We discuss the latest news about DB2 for z/OS, including the changes in disks, IBM System z9 and z10 Integrated Information Processor or zIIP and new z10 processors. Then the discussion will move DB2 9 for z/OS, with XML and SQL that is more consistent across the DB2 family. Utility enhancements help with new function, more LOB and XML support, better performance and improved availability, removing the BUILD2 step from online reorg. DB2 9 enhances DB2's ability to handle new web and enterprise applications. DB2 9 improves the ability to handle new applications with XML, large objects, and many SQL and security improvements. DB2 9 builds upon and extends DB2 traditional strengths and the ground-breaking V8 in many areas: online schema evolution, Unicode, XML, DB2 family SQL, utilities, security and 64-bit virtual storage.
What's new in DB2 for z/OS?

- Synergy with DB2 for z/OS and System z
- What does DB2 9 have for me?
- What are DB2 9 performance characteristics?
- How is database administration improved?
- What are the improvements for application programming? Is XML for me?
- What is the vision beyond DB2 9? What early planning can I do for DB2 9 and beyond?

We discuss the latest news about DB2 for z/OS, including the changes in disks, IBM System z9 and z10 Integrated Information Processor or zIIP and new z10 processors. Then the discussion will move DB2 9 for z/OS, with XML and SQL that is more consistent across the DB2 family. Utility enhancements help with new function, more LOB and XML support, better performance and improved availability, removing the BUILD2 step from online reorg. DB2 9 enhances DB2's ability to handle new web and enterprise applications. DB2 9 improves the ability to handle new applications with XML, large objects, and many SQL and security improvements. DB2 9 builds upon and extends DB2 traditional strengths and the ground-breaking V8 in many areas: online schema evolution, Unicode, XML, DB2 family SQL, utilities, security and 64-bit virtual storage.
DB2 Deep Synergy With System z

Key integration points include:
- Data sharing (availability and scale out)
- Hardware data compression & encryption
- zIIP specialty engines
- Unicode conversion
- Encrypted TCP/IP communication (SSL), encrypted data
- Cross-memory, memory protection keys
- Sorting
- Multi-core, large N-way
- 1 MB page size (z10)
- Decimal float arithmetic (z10)
- 64-bit addressing and large memory
- z/OS Workload Manager
- z/OS Security Server (RACF)
- z/OS RRS integrated commit coordinator
- Solid state disks

Data sharing is a prime example of deep synergy with System z. DB2 worked with the System z design team for nearly 10 years to produce a robust platform for horizontal scaling. The evolution has continued for 15 more years now after delivery.

Hardware data compression and encryption provides improved costs, easier management and robust resilience for the platform. Cross-memory and protection keys work with APF authorization and RACF for the underlying system integrity.

Specialty engines can reduce costs very substantially, reducing both hardware and software costs.

The z/OS workload manager (WLM) has changed in almost every release to improve work flow with DB2. DB2 has a dispatcher, the z/OS WLM.

Sorting, decimal arithmetic, decimal float, encryption, and Unicode conversions are examples of unique instructions in z/Architecture that DB2 uses.
The design of the IBM System z10™ processor chip is the most extensive redesign in over 10 years, resulting in an increase in frequency from 1.7 GHz (z10 EC) to 4.4 GHz on the z10 EC. It is designed for secure data serving, yet also was enhanced to provide improvement enhances for CPU intensive workloads. The result is a platform that continues to improve upon all the mainframe strengths customers expect, yet opens a wider aperture of new applications that can all take advantage of System z10s extreme virtualization capabilities, and lowest TCO versus distributed platforms.

