific platform and purpose. Internally, these drivers used different protocols and required different handling within the Java application, making it necessary to amend code when porting the Java client between platforms – a major additional effort.

Version 8 introduces a new Universal Driver that uses a single protocol[16] and code base regardless of the target platform (local or remote). As the application interface is now identical across all platforms, application portability for Java applications has been significantly enhanced. Performance is also improved when using this new driver (due to a smaller memory footprint) and some useful functional enhancements are also delivered (see section on "Web Enablement" below).

The bottom line*: The new Universal Driver removes yet another set of code changes needed when porting applications to the zSeries by delivering a single driver on all platforms (including DB2 for z/OS V7 and V8), while also providing useful performance and functionality enhancements.*

[15] From 255 bytes to 2,000 bytes
[16] DRDA, an Open Group standard

Web Enablement

Exploitation of Web Technologies

As customers embrace the many opportunities offered by e-business, more and more of their traditional mainframe production workload is being web-enabled. However, for all its benefits, exploitation of web technology brings its own challenges:

- **Use of Java**. Java is becoming more and more popular as a zSeries development language. IBM moved early to support Java development within DB2, and Version 7 provides a robust set of capabilities in this area (such as support for high-performance static SQL in Java programs via the SQLJ standard). However, despite these advances, there is still room for improvement in the Version 7 performance of both dynamic (JDBC) and static (SQLJ) SQL within Java applications.

- **Application Changes**. Regardless of the language used, web-enabling existing applications often means having to re-write or add additional code. As the costs of hardware and software decrease, developer productivity is a key factor in the overall cost of any application and the ability to rapidly implement this new code becomes increasingly important.

- **Use of XML**. In recent years, XML has rapidly established itself as the de-facto standard for data exchange on the Internet and corporate intranets, and a key technology underlying many business-to-business applications. A common requirement with such applications is to publish data held in a DB2 database in XML format. Although Version 7 makes this possible, the solutions available either require significant developer effort or involve performance compromises.

- **Security concerns**. Allowing customers and/or suppliers access to your operational data obviously requires robust, auditable access control mechanisms to be in place, with the flexibility to limit privileges to specific rows in a table. It is possible to accomplish this with Version 7 only through the use of views, which can be cumbersome to implement.

The scalability and resilience of the zSeries platform makes it an ideal choice for today's mission-critical On Demand applications. In Version 8, IBM has retained its focus on web-enablement, extending DB2's Java and XML capabilities and reinforcing its reputation as the leading engine for high-volume e-business.

Java Enhancements

The new DB2 Universal Driver previously mentioned delivers some useful functionality enhancements in addition to much improved portability. The driver is fully compliant with the JDBC 3.0 industry standard, and introduces a number of valuable new functions including:

- **Scrollable cursor support**. This allows developers to navigate backwards and forwards through the result set from a query without having to resort to program code, thereby improving developer productivity and application performance.
- **Batched updates**. This feature allows an application to group together multiple database update operations and submit them to DB2 in a single network exchange. For certain applications, this could yield significant performance benefits.

- **Enhanced SQL error messages**. Extensions to the standard error handling allow much more specific information to be passed back to the application in the even of an SQL error. This saves developers from having to write additional code to determine the error information, improving their productivity.
- **Java client information**. An extension to the Java API allows the client to pass user-defined information to the back-end DB2 server. This allows individual clients to be allocated specific execution priorities on the server, potentially improving workload management and allowing better throughput for business-critical transactions.
- **SAVEPOINT support**. Savepoints allow application-defined consistency points to be requested by the application, to allow data to be easily restored to an earlier point in time. This facility will save the developer from having to manually code the equivalent function, improving productivity.

***The bottom line**: The new Java function within the DB2 Universal Driver provides some valuable productivity and performance enhancements, consolidating Java's position as a primary zSeries programming language .*

XML Enhancements

XML is becoming increasingly important as the de facto standard for information exchange between internal and external IT systems. A common requirement for such systems is to be able to represent data held within DB2 databases in XML format (a process known as XML publishing).

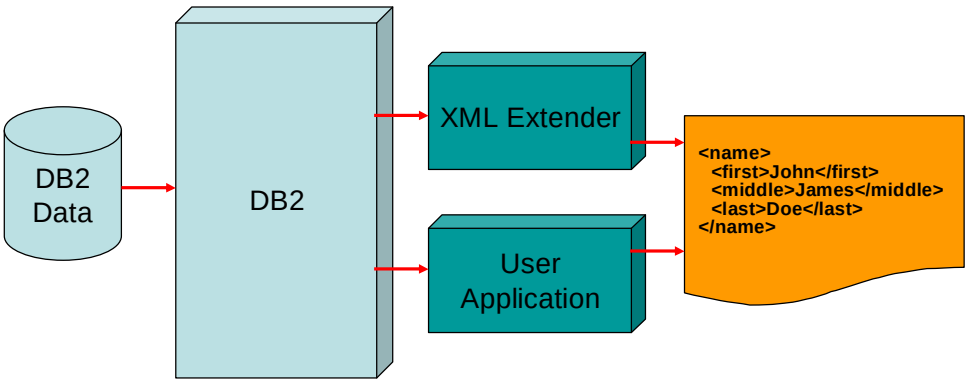

Figure 9 – Version 7 XML Publishing Architecture

Version 7 provided an XML Extender to accomplish this, but in common with other User Defined Functions, these run in an area that is isolated from the main DB2 code and are therefore less efficient than internal DB2 functions (see Figure 9 above).

Version 8 implements a full set of XML publishing functions that are implemented as internal SQL functions, as depicted in Figure 10 below. This enhancement allows a standard SQL statement to return a native XLM data stream with significantly improved performance.

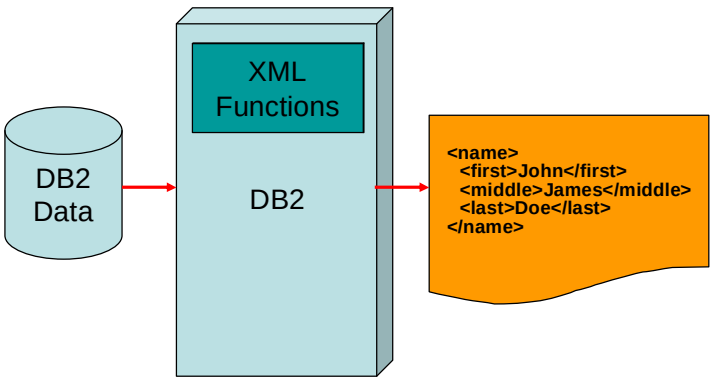

Figure 10 – Version 8 XML Publishing Architecture

***The bottom line**: Although the XML Extender will still be required to decompose and store XML data in DB2, Version 8's built-in XML publishing functions offer some welcome performance improvements and point the way for more extensive native XML support in the future.*

Enhancing Performance & Reducing Operating Costs

The need to constantly improve performance is a recurring theme with most relational database customers. Whereas in the past this may have been driven largely by the desire to meet demanding service level objectives set by the business, today the desire to "do more with less" and sustain larger workloads with a given hardware investment (and therefore reduce the total operating costs) is a major factor.

Every release of DB2 delivers significant performance enhancements designed to drive down the amount of resource required for a given DB2 transaction, and Version 8 is no different. In the sections that follow, we'll take a look at the major performance-related features delivered within the new release. These features take on a special significance in Version 8, where they are expected to compensate for the additional CPU overheads associated with the move to a 64-bit environment (see 64-Bit Support on page 15 for more details).

Multi-Row Insert & Fetch

The insert and fetch operations are two of the most common activities performed against a DB2 table. Typically, they will be executed many times in a program loop, as an application adds or retrieves multiple rows from the database, one row at a time.

For the first time, Version 8 introduces the capability of fetching or inserting multiple rows in a single SQL operation. In the case of an insert, the developer simply places all of the rows to be inserted together in program storage, and passes them to DB2 as a group. Similarly, multi-row fetching simply involves specifying an area of program storage and the number of rows to be retrieved, and DB2 will place the rows into storage in a single operation.

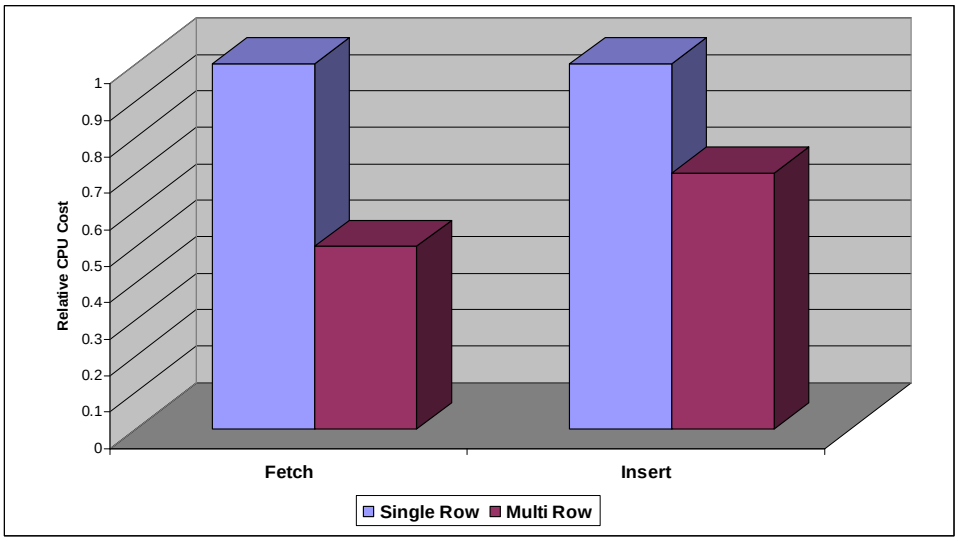

Figure 11 - Multi-Row Insert / Fetch - Potential CPU Savings

As shown in Figure 11 above, initial IBM tests have indicated CPU reductions of up to 50% for fetch and 30% for insert when using this feature within non-distributed applications. Early Version 8 customers have confirmed these savings, with 30-40% savings being common. Eliminating expensive network delays makes the savings even more impressive in a distributed environment, where elapsed time can be decreased by up to 8 times, with a 4 times CPU reduction.