See the DB2 9 for z/OS Performance Topics book, SG24-7473, for the latest information about DB2 performance on the z10. A range of lab measurements are provided in section 4.3 of the September 2008 update. Watch for more updates.
Software costs are higher than hardware costs for most customers, generally in a range of three to ten times higher. This is the reason that specialty engines are so attractive, as the software running on them does not incur software charges. Unit software costs continue to decline in many ways. Software charges have many options, but some of the basic tenets are consistent for most. System z software is charged for the processing capacity of the machine, taking into account multiprocessor effects, rather than using a linear scale. Larger amounts of processing power reduce the unit cost by more than 90%. The z990, z9, and z10 generations have each reduced the charging units by 10% over the prior generation. Changing from z900 to z10 processors would reduce the software charge units by about 28% for the same processing power. Additional options for reducing software charges include parallel sysplex charging, subcapacity pricing, zNALC pricing, and the new Value Unit Edition. I/O costs include the disk space and the transfer to the processor. Both costs are addressed by the work to use System z effectively. Compression for data uses z/Architecture instructions to be efficient, with a solid track record since 1993 and improved performance in the latest generations. Index compression in DB2 9 adds to the savings.

The key for pricing is the value. DB2 for z/OS delivers a very high quality of service. The service is shared, so that administration time is reduced. The amounts of processing, memory, disk and people can be reduced by effective sharing, delivering the best value for the money.
zIIP can help to integrate data across the enterprise by lowering the cost of ownership for eligible data and transaction processing workloads.

Centralized data serving – First to exploit zIIP were workloads such as BI, ERP, and CRM applications running on distributed servers with remote connectivity to DB2® V8.

Network encryption – zIIP becomes an IPSec encryption engine helpful in creating highly secure connections in an enterprise.

Serving XML data – zIIP is enabled for XML parsing, first to exploit this was inserting and saving DB2 9 XML data over DRDA®. See this paper for more detail. DB2 9 and z/OS System Services Synergy Update, http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101227

Remote mirror – zIIP becomes a data mirroring engine with zIIP assisted z/OS Global Mirror function (zGM, formerly XRC) helpful in reducing server utilization at recovery site (with z/OS V1.8 and above).

Exploiting of zIIPs by ISVs zIIPs offer economics to help you.

The newest changes allow HiperSockets to be use zIIP for large messages, such as DRDA data and for financial reporting.

PLUS zIIP price is same for z10 EC as z9 EC and we offer no charge MES upgrades when moving to new technology.
Why do zIIPs, zAAPs and IFLs Reduce Cost?

1. Hardware costs: By moving workload from general purpose processors to zIIP, zAAP and IFL processors (higher cost to lower cost processors).

2. Software Costs: license/maintenance costs based on number of and usage of general purpose central processors. Specialty engines can reduce number of CP’s.

No z/OS software charges based on zIIP, zAAP and IFL processors or usage.

zIIPs and zAAPs do not add functional capabilities, but they do address hardware and software costs. In addition to being lower cost processors for the specialty purposes, they allow you to reduce the license and maintenance cost for software on z/OS, as there is no z/OS software charge for processing running on zIIP, zAAP or IFL processors.
One of the key initiatives of V8 was online schema evolution, and that theme is expanding and changing to be data definition on demand. These are key improvements for resilience. One of the important changes is to be able to replace one table quickly with another. Another is to be able to rename a column or an index. A new type of table space combines the attributes of segmented and partitioned, without a partitioning key. Rebuild index can be run with much less disruption. Online table space reorganization for a few partitions is improved a lot, removing the BUILD2 phase for all types of secondary indexes. Table space and index logging can be altered. Changing DB2 early code does not require an IPL.

Many other improvements help with performance, with scalability and with availability. Index on an expression can be combined with caseless comparisons to improve text search. Improved insert rates can result from improved latching of the log data. Significant reductions in cpu usage are provided with new utilities.

Today’s complex applications include both transactions and reporting, so performing both well is imperative. The key improvements for reporting are optimization enhancements to improve query and reporting performance and ease of use. More queries can be expressed in SQL with new SQL enhancements. Improved data is provided for the optimizer, with improved algorithms. Improved cpu and elapsed times can be achieved with the FETCH FIRST clause specified on a subquery. The INTERSECT and EXCEPT clauses make SQL easier to write.
This chart shows the relationship of SQL in the DB2 family comparing DB2 for Linux, Unix & Windows with DB2 for z/OS for key language constructs. This chart compares the z/OS Version 8 from March 2004 with the LUW version from October 2004.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. Sheryl Larsen provided the base for this information, but the mistakes are mine.