In addition to their obvious use within customer-written applications, these enhancements have also been incorporated into IBM-supplied utilities and sample code, and are also used during distributed data access.

***The bottom line**: Multi-row insert and fetch is a significant performance enhancement that can potentially benefit many types of application, with CPU savings of up to 50% for some operations. Distributed applications will see even bigger improvements due to the ability to replace multiple (and time consuming) trips across the network with a single call.*

Optimisation Enhancements

The optimiser is the component responsible for selecting the most efficient way to execute a given SQL statement. DB2's industry-leading optimiser has been further enhanced in a number of areas within Version 8, including:

- **Unlike data type handling**. One of the traditional DB2 performance pitfalls often encountered by developers involves the use of unlike data types for SQL statements. For example, a decimal value may be specified in an SQL statement, and compared to an integer value in a table. DB2 is able to execute this SQL statement, but performance suffers due to the conversion overhead and DB2's inability to efficiently use indexes for such queries. While it is good practice to code SQL queries in such a way as to avoid this problem, it is not always possible (for example, Java does not support fixed length character strings, while this is a data type commonly used within DB2 tables). Version 8 becomes much more tolerant of such mismatches, allowing the query to be processed more efficiently.

- **Enhanced distribution statistics**. DB2's ability to collect data statistics has been enhanced in Version 8. This allows DB2 to make better decisions on how to access the data, thereby improving the performance for many queries.

- **Parallel Sort**. Prior to Version 8, DB2 would always attempt to do as much sorting in parallel as possible. In some situations (for small sorts, for example) it is more efficient not to perform parallel sorts. The Version 8 optimiser makes an intelligent decision as to whether to parallelise the sort or not, based upon the amount of data to be sorted. This can improve the CPU utilisation for certain queries.

***The bottom line**: DB2's industry-leading SQL optimiser has been further enhanced in Version 8, offering more efficient query execution and lower CPU costs for many common queries.*

Materialised Query Tables

In a decision support / data warehousing environment, a common query requirement is to summarise the information in a detail table to a higher level. For example, a query running against a table containing sales detail data for a retail company may present a summary of sales broken down by region.

Such summarisation queries can be expensive to run, as they involve scanning large amounts of data in the detail table. Where these queries are executed frequently, the cost becomes correspondingly higher.

This feature seeks to address this issue by building the summarised result data only once, and storing it in a special table known as a Materialised Query Table (MQT). The next time a user executes a query against the detail table that can be more efficiently satisfied by the MQT, DB2 will automatically rewrite the query internally to execute against the MQT instead. No user intervention is required – the only impact upon the user will be a response time that may be 10 to 1,000 times faster than they are used to seeing.

***The bottom line**: MQTs have already proven their worth in DB2 for Linux, Unix & Windows, and are a powerful means of dramatically improving query response times and reducing resource consumption without any effort on behalf of the query user. A very valuable feature for many decision support / data warehousing environments.*

Other Performance Enhancements

A number of other performance enhancements are introduced in Version 8, including:

- **Partitioning enhancements.** The ability to separately specify the partitioning attribute and the physical order of the data (see "Partitioning Enhancements" on page 23) can yield very significant performance benefits for certain queries (up to 20 times CPU reduction in some tests).

- **Bi directional indexes.** Prior to Version 8, DB2 was able to read indexes in the forward direction only. This would occasionally mean that two nearly identical indexes would have to be defined on some tables, differing only in the order in which the index keys were stored. Version 8 allows these redundant indexes to be removed, as it is able to read an index in both the forward and reverse directions. This saves DASD space and avoids the CPU and elapsed time overheads of having to maintain these redundant indexes for insert, update and delete operations.

- **Data sharing locking**. An enhancement to the way in which locking is handled in a data sharing environment allows more lock requests to be managed by the local Version 8 system instead of being passed to the Coupling Facility. In IBM tests, this has resulted in a 6% reduction in overall CPU time for transactions in IBM's Relational Warehouse Workload[17].

- **Enhanced dynamic statement caching**. The dynamic statement cache allows DB2 to remember the access path it selected for previously executed dynamic SQL statements. Improvements to the way in which memory is allocated have resulted in significant improvements in transaction throughput – up to 45% in some IBM tests.

- **Page fixing buffer pools**. The buffer pools are DB2's main data cache, and Version 8 introduces some enhancements that allow DB2 to utilise the storage more efficiently. In tests, this has resulted in a 7% overall CPU time reduction for the standard IBM Relational Warehouse Workload.

***The bottom line**: Together, these performance enhancements provide considerable scope for reducing the CPU requirements and increasing the throughput of customer workloads.*

Developer / DBA Productivity

As the costs of hardware and software continue to decrease, IT staff costs are becoming relatively more expensive. Improving productivity and managing more systems with the same number of people is therefore a key requirement if IT is to continue to deliver value to the business.

IBM continues to provide additional features aimed at improving the productivity of DB2 application developers and database administrators. New productivity features within Version 8 include:

- **Dynamic schema change**. The dynamic schema enhancements described on page 21 dramatically reduce the amount of effort required to make changes to the structure of DB2 objects. In environments with a large amount of application change, this enhancement will allow a given DBA to support and administer many more databases than was previously possible in a Version 7 system.

- **Automatic secondary space allocation**. Space management – the process of monitoring the amount of disk space used by tables and manually resizing them when necessary – is a major part of the effort expended by a database administrator on a day-to-day basis. This enhancement allows the administrator to instruct DB2 to automatically deal with tables that are nearing their maximum size. While this does not completely free the administrator from all space management responsibilities, it will significantly reduce the amount of time required in managing a busy development or production environment. For environments with less proactive space management policies, this feature will also reduce application failures and data unavailability due to "out of space" conditions being encountered.

- **Partition rebalancing**. New capabilities within the data reorganisation utility allow the administrator to evenly rebalance the amount of data in each

[17] Using 2-way data sharing.

partition of a partitioned table with a single command. This would previously have required a large degree of manual effort on behalf of the administrator.

- **Intelligent utility defaults**. The DB2 data housekeeping utilities have been given more intelligent default values, allowing efficient processes to be defined with less manual tuning effort being required by the database administrator.

- **System level backup and recovery**. The facilities described on page 24 allow entire DB2 systems to be backed up and recovered using a small number of commands. As well as the obvious advantages for system and disaster recovery scenarios, these facilities also offer some useful productivity savings in development environments, where they can be used to very rapidly clone development systems – a task that can otherwise require many hours of dedicated effort to accomplish.

- **Recursive SQL**. Many applications require the ability to recursively execute an SQL statement against a given table (commonly known as a "bill of materials" queries). Previously, such requirements could only be satisfied though relatively complex application coding and/or database design techniques, which required a large amount of developer and administrator effort to build and maintain. Version 8 introduces support for recursive SQL, allowing such queries to be handled directly by DB2.

- **Sequences**. A common application requirement involves the generation of a unique number, which is often used as the key when generating a new row in a table (an invoice number, for example). Although Version 7 supported this requirement via a feature known as Identity Columns, there were a number of operational issues with this implementation which restricted their use. Version 8 introduces Sequences, a considerably more flexible option that will allow the automated generation of unique numbers and therefore deliver developer productivity gains.

In addition to these new features, the DB2 Tools[18] from IBM have been extensively enhanced to exploit the new release, delivering further productivity enhancements for DBAs and developers so that they can focus on higher-value tasks and help their companies increase the return on their data centre investment.

***The bottom line**: Version 8 introduces a number of very significant new features that will allow both DB2 administrators and application developers to become more productive in their day-to-day roles. With staff costs consuming an increasingly significant proportion of today's IT budgets, these enhancements could be an important factor in any cost justification for the move to Version 8.*

Global Integration and Deployment

An On Demand business is blind to national boundaries, and in today's global economy an increasing number of DB2 customers have IT systems which are used all around the world. This globalisation introduces some obvious availability challenges due to time zone differences, but there are some more fundamental issues to be faced in the way in which individual characters are represented on the client systems spread around the various geographies.

[18] A comprehensive set of separately-chargeable tools that are designed to complement DB2's core capabilities to cater for special requirements (such as very large enterprises and complex configurations). See http://www.ibm.com/software/data/db2imstools/ for more information.

Traditionally, each locale had its own way of encoding the characters within its alphabet into a numeric representation that could be stored on a computer. These encoding schemes, known as code pages, often conflict with one another – especially when it comes to "non-standard" characters. For example, 192 represents the "{" symbol using the standard US code page, but that same number represents the "ä" symbol in the standard German code page. These conflicts can often result in incorrect results being displayed on 3270 clients that have not been properly configured. Forcing the use of a single code page on all of these clients is an option, but does not take into account the need for local users to access their specific national language characters.

Unicode was introduced to solve these problems, by providing a representation for every character regardless of the platform and language, thereby avoiding the sort of conflicts described above. The most current version of this standard (Version 4.0) contains 96,248 distinct characters, and this will undoubtedly continue to grow in the future.

Version 7 introduced initial support for Unicode, allowing the data in tables to be stored in Unicode format rather than the more traditional mainframe EBCDIC encoding schemes. This provides benefits for multi-national customers wishing to store data from different geographies in the same table, but some challenges remain (such as complications with SQL statements executed against the Unicode tables, and the inability to join Unicode tables with tables that use a different encoding scheme).

Version 8 includes significant enhancements to the Unicode support initially implemented within DB2 Version 7, removing many of the restrictions inherent in that release.

These changes include:

- **Multiple encoding schemes within a single SQL statement**. Version 8 allows a single SQL statement to join tables with different encoding schemes. This allows the encoding scheme for each table to be chosen for performance reasons on a case-by-case basis, without compromising the functionality of existing applications.

- **SQL processing in Unicode**. This enhancement allows SQL statements themselves to include Unicode references, as the SQL is converted to Unicode internally before processing.

- **DB2 Catalog in Unicode**. As SQL statements will now be converted to Unicode prior to processing, most of the information in DB2's internal catalog tables is also converted to Unicode as part of the migration to Version 8.

***The bottom line**: Version 8 builds upon the initial Unicode support implemented within the previous release, removing many of the restrictions that were preventing customers from fully exploiting Unicode's considerable benefits in a multi-national application environment.*

Data Security

If data is one of your organisation's most valuable assets, then protecting and securing that data must be one of your most important priorities. DB2 has always had a very robust security implementation, which is generally well suited to the requirements of traditional OLTP workloads.

However, many of today's On Demand applications need a more flexible approach. In particular, a common requirement is to limit access to specific rows in a table. DB2's traditional security mechanisms are not able to satisfy this requirement directly, and although some workarounds are possible they invariably involve a degree of compromise.

Version 8 introduces a comprehensive overhaul of DB2's approach to security, with the option to implement "multi-level security" as an alternative to DB2's traditional approach. Multi-level security is based upon facilities within RACF V1.5. and provides the flexibility to secure DB2 resources at any required level, from the entire table down to individual rows. The new option also provides sophisticated hierarchical security capabilities. For example, access controls for a payroll table may be easily defined such that an employee can see only his or her own data, but their manager (one level up in the security hierarchy) is able to see all of the row for his/her department.

The bottom line: Version 8's multi-level security features fundamentally change the way in which DB2 access privileges are managed. Implementing them will require careful planning and analysis, but the vastly increased flexibility will make the effort worthwhile for those customers who require an even greater level of control and capability in security management and access to data.

Version 8 Business Benefit Summary

In this section, we will summarise the major business benefits that Version 8 has to offer. It should be noted that not all of these benefits will apply to every customer, but most should see significant improvements in many of the areas that follow.

Scalability & zSeries Synergy

Feature	Business Benefit	Page
64-bit support	Version 8 allows customers who have invested in IBM's latest generation of hardware to fully exploit the immense storage capabilities of the 64-bit memory model, with no changes being required to their applications. This in turn allows DB2 systems to cache significantly more data and to vertically scale much higher than was previously possible, resulting in a reduction in system overheads and more efficient use of limited machine resources. In some cases, these enhancements may be sufficient to allow customers to defer the implementation of data sharing, if this was being considered purely for scalability reasons. Although customers can expect some increase in CPU requirements following the move to a 64-bit environment, these increases are expected to be offset by the significant performance and productivity enhancements within the new release.	15
Partitioning Enhancements	Version 8 provides some major enhancements to the partitioning options available to DB2 database designers, allowing them to create more efficient designs that are able to support higher transaction loads for a given amount of machine resource. These benefits can be implemented on existing database designs with minimal loss of availability, due to the new dynamic schema change capabilities.	18
Recovery Log Enhancements	Version 8 allows significantly more recovery log data to be held both online and offline, allowing a DB2 system to handle larger workloads while maintaining the recoverability of the data.	19
Other zSeries Synergy	Version 8 exploits specific features in z/OS and the zSeries hardware to deliver highly efficient character conversion and encryption services.	20

Data Availability

Feature	Business Benefit	Page
Dynamic Schema Change	Version 8 provides a good initial set of capabilities for changing database structures while retaining application access to the data. Many customers will be able to use these features to implement changes more frequently, allowing them to react more rapidly to changing market conditions or unexpected performance issues. Database administrator productivity will also be improved (and the risks associated with the change reduced), due to the significantly simpler change process.	21
Partitioning Enhancements	The new partitioning model introduced in Version 8 allows true partition independence for the first time. This reduces the availability impact of table reorganisations, allowing the administrator to keep the data better organised and therefore potentially improving the performance of the application.	23
Dynamic System Parameters	The additional dynamic system parameters in Version 8 decrease the likelihood of the entire DB2 system having to be made unavailable while configuration parameters are changed, which improves overall availability and allows system administrators to be more responsive to changing application demands.	23
System Wide Backup & Recovery	Provided you have the right DASD subsystem, the new system-wide backup and recovery capabilities can reduce the time that DB2 data has to be unavailable due to system backups. Perhaps more importantly, the ability to rapidly restore a system to a given point in time means that applications can be made available again in the minimum possible time in the event of a major problem.	24

Application Porting

Feature	Business Benefit	Page
SQL Enhancements	Together, these SQL enhancements reduce the effort, complexity, risk and cost of porting database definitions and application code to DB2 Version 8, making it an even more attractive platform for vendor application porting infrastructure consolidation projects and development on Windows, Unix or Linux.	25
DB2 Universal Driver	The new Universal Driver removes yet another set of code changes needed when porting applications to the zSeries by delivering a single driver on all platforms (including DB2 for z/OS V7 and V8, while also providing useful performance and functionality enhancements.	26

Web Enablement

Feature	Business Benefit	Page
Java Enhancements	The new Java function within the DB2 Universal Driver provides some valuable productivity and performance enhancements, consolidating Java's position as a primary zSeries programming language.	27
XML Enhancements	Although the XML Extender will still be required to decompose and store XML data in DB2, Version 8's built-in XML publishing functions offer some welcome performance improvements and point the way for more extensive native XML support in the future.	28

Performance & Operating Costs

Feature	Business Benefit	Page
Multi-Row Fetch & Insert	Multi-row insert and fetch is a significant performance enhancement that can potentially benefit many types of application, with CPU savings of up to 50% for some operations. Distributed applications will see even bigger improvements due to the ability to replace multiple (and time consuming) trips across the network with a single call.	29
Optimisation Enhancements	DB2's industry-leading SQL optimiser has been further enhanced in Version 8, offering more efficient query execution and lower CPU costs for many common queries.	30
Materialised Query Tables (MQTs)	MQTs have already proven their worth in DB2 for Linux, Unix & Windows, and are a powerful means of dramatically improving query response times and reducing resource consumption without any effort on behalf of the query user. A very valuable feature for many decision support / data warehousing environments.	31
Other Performance Enhancements	Together, these performance enhancements provide considerable scope for reducing the CPU requirements and increasing the throughput of customer workloads.	31

Developer / DBA Productivity

Feature	Business Benefit	Page
Productivity Enhancements	Version 8 introduces a number of very significant new features that will allow both DB2 administrators and application developers to become more productive in their day-to-day roles. With staff costs consuming an increasingly significant proportion of today's IT budgets, these enhancements could be an important factor in any cost justification for the move to Version 8.	32

Global Integration & Deployment

Feature	Business Benefit	Page
Unicode Support	Version 8 builds upon the initial Unicode support implemented within the previous release, removing many of the restrictions that were preventing customers from fully exploiting Unicode's considerable benefits in a multi-national application environment.	33

Data Security

Feature	Business Benefit	Page
Row Level Security	Version 8's multi-level security features fundamentally change the way in which DB2 access privileges are managed. Implementing them will require careful planning and analysis, but the vastly increased flexibility will make the effort worthwhile for those customers who require an even greater level of control and capability in security management and access to data.	34

A DB2 for z/OS Version 8 Customer Case Study: Univar USA

Univar USA is a leading chemical distributor in the United States, providing more chemicals and related chemical distribution services than any other company in the marketplace.

Until recently, Univar's IT environment was based upon a combination of technologies: a number of Unix servers ran Oracle applications and provided a Business Information environment, while an IBM 9672 RB6 mainframe hosted more traditional VSAM applications (see Figure 12 below).

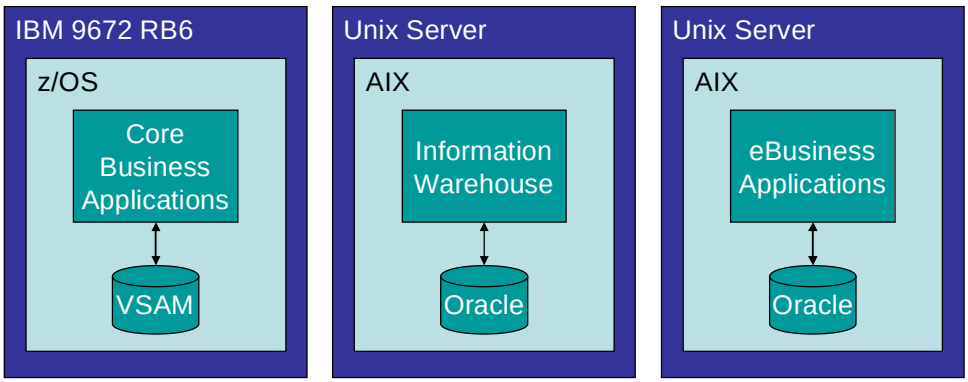

Figure 12 – Previous Univar IT Environment

"For some years we had wanted to convert our existing VSAM applications to use DB2." said Kevin Campbell, Univar's Application Architect and the driving force behind the Version 8 migration project. "Our reasons were primarily concerned with availability and easier integration with external applications, which were pressing enough issues for us but made for a very difficult ROI presentation. There just were not a lot of hard dollar savings that we could really guarantee to offset the DB2 licence fees."