If you want to improve DB2 family consistency, then DB2 for z/OS Version 8 is a big step, changing the game from one of catch up to one of leapfrog.

If you want to have a book for SQL across platforms, see the 2004 Cross-Platform SQL Reference.

● Cross-Platform Development,

This chart shows the 2007 relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 9 for z/OS and DB2 9 for LUW. DB2 9 moves about half of the LUW unique items into the common set and adds a little more that is unique to the z platform. We are able to move more from the z list to the common list with Viper.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group.

The Cross-Platform SQL Reference Version 3 documents this combination, with DB2 for i5/OS V5R4.
This chart shows the 2008 relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 9 for z/OS, (DB2 9). DB2 9 moved about half of the LUW unique items into the common set and adds a little more that is unique to the z platform. DB2 9.5 for LUW, delivered in 2008. We are able to move more from the unique z list to the common list with DB2 9.5 for LUW.

There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. The changes in a specific version are not consistent. As we introduce new function, sometimes it will be on one platform first, but movement from unique lists into the common list continues to be the strongest trend.

The Cross-Platform SQL Reference Version 3.1 documents this combination, with DB2 for i V6R1.

This chart shows the 2009 relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 9.7 for LUW. DB2 9 for z/OS moved about half of the LUW unique items into the common set and adds a little more that is unique to the z platform. DB2 9.5 for LUW, delivered in 2008 and 9.7 in 2009. We are able to move more from the unique z list to the common list with DB2 9.5 and 9.7 for LUW, while bringing in some new unique function. There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. The changes in a specific version are not consistent. As we introduce new function, sometimes it will be on one platform first, but movement from unique lists into the common list continues to be the strongest trend.

The Cross-Platform SQL Reference Version 3.1 documents the prior combination, with DB2 for i V6R1.

Cross-Platform Development Version 3.1,
What You Can Do with pureXML

- Create tables with XML columns or alter table add XML columns
- Insert XML data, optionally validated against schemas
- Create indexes on XML data
- Efficiently search XML data
- Extract XML data
- Decompose XML data into relational data
- Construct XML documents from relational and XML data
- All the utilities and tools support for XML

The amount of business information in XML form is already as great or greater than other forms and growing faster - failure to leverage efficiently as structured data means high cost and/or missed opportunity. DB2 9 provides the best of both worlds, pureXML™ for native storage and integrating XML with object-relational. Performance, integrity, protection, and scale from the proven DB2 infrastructure with the flexibility of XML/XPath and relational/SQL. This overcomes the complexity & limitations of prior models (shred, CLOB, or XML only). In 2006 IBM introduced a new generation data server with the availability of DB2 9. The explosive growth of XML based data standards in all industries means competitive advantage for those businesses that use it most effectively and efficiently.

Client, policy and claims processing in Insurance; supply chain management in Retail; financial transactions and asset management in Banking; patient care in Healthcare; citizen service in Government; implementing Service Oriented Architectures (SOA) in Computing Software and Services - and many other processes across all industries - increasingly rely on information captured and exchanged in XML form. Our clients are increasingly managing XML format text documents in a content management system for proper governance and efficient use in the business process workflow. But few are realizing the full value of all the business data they possess that are in XML format.

The impact of pureXML is seen by a large banking client with a requirement to update over 500,000 XML data records per day. Attempts to use a competitors relational data server failed. Using DB2 9 with pureXML, the application was able to update more than half a million data records in less than an hour.

A large insurance client has seen the impact of pureXML to development time and cost with a 65% reduction in lines of code and more than 75% reduction in time required to develop services accessing XML data.
We need a partitioned table space with some of the segmented table space features.
The advantages of segmented space maps for partitioned table spaces:
Universal Table space (Partition By Range): A partitioned segmented table space.
Partitioning column required. One table per table space.
Universal Table space (Partition By Growth) A partitioned segmented table space.
No partitioning column required. One table per table space.