As far as Unix was concerned, Univar had long recognised that their mainframe environment offered superior systems management and a cheaper per-transaction cost than any of their other platforms, but the costs associated with rewriting their Unix/Oracle applications and porting them to the zSeries had prevented them from making the move. "We have a substantial, mature Information Warehouse running on Oracle, which we had always shied away from migrating because of the difficulties associated with changing all of our OLAP infrastructure to accommodate shorter column names. We also have an eCommerce application responsible for many millions of dollars per year in revenue that used Oracle, and this posed similar porting hurdles."

DB2 for z/OS Version 8 changed all that. With its support for longer column names and larger SQL statements, Version 8 offered the potential to dramatically reduce both the costs and the risks of porting Univar's applications, and made it possible for the company to contemplate an ambitious server consolidation project onto the zSeries platform. At the same time, the existence of DB2 on the mainframe would allow them to convert their VSAM applications and realise the integration and availability improvements they needed for their critical line-of-business applications.

"With the introduction of DB2 Version 8 we saw our opportunity: we could gain the long term availability and stability benefits for our core transaction processing system and realize hard cost savings by moving our Oracle applications to DB2 for z/OS. In the process we would also substantially streamline and simplify our computing environment, and bring the easy recoverability of the mainframe to a broader

audience." said Kevin. "The support for longer column names is easily the single most valuable Version 8 feature for us. It significantly reduced both the effort and the risks associated with porting these applications to the mainframe, allowing us to consolidate multiple Unix images into a single z/Linux environment." (see Figure 13 below).

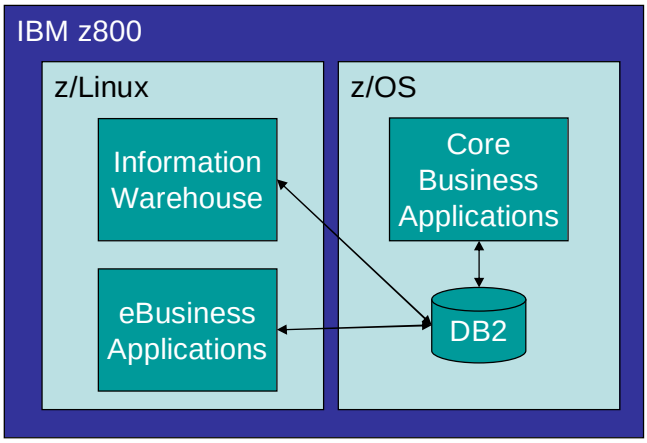

Figure 13 – New Univar IT Environment

In order to assess the viability of the move, Univar participated in IBM's Version 8 early release programme. "This allowed us to get some very valuable Version 8 experience, and validate the feasibility of porting the Unix Applications." said Kevin. Univar's experiences during the early release programme were positive, and convinced them to go ahead with the migration to DB2 Version 8.

The first step in this process happened in early April 2004, and consisted of upgrading the mainframe to a 64-bit capable eServer z800. Kevin was extremely pleased with the upgrade. "Our Technical Services group managed to complete the whole process in a matter of hours. In fact, the only thing that our users noticed was the improved response time the next day!"

With the new hardware in place, Univar's Technical Services team was able to begin constructing the new environment in readiness for the application related work. The beta system was replaced with the final Version 8 code just eight days after it became generally available, making Univar one of the first customers to formally adopt the new release. With help from the DB2 labs, the system was installed and ready to go in just 2 days.

A team of 25 developers then began the task of converting Univar's VSAM applications to use DB2, with the VSAM files being migrated to a DB2 for z/OS database by the 3-person DB2 administration team. At the time of writing, Univar are in the final stages of stress testing the converted applications, prior to moving them into the Production environment in September 2004.

The final piece of the puzzle will be put in place towards the end of 2004, when work will begin on porting the Unix applications from Oracle to DB2 in order to realise the licence and infrastructure cost savings that will fund the entire project. Kevin is looking forward to the impact this will have on his DB2 administration team. "The move to DB2 will free the DBAs from having to worry about lower-level Unix system management activities such as manual backups" said Kevin. "They will then have additional time to use their skills more effectively and add value to the organisation by getting more involved with things like optimisation and SQL tuning."

Does Kevin see any other Version 8 benefits in addition to the application porting enhancements? The dynamic schema change facilities will be a key feature. Many of the applications to be ported are e-business systems with no planned downtime, and

the ability to change database structures on the fly is very important to us" he says. Although Univar is a multi-national organisation, each country tends to use its own IT systems so Unicode support wasn't a major issue for them. "However, we're likely to want to consolidate the data across those international systems for reporting and analysis in the future" says Kevin "so the Unicode support could prove to be very valuable to us."

So the move to Version 8 made good technical sense to Univar, but what about that cost justification? "For us, the decision to move to DB2 Version 8 was primarily a financial one." said Kevin. "We're able to make significant savings in three areas as a result of the move. Firstly, we'll be eliminating the Oracle licence fees and the infrastructure support costs for all of those Unix servers. Secondly, we'll avoid having to renew the leases on the Unix servers when they expire. Finally, we've substantially reduced the effort and costs associated with populating our Information Warehouse, as this will become a DB2 to DB2 process rather than a VSAM on z/OS to Oracle on Unix process. In combination, these savings will allow us to rapidly recoup our investment in moving to DB2."

Justifying The Upgrade

This section of the paper will present some sample scenarios to illustrate the cost/benefit case for the migration from Version 7 to Version 8. We will focus on the quantifiable financial costs and benefits associated with the upgrade, ignoring the many strategic or non-quantifiable benefits that have been previously described.

In an effort to address the major types of customer considering a Version 8 migration, we will cover three separate scenarios:

- An OLTP customer, running a traditional CICS/DB2 workload
- An SAP customer, using DB2 for z/OS as the back-end database server for an ERP workload only
- A Java customer, running a high-volume web-based transaction workload

These are defined in more detail in the sections that follow, together with the estimated costs and benefits for each customer type.

It should be noted that while these scenarios use real hardware and software costs, they are by necessity extreme generalisations and are based on a number of factors and assumptions that will vary significantly from customer to customer. However, the models presented should be of use to many customers in providing an outline of the potential financial position.

Sample Customer Environment

For the purposes of our examples, our customer is running an eServer z990[19] rated at approximately 330 MSUs[20] (e.g. a Model 2064-2C8) running z/OS 1.4. The total operating cost for this system (including hardware, operating system, middleware and ISV software) is $5,573,000 per year.

This customer is currently paying $410,000 per annum in PSLC[21] licence fees for DB2 Version 7 (included in the ownership cost figure above), and will pay an additional $61,000 per annum following the upgrade to Version 8.

In addition, due to the overheads associated with the move to a 64-bit environment, this customer will experience a 7% increase in CPU requirements, amounting to an additional annual cost of $387,000 per year[22].

Staff costs for the upgrade are estimated as follows:

- Planning (per customer) – 20 days systems programmer effort, 20 days DBA effort, 5 days application developer effort.
- Implementation (per DB2 system) – 5 days systems programmer effort, 10 days DBA effort, 10 days developer effort.

This customer uses a nominal internal cost of $60 per hour for all staff when calculating project costs and benefits.

[19] Customers intending to move to a z990 platform as part of the migration should consider the additional impact of the "technology dividend" announced by IBM in September 2003. This benefit has **not** been included in the scenarios that follow, as it has been assumed that the sample customers are already on the z990 platform prior to the Version 8 migration.

[20] MSU - Million Service Units. Performance/capacity ratings assigned to IBM S/390 and zSeries systems..

[21] PSLC – Parallel Sysplex Licence Charge.

[22] Calculated by taking 7% of the system total ownership cost ($5,573,000) with the disk storage costs ($46,000) subtracted, as they are not impacted by move to Version 8.

Scenario 1: OLTP Customer

In this scenario, we will consider a customer running a traditional OLTP workload consisting of high-volume CICS/DB2 transactions during an online day, with a 10 hour overnight batch window from 8.00pm to 6.00am each day.

This customer runs three separate DB2 systems – two development systems and a production system on a separate logical partition. In addition to the OLTP workload, the production system also hosts a small Management Information system, using extracts taken from the main operational databases.

Costs

The upgrade costs for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Additional DB2 licence costs	$61,500	$61,500	$61,500
CPU costs – performance regression	$386,890	$386,890	$386,890
Staff Costs – Upgrade	$57,600	$0	$0
TOTAL	**$505,990**	**$448,390**	**$448,390**

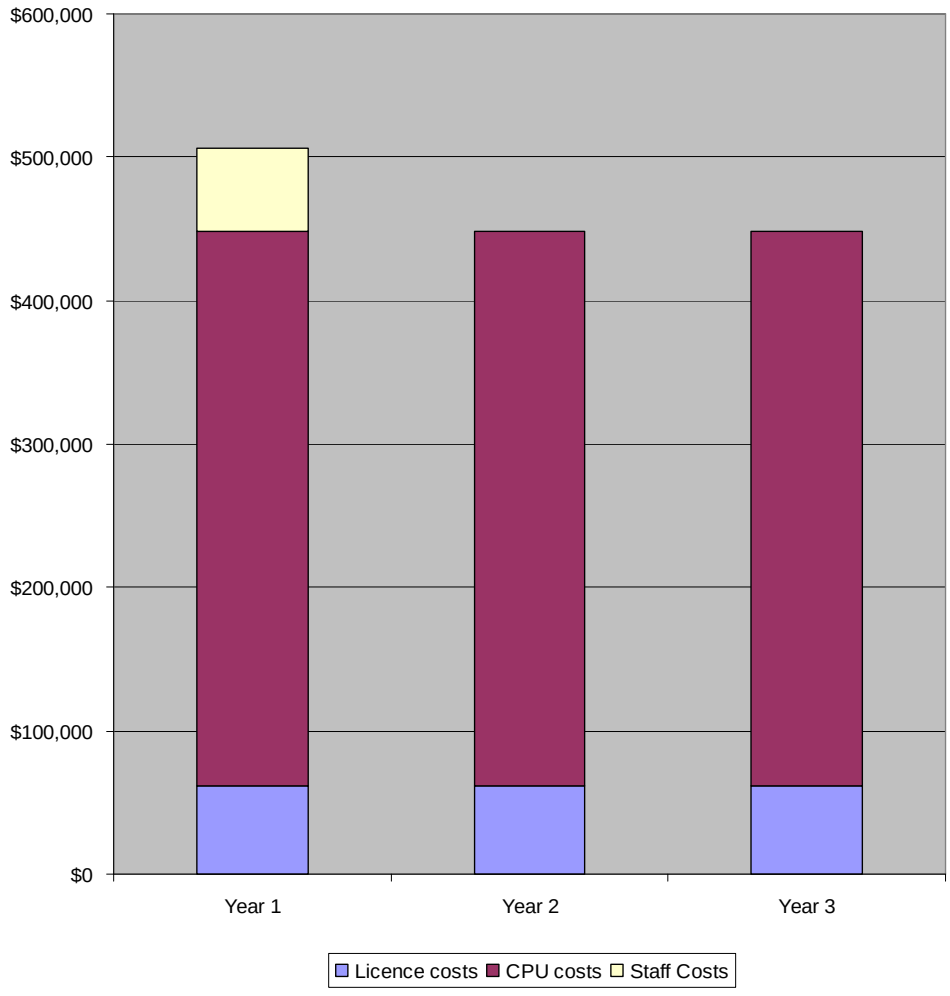

Figure 14 – OLTP Customer Upgrade Costs

Notes

- No hardware costs, as customer is already on a 64-bit capable server
- DB2 licence and 64-bit regression CPU costs as per calculations shown in "Sample Customer Environment" on page 43.
- Staff costs estimated as per calculations shown in "Sample Customer Environment" on page 43.

Financial Benefits

The financial benefits for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Scalability	$167,190	$167,190	$167,190
Multi-Row Insert/Fetch	$111,460	$111,460	$111,460
Materialised Query Tables	$55,730	$55,730	$55,730
Optimisation Improvements	$111,460	$111,460	$111,460
Staff costs – dynamic schema change	$21,600	$21,600	$21,600
Staff costs – PIT recovery	$2,880	$2,880	$2,880
Staff costs – automated space management	$14,400	$14,400	$14,400
Staff costs – utilities enhancements	$3,600	$3,600	$3,600
Total	**$488,320**	**$488,320**	**$488,320**

Notes

- Scalability savings due to increased throughput and more efficient use of memory available on 64-bit architecture. Estimated at 3% of the total ownership cost..
- Multi-Row Insert/Fetch savings estimated at 2% of the total ownership cost, based on fewer separate instructions and reduced network overhead for remote operations.
- Materialised Query Tables savings due to avoidance of large table scans in Management Information environment. Estimated at 1% of the total ownership cost.
- Optimisation savings due to more efficient access paths chosen by DB2 for existing applications. Estimated at 2% of the total ownership cost.
- Staff cost savings estimated as follows:
  - Dynamic schema change – 30 hours per month (10 changes x 3 hours per change) due to quicker implementation of database structure changes
  - Point in time recovery – 4 hours per month (2 recoveries x 2 hours per recovery) due to quicker and simpler backup/recovery processes
  - Automated space management – 20 hours per month (20 days x 1 hour per day) saved in no longer having to fix regular out-of-space conditions

- Utilities enhancements – 5 hours per month (20 days x 0.25 hours per day) saved in having to tune utilities, due to more intelligent default settings

- No benefits have been claimed within this scenario for the anticipated increase in data availability, but for some customers this will have a quantifiable financial benefit.
- No benefits have been claimed within this scenario for the improved ability to port applications, although customers who develop on Windows, Unix or Linux and then run production on DB2 for z/OS may find some significant financial benefits.

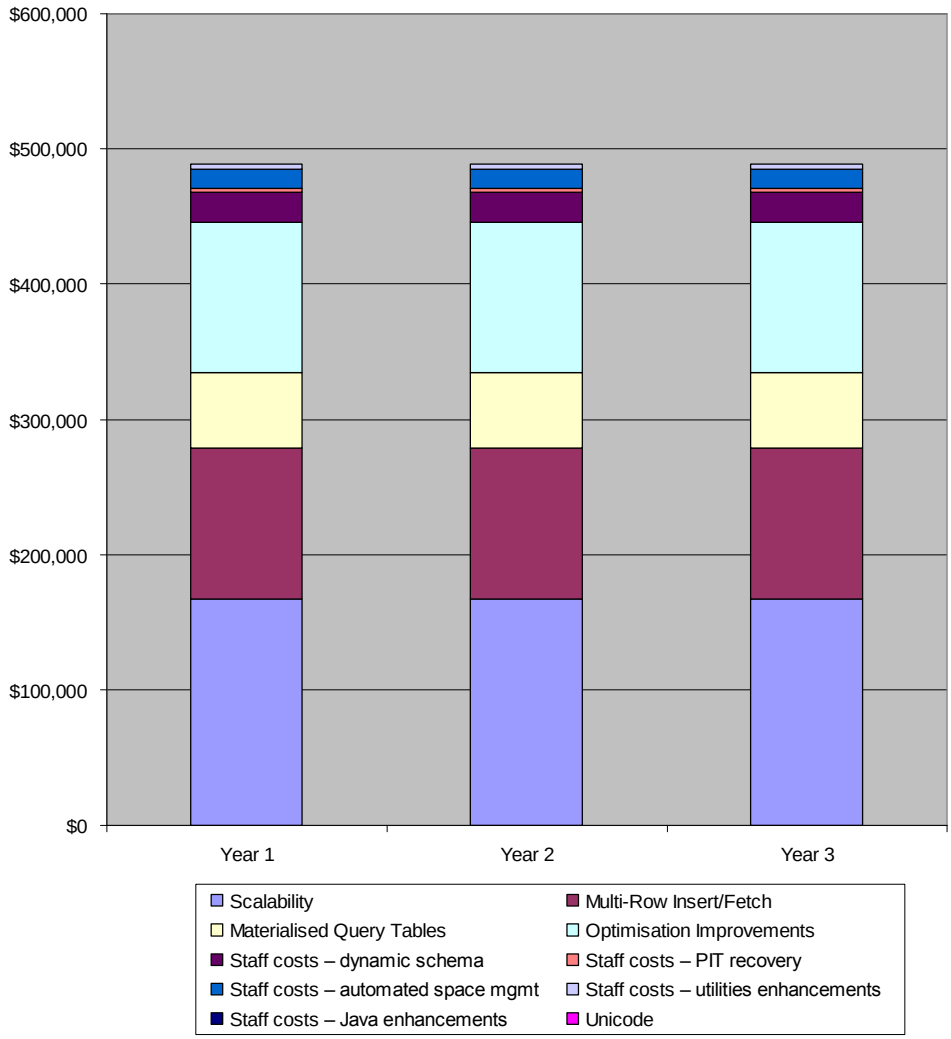

Figure 15 – OLTP Customer Financial Benefits

Return On Investment

The costs and benefits outlined above result in a return on investment within 18 months, as shown in the table below

	Year 1	Year 2	Year 3
Costs	$505,990	$448,390	$448,390
Benefits	$488,320	$488,320	$488,320
Savings	**-$17,670**	**$39,930**	**$39,930**
Overall	**-$17,670**	**$22,260**	**$62,190**

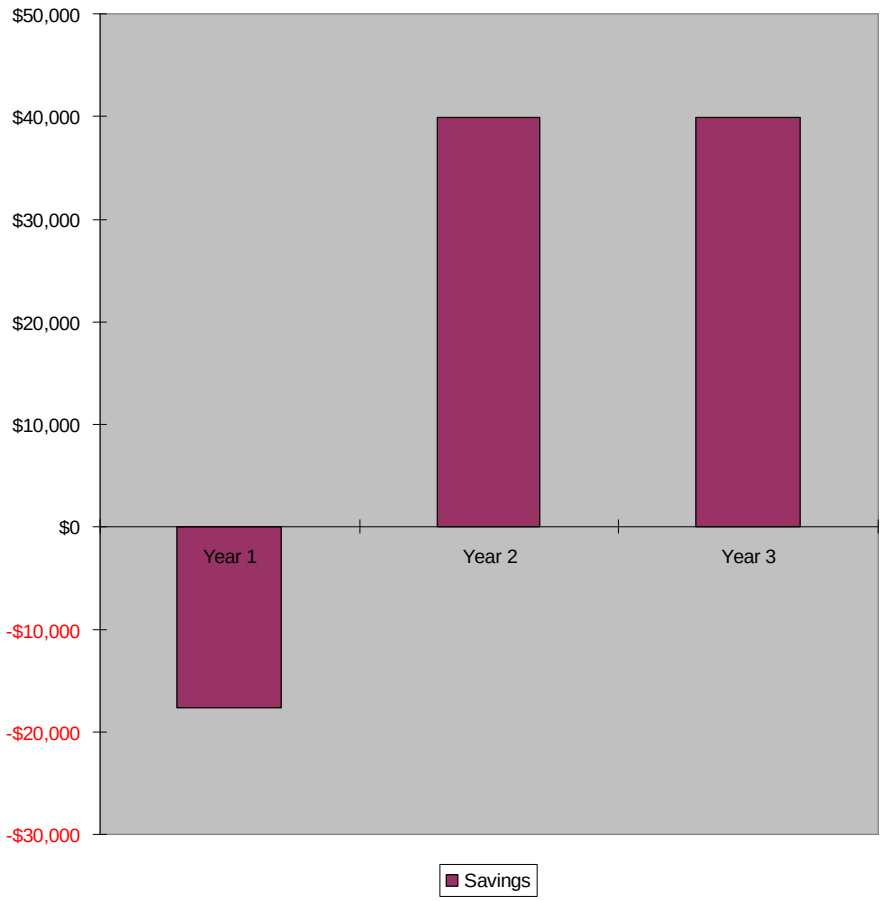

Figure 16 – OLTP Customer Overall Savings

Scenario 2: SAP Customer

In our second scenario, we will look at a customer that is using DB2 purely for the purposes of supporting their zSeries SAP system. The customer makes use of the SAP Business Information Warehouse feature to analyse their vital business information.