**Partition By Growth (PBG):** Single-table table space, where each partition contains a segmented page set (allows segmented to increase from 64GB to 16TB or 128 TB with 32K pages). Eliminates need to define partitioning key and assign key ranges. Partitions are added on demand. A new partition is created when a given partition reaches DSSIZE. DSSIZE defaults to 64G. Up to MAXPARTITIONS Retains benefits of Utilities and SQL parallelism optimizations for partitioned tables SEGSIZE defaults to 4 & LOCKSIZE defaults to ROW Considerations: Single-table table space Always defines as LARGE (DSSIZE) Need PBR for query partition elimination No LOAD PART, ALTER ADD PART, or ROTATE PART All indexes are NPSIs

**Automatic repositioning of Variable columns** to end of row. Length attributes replaced with Indicators positioned after fixed length columns. Any table space created or reorged in DB2 9 NFM.
To Convert: REORG or LOAD REPLACE a table space or partition, or ADD PARTITION No EDITPROCs or VALIDPROCs PIT RECOVER will set the table space to the row format of the PIT
Catalog / Directory remains in Basic Row Format (BRF)
LOB Improvements

- Progressive Streaming for LOB Locator Values
  - DB2 uses LOB size to determine whether to send LOB data to Java or DB2 CLI clients in one go (<32KB), in chunks (<1MB) or as LOB locator (>=1MB)
    - Transparent to application using LOB locators
- FETCH CONTINUE
  - Allows applications to retrieve LOB/XML data in pieces without the use of locators
- File reference variables
  - A file reference variable allows direct transfer of LOB data between DB2 and the file named in the variable
- Utility Changes
  - REORG LOB reclaim space
  - LOAD / Cross load LOB column lengths > 32KB in V7, V8 APARs
  - Logging for > 1GB LOBs
  - Online CHECK LOB and DATA
- Elimination of LOB locks for improved availability and performance

For Java and DB2 CLI programs that use locators with LOBs, Improves performance and less network traffic for LOBs that are less than 1MB
Default behavior if using DB2 9 for z/OS  Requires DB2 Connect 9.1 FP 1  No changes required to programs using locator values
DB2 Client and Type-4 driver manage progressive streaming of data to program
DB2 for z/OS determines whether to flow LOB values or Locators to client based on size thresholds for JDBC, SQLJ, and CLI
  - For small LOBs, (Default <= 32KB) the performance should approximate that of retrieving a VARCHAR column of comparable size
  - Medium size LOBs (Defaults > 32KB and <= 1MB)
  - For large LOBs (Default over 1MB) locators are still used
Specific FETCH that contains LOB or XML columns  Used with programs that materialize LOBs
Application uses a buffer that might not be large enough to hold the entire LOB or XML value.  If any of the fetched LOB or XML columns do not fit, DB2 returns information about truncated columns and the actual length.
Retrieval of LOB or XML data in multiple pieces without use of locators  Must specify WITH CONTINUE on initial FETCH  Subsequent fetches use FETCH CURRENT CONTINUE
Application must manage buffers & reassemble data  Not required to fetch entire object before moving to next  SQLCA indicates whether data is truncated
LOAD / Cross load LOB column lengths > 32KB supported:  in V7 and V8 APARs
Logging for > 1GB LOBs
REORG LOB reclaim space  SHRLLEVEL(REFERENCE) Allows LOG NO  SHRLLEVEL(NONE) is still an option.
Online CHECK LOB and DATA
Elimination of LOB locks  Now using LRSN & page latching for consistency checks  Prior to DB2 9, LOB locks were held until commit  Even for UR
Space Search for LOB allocation  No LOB locks acquired for space search  Read LSN granularity improved to page level in LOB table space
Improved availability & performance  Particularly for UR readers
Requirements::  NFM “Locking protocol 3” GBP changes, APAR.  Automatic in non-data sharing
Indexing Enhancements

- Larger index pages allow for more efficient use of storage
  - Fewer page splits for long keys
  - More key values per page
- Index compression provides page-level compression
  - Data is compressed to 4K pages on disk
  - 32K/16K/8K pages results in up to 8x/4x/2x disk savings
  - No compression dictionaries
    - Compression on the fly
    - No LOAD or REORG required
- Rebuild Index SHRLEVEL CHANGE
- Define RANDOM index keys to avoid hot spots with multiple processes inserting sequential keys

Indexing improvements contribute to the overall improvements in query performance. Specific improvements include index compression, index on expression, index key randomization, and larger index page sizes.