This customer runs four separate DB2 systems – two development systems and two production systems on a separate logical partition to support their SAP operational and Business Information Warehouse workloads respectively.

Costs

The upgrade costs for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Additional DB2 licence costs	$61,500	$61,500	$61,500
CPU costs – performance regression	$386,890	$386,890	$386,890
Staff Costs – Upgrade	$69,600	$0	$0
TOTAL	**$517,990**	**$448,390**	**$448,390**

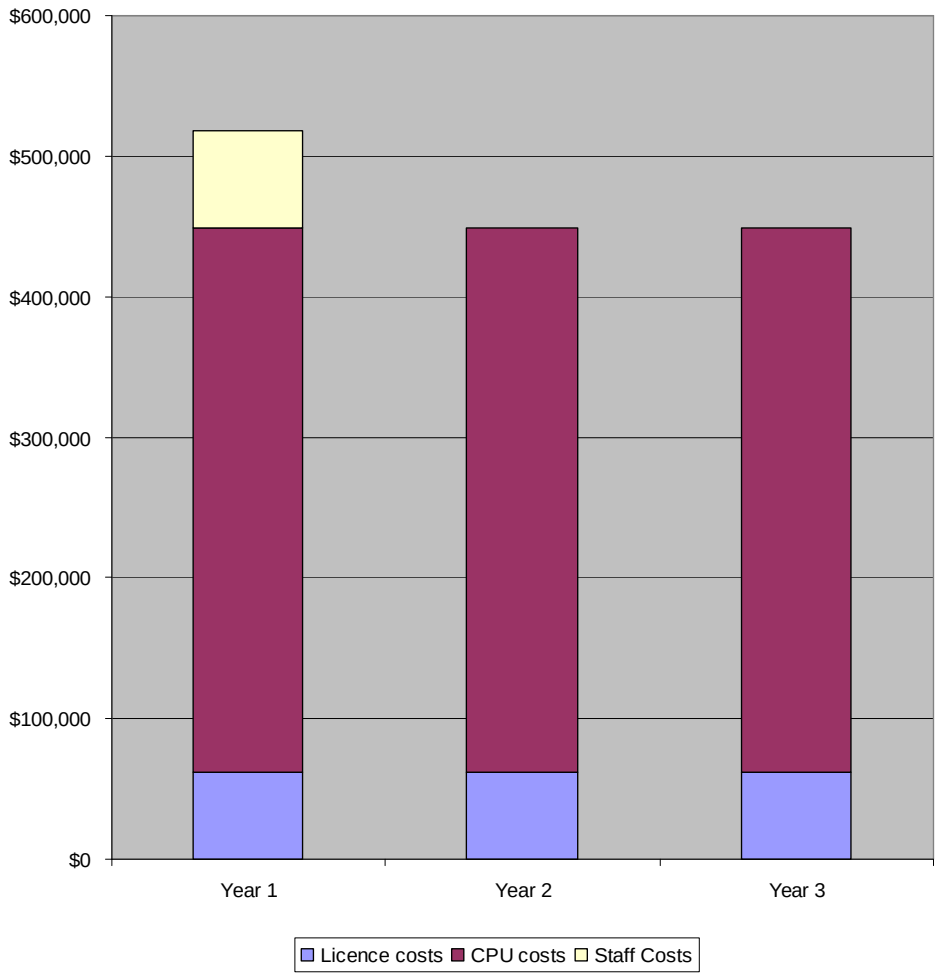

Figure 17 – SAP Customer Upgrade Costs

Notes

- No hardware costs, as customer is already on a 64-bit capable server
- DB2 licence and 64-bit regression CPU costs as per calculations shown in "Sample Customer Environment" on page 43.
- Staff costs estimated as per calculations shown in "Sample Customer Environment" on page 43.

Financial Benefits

The financial benefits for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Scalability	$278,650	$278,650	$278,650
Multi-Row Insert/Fetch	$83,595	$83,595	$83,595
Materialised Query Tables	$5,573	$5,573	$5,573
Optimisation Improvements	$111,460	$111,460	$111,460
Staff costs – dynamic schema change	$5,760	$5,760	$5,760
Staff costs – PIT recovery	$14,400	$14,400	$14,400
Staff costs – automated space management	$21,600	$21,600	$21,600
Staff costs – utilities enhancements	$3,600	$3,600	$3,600
Total	**$524,638**	**$524,638**	**$524,638**

Notes

- Scalability savings due to increased throughput and more efficient use of memory available on 64-bit architecture. Estimated at 5% of the total ownership cost..
- Multi-Row Insert/Fetch savings estimated at 1.5% of the total ownership cost, based on fewer separate instructions and reduced network overhead for remote operations.
- Materialised Query Tables savings due to avoidance of large table scans in Management Information environment. Estimated at 0.1% of the total ownership cost[23].
- Optimisation savings due to more efficient access paths chosen by DB2 for accessing SAP application. Estimated at 2% of the total ownership cost.
- Staff cost savings estimated as follows:
  - Dynamic schema change – 8 hours per month (2 changes x 4 hours per change) due to quicker implementation of database structure changes
  - Point in time recovery – 20 hours per month (10 recoveries x 2 hours per recovery) due to quicker and simpler backup/recovery processes

[23] The SAP Business Information Warehouse already implements the concept of aggregate tables, so MQTs are unlikely to yield much benefit in this environment. However, as aggregates cannot be defined on the main SAP operational data store, some limited benefits may accrue from the implementation of MQTs.

- Automated space management – 30 hours per month (20 days x 1.5 hour per day) saved in no longer having to fix regular out-of-space conditions
- Utilities enhancements – 5 hours per month (20 days x 0.25 hours per day) saved in having to tune utilities, due to more intelligent default settings

- No benefits have been claimed within this scenario for the anticipated increase in data availability, but for some customers this will have a quantifiable financial benefit.

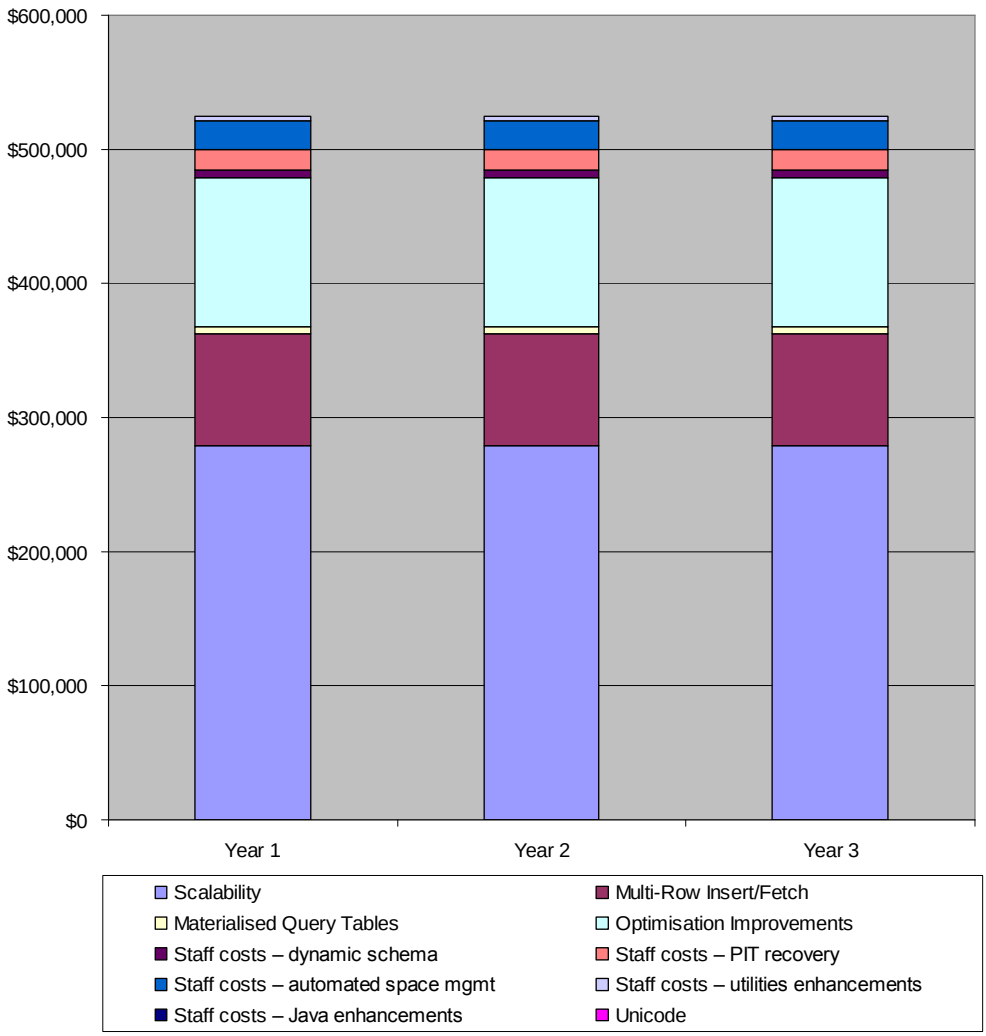

Figure 18 – SAP Customer Financial Benefits

Return On Investment

The costs and benefits outlined above result in a return on investment within 18 months, as shown in the table below

	Year 1	Year 2	Year 3
Costs	$517,990	$448,390	$448,390
Benefits	$524,638	$524,638	$524,638
Savings	**$6,648**	**$76,248**	**$76,248**
Overall	**$6,648**	**$82,896**	**$159,144**

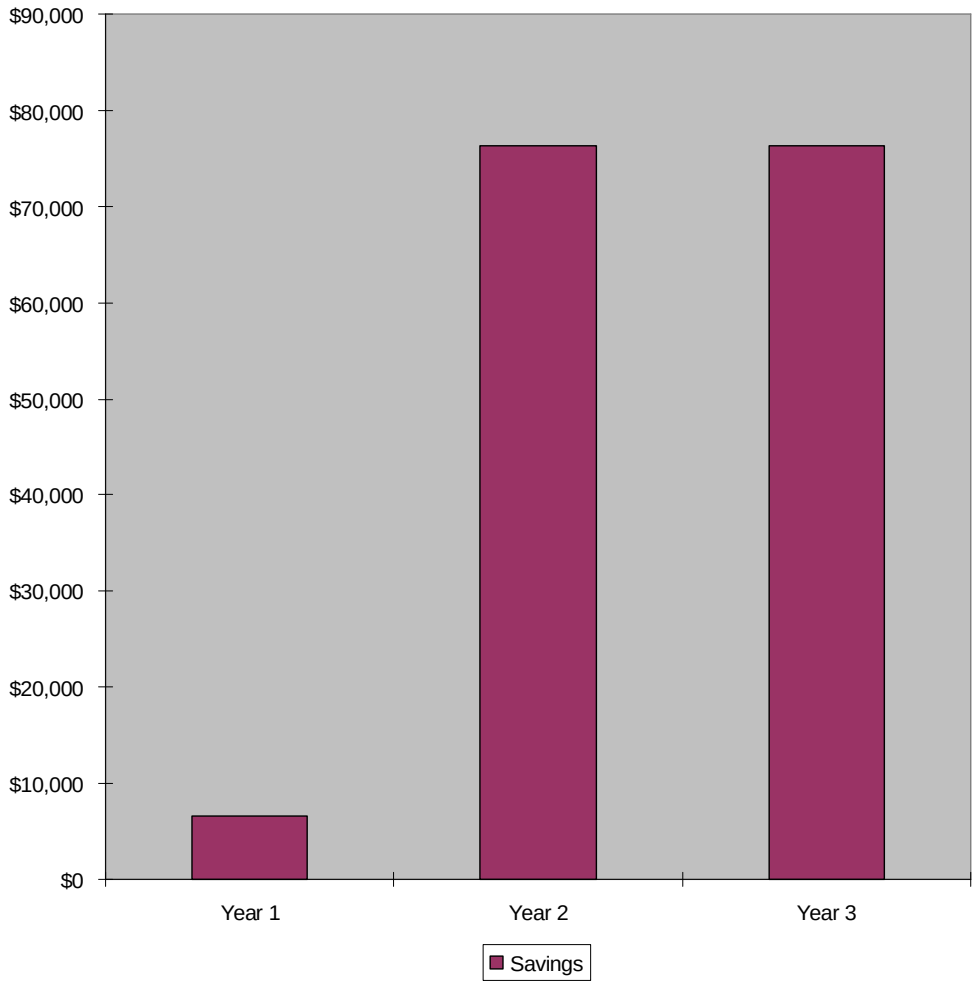

Figure 19 – SAP Customer Overall Savings

Scenario 3: Java Customer

In our final scenario, we will focus on a typical Java customer, who is using DB2's excellent web enablement capabilities to run a high-volume web-based transaction processing system with DB2 at its core.

This customer runs four separate DB2 systems – two development systems and two production systems in a data sharing configuration, to provide additional resilience.

Costs

The upgrade costs for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Additional DB2 licence costs	$61,500	$61,500	$61,500
CPU costs – performance regression	$386,890	$386,890	$386,890
Staff Costs – Upgrade	$69,600	$0	$0
TOTAL	**$517,990**	**$448,390**	**$448,390**

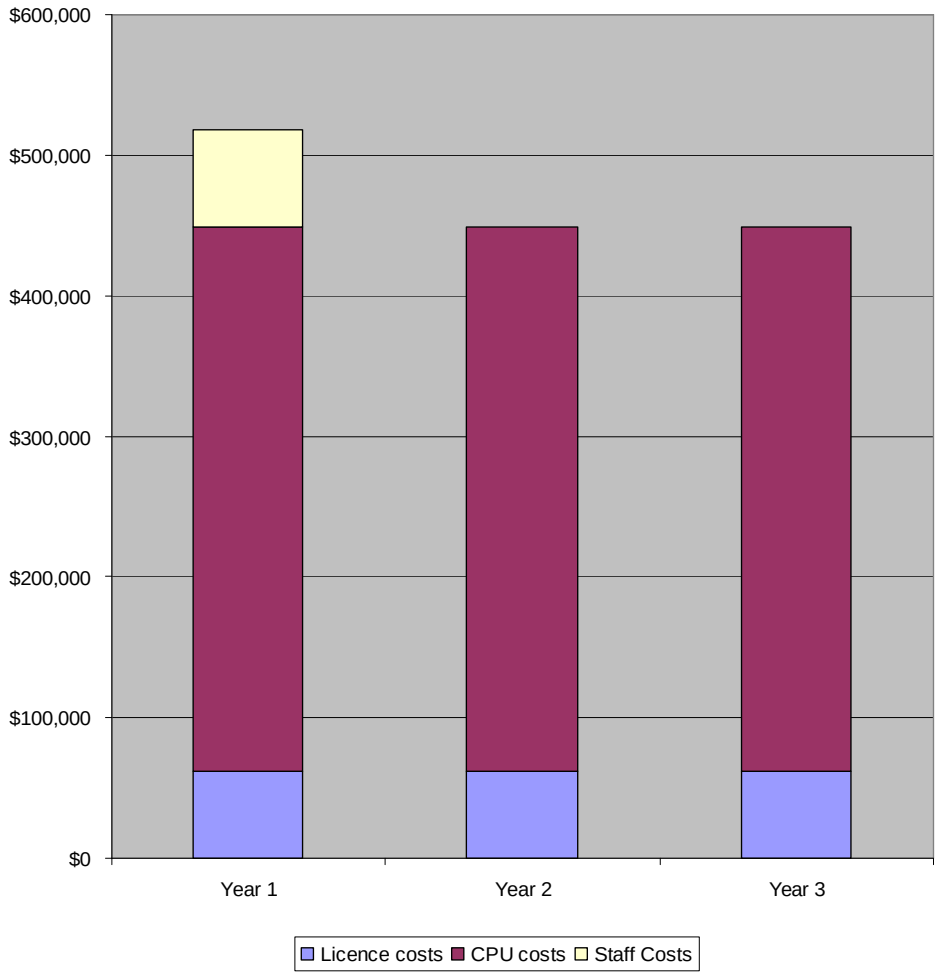

Figure 20 – Java Customer Upgrade Costs

Notes

- No hardware costs, as customer is already on a 64-bit capable server
- DB2 licence and 64-bit regression CPU costs as per calculations shown in "Sample Customer Environment" on page 43.
- Staff costs estimated as per calculations shown in "Sample Customer Environment" on page 43.

Financial Benefits

The financial benefits for this customer can be summarised as follows:

	Year 1	Year 2	Year 3
Scalability	$222,920	$222,920	$222,920
Multi-Row Insert/Fetch	$55,730	$55,730	$55,730
Materialised Query Tables	$0	$0	$0
Optimisation Improvements	$278,650	$278,650	$278,650
Staff costs – dynamic schema change	$43,200	$43,200	$43,200
Staff costs – PIT recovery	$2,880	$2,880	$2,880
Staff costs – automated space management	$14,400	$14,400	$14,400
Staff costs – utilities enhancements	$3,600	$3,600	$3,600
Staff costs – Java enhancements	$14,400	$14,400	$14,400
Total	**$635,780**	**$635,780**	**$635,780**

Notes

- Scalability savings due to increased throughput and more efficient use of memory available on 64-bit architecture. Estimated at 4% of the total ownership cost..
- Multi-Row Insert/Fetch savings estimated at 1% of the total ownership cost, based on fewer separate instructions and reduced network overhead for remote operations.
- Materialised Query Tables savings not applicable to this environment.
- Optimisation savings due to more efficient access paths chosen by DB2 for existing applications. Estimated at 5% of the total ownership cost.
- Staff cost savings estimated as follows:
  - Dynamic schema change – 60 hours per month (20 changes x 3 hours per change) due to quicker implementation of database structure changes
  - Point in time recovery – 4 hours per month (2 recoveries x 2 hours per recovery) due to quicker and simpler backup/recovery processes
  - Automated space management – 20 hours per month (20 days x 1 hour per day) saved in no longer having to fix regular out-of-space conditions

- Utilities enhancements – 5 hours per month (20 days x 0.25 hours per day) saved in having to tune utilities, due to more intelligent default settings

- No benefits have been claimed within this scenario for the anticipated increase in data availability, but for some customers this will have a quantifiable financial benefit.