Larger index pages allow for more efficient use of storage, with fewer page splits for long keys and more key values per page. Multiple processes inserting sequential keys can create hot spots on indexes. Randomized index keys avoid hot spots. Application insert throughput improved via avoidance of locking conflicts, but retrieval of sequential rows is likely to be slower.

Bigger index page: 4K, 8K, 16K, or 32K page → Up to 8 times less index split

  Good for heavy inserts to reduce index splits. Especially recommended if high latch class 6 contention in data sharing. Two forced log writes per split in data sharing
  Or high latch class 254 contention in non data sharing shown in IFCID 57
Many improvements help data sharing efficiency and usability in DB2 9. Logging was improved substantially for data sharing customers. Several enhancements help with faster restart, releasing retained locks faster and allowing data sets to be opened sooner.

- ACCESS DB…., watch for a PTF to allow ranges and wild cards in the –ACCESS command

**Mode (OPEN)** Forces the physical opening of the page set or partition on just the local member. This moves the overhead of the physical open from an SQL thread to the command thread. This improves the transaction rate for the first SQL thread to reference a given page set or partition.

**Mode (NGBPDEP)** Converts the specified page set or partition, non-disruptively to a non-group buffer pool dependent. You should use this before running large batch processes against a particular page set or partition to improve performance in a data sharing environment. Only issue this command to the member on which you plan to run the batch programs.
Utilities Highlights

- More online utilities
  - Rebuild Index SHRLEVEL CHANGE
  - Reorg LOB now supports SHRLEVEL REFERENCE (space reclamation)
  - Check data, LOB and repair locate … SHRLEVEL CHANGE
  - Check index SHRLEVEL REFERENCE supports parallel for > 1 index
  - Clones for “online LOAD REPLACE”
- Online REORG BUILD2 phase elimination
- Substantial CPU reductions
- REORG parallelism for UNLOAD, RELOAD, LOG phases
- Utility TEMPLATE switching
- UNLOAD SKIP LOCKED DATA option

If you look at all the utility offerings, we have been adding more SHRLEVEL CHANGE and REFERENCE utilities to improve availability in every release. Now with DB2 9, the following utilities have SHRLEVEL CHANGE (RUNSTATS, COPY, REORG TABLESPACE, REORG INDEX, LOAD RESUME, REBUILD, UNLOAD, CHECK INDEX, CHECK DATA, CHECK LOB, and REPAIR). Cloned tables effectually function as LOAD REPLACE SHRLEVEL CHANGE. We have also improved availability for REORG TABLESPACE of a part/part range by removing the BUILD2 phase.

From a performance perspective, REORG of partitioned table spaces now has partition parallelism much like already existed in REBUILD INDEX, CHECK INDEX, COPY, RECOVER, and LOAD.

Utility template switching allows flexibility in which template applies to data sets based on attributes of the object processed.

UNLOAD SKIP LOCKED DATA gives another option besides just UR processing or CS processing. UR will not get locks, and therefore can unload “dirty” data. CS acquires locks and can therefore be slowed with lock contention. SKIP LOCKED is a CS option that will skip those rows/pages that are locked and give consistent data (but not all rows) without being slowed much by contention.
The COPY and RECOVER utilities each had several enhancements in DB2 9. One of the most important is RECOVER to point in time with consistency. We see that customers can’t quiesce their workloads to provide a clean point for recovery, so RECOVER now recovers to a point on the log, then rolls back the in-flight work. The quiesce is not needed.

Prior to DB2 9 the user specified the deletion criteria either as before a specific date or by greater than a given age in days. DB2 9 has an alternative, by which instead of deletion criteria, retention criteria can be specified.

There are a number of improvements and extensions to the volume base utilities, or the BACKUP/RESTORE SYSTEM utilities.