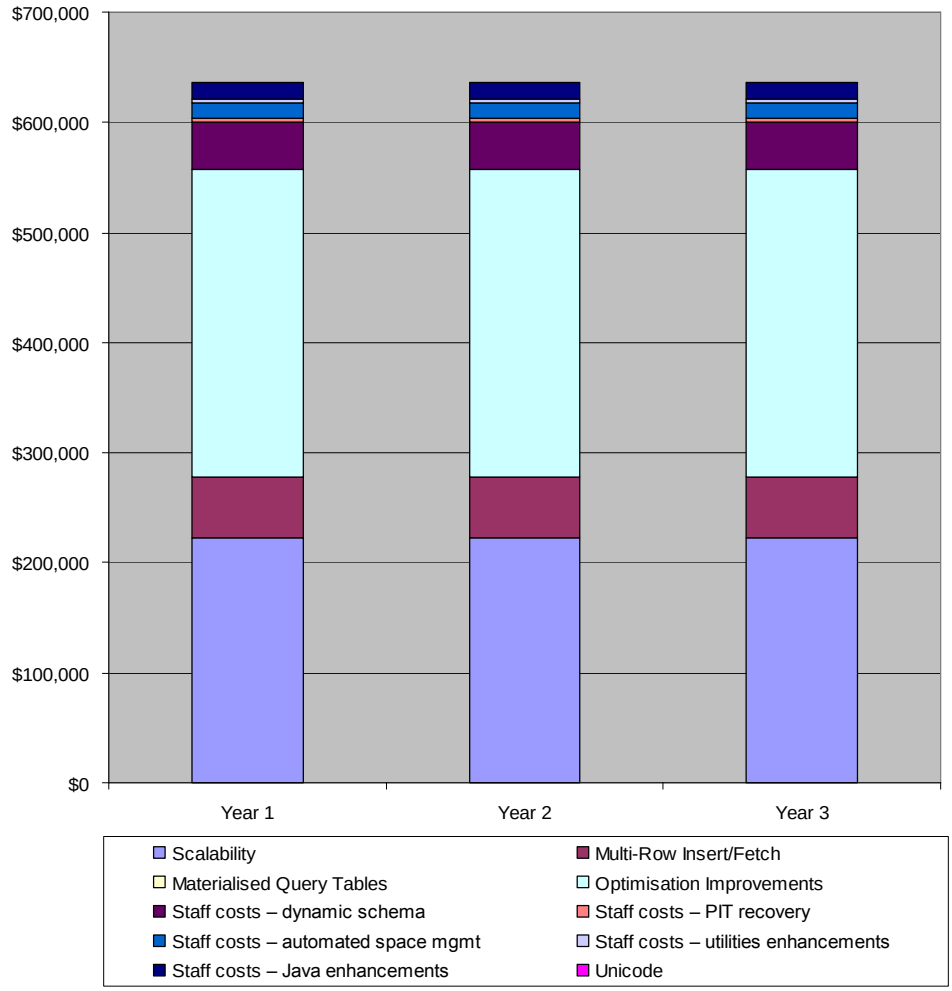

Figure 21 – Java Customer Financial Benefits

Return On Investment

The costs and benefits outlined above result in a return on investment within 18 months, as shown in the table below

	Year 1	Year 2	Year 3
Costs	$517,990	$448,390	$448,390
Benefits	$635,780	$635,780	$635,780
Savings	**$117,790**	**$187,390**	**$187,390**
Overall	**$117,790**	**$305,180**	**$492,570**

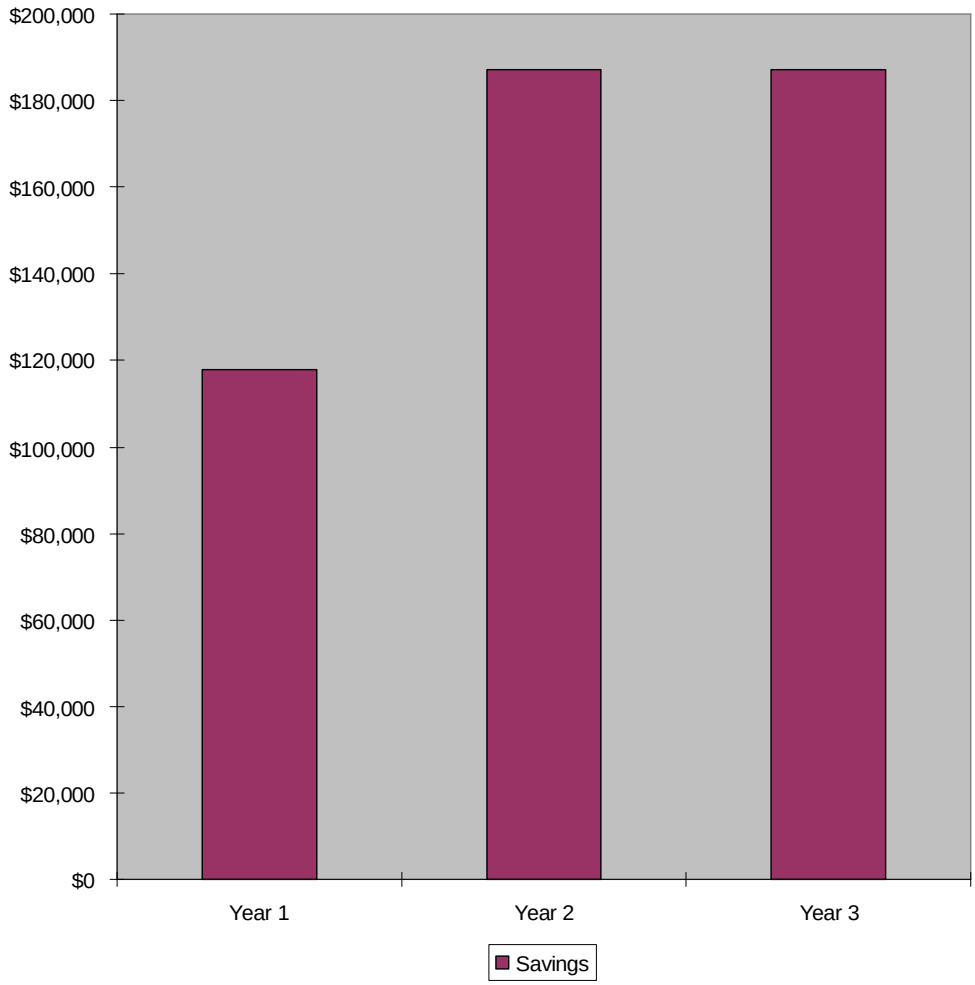

Figure 22 – Java Customer Overall Savings

Summary

DB2 for z/OS Version 8.1 promises to be one of the most significant releases ever for IBM's flagship database management product. It is the first middleware product to break the bonds of 31-bit computing and begin to exploit the enormous scalability within IBM's 64-bit zSeries machines, allowing customers to drive greater workloads through their systems and "do more with less".

Major advances have also been made in improving data availability, further reducing planned downtime and allowing today's 24/7 On Demand applications to stay active for longer. Many customers will be able to use these features to implement changes to their databases more frequently, allowing them to react more rapidly to changing market conditions or unexpected performance issues.

Although the move to a 64-bit environment introduces its own CPU demands, these are likely to be more than offset by the significant performance enhancements in Version 8 – many of which can be realised with little or no application changes being required. The performance enhancements drive down the total ownership cost for DB2 applications, making each transaction cheaper to execute and allowing greater workloads to be handled by a given server.

The new release also removes many of the barriers preventing non-DB2 applications from being ported to the zSeries platform. As shown in the case study on page 40, this can open the way for some compelling cost cases to be built around server consolidation projects.

Finally, significant productivity savings are possible due to the increased automation and intelligence built into the product. This allows skilled DB2 administrators and developers to spend less time performing lower-level routine tasks and more time adding value to the organisation by producing better written, more efficient applications.

In combination, these new features represent a significant advancement in DB2's capabilities, and most customers are expected to see significant financial and technical benefits as a result of migrating to the new release.

Appendix A – References

SG24-6871	DB2 UDB for z/OS Version 8 Technical Preview
GC18-7428	DB2 UDB for z/OS V8: What's New?
SG24-6079	DB2 UDB for z/OS V8: Everything You Ever Wanted To Know…And More
SG24-7088	DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There

Appendix B – Acknowledgements

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

John Campbell	Distinguished Engineer, DB2 for z/OS Development, IBM Silicon Valley Laboratory
Kevin Campbell	Application Architect Univar USA
Jackie Chu	SWG Pricing, IBM Software Group - Data Management Division
Catalin Comsia	Senior IT Specialist, DB2 for z/OS
Curt Cotner	IBM Fellow DB2 for z/OS Development
Malcolm Nolan	Marketing Manager, DB2 for z/OS IBM Software Group - Data Management Division
Ian Cook	DB2 for z/OS Product Introduction Center, Europe
Sandy Smith	Director of Data Management Hewitt Associates
Roger Miller	Senior Strategist DB2 for z/OS Development, IBM Silicon Valley Laboratory
Robert Perih	Manager Database Systems, Citigroup
Mark Wilson	DB2 for z/OS Product Introduction Center, Americas
Maureen Zoric	Program Manager DB2 UDB for z/OS and OS/390