If you need to do a conditional restart, DSNJU003 now allow you to specify a timestamp instead of an RBA/LRSN. The option is also available for the SYSPITRT for easier prep to truncate the log before running RESTORE SYSTEM.
Why Migrate to DB2 9 for z/OS?

- **Business needs**
  - Reduce CPU time & disk space
  - Improve business agility
  - Service Oriented Architecture
- **Application developers need**
  - PureXML for a powerful SQL and XML interface to XML data
  - Powerful new SQL enhancements
  - Portability with SQL and data definition compatibility
- **Database Administrators need**
  - Improve availability and performance
  - More flexible security and easier regulatory compliance
  - Better web application & data warehouse function and performance
  - LOB function, performance, usability

DB2 9 has a lot for everyone, unlocking the potential of V8. Here are just a few of the highlights. The business needs include CPU cycle reductions that deliver in most utilities, improved query optimization, improved business agility via faster implementation cycles, and new pureXML™ that builds a strong foundation for SOA and XML initiatives. Kevin Campbell, an Application Architect at Univar USA said it better than I can, “This is not a bolt-on or band-aid approach, DB2 9 for z/OS is XML without compromise.”

Database Administrators (DBAs) need improved database availability and performance including LOBs, reorganization, backup and recovery, and partitioning enhancements. DBAs also get more flexible trusted network context and role-based security to help with regulatory compliance. A wide range of enhancements improve ERP application and data warehouse functionality and performance. Large object (LOB) function is added with file reference variables and REORG, while performance is improved.

Application developers are most excited by PureXML, which adds a powerful SQL and XML interface to access XML data stored in a native format. Application developers need powerful new SQL enhancements including MERGE and TRUNCATE statements, INTERSECT and EXCEPT set operations, and spatial support for geographical data. Text handling is improved with the XML changes, many new built-in functions, and an upcoming text server. Improved SQL and data definition compatibility with other DB2 platforms makes porting much easier.
Why is migration easier to DB2 9 for z/OS?

- Migration process enhancements: ENFM speed, CM*
- Much less performance regression:
  - Earlier improvements
  - Package stability & tools for avoiding access path issues
- CCSIDs and old product issues resolved in V8
- Simpler virtual storage considerations
- Less impact from incompatible changes
- Earlier deliveries from vendors

DB2 9 migration has been easier for many customers and tends to be faster than DB2 V8. Many customers note that the process and the quality are solid.

Migration process enhancements
ENFM is shorter (10 minutes versus an hour), as only two table spaces need to be reorganized in this phase (versus 18 in V8 including the largest ones).
Drop back to CM* if needed (no single steps without a drop back).

Much less performance regression
More consumable performance improvements
Utility CPU improvements in CM
Bind stability to reduce concern about access path regression.
CCSIDs and old product issues resolved in V8 migration, not a problem.
Simpler virtual storage considerations – incremental improvements.
Less impact from incompatible changes
Many vendors are ready today.
Here are some highlights for items that deliver the most quickly and easily:

Very little to no action is required for the utility CPU reductions, logging improvements, improved index page split, larger prefetch, write & preformat quantities, some LOB performance, DDF virtual storage constraint relief. The first group delivers in CM.

The next items require some work. Changed online REORG and other utility improvements require process changes and use of SHRLEVEL(CHANGE). Improved RUNSTATS statistics needs some analysis to determine where the value is greater than the cost of gathering the new statistics.

Optimization improvements are automatic for dynamic SQL, but require work to REBIND for static SQL. In both cases, we need baselines to check for regression. REOPT(AUTO) for dynamic SQL needs analysis to be sure the improvement is working. EDMPOOL virtual storage constraint relief also requires a REBIND.

Optimization Service Center takes some learning, but should be fast for those who have used Visual Explain in the past. See the new redbook, SG24-7421, DB2 9 for z/OS: New Tools for Query Optimization.

LOB lock avoidance requires a quiesce of all subsystems in NFM until APAR PK62027. Reordered row format requires a REORG in NFM and varying length columns, and use with small columns can reduce compression. See APARs.

Index improvements for larger page sizes, compression, index on expression require database design work to determine where they are applicable. ALTERs, REORGs and creation of new indexes are needed.
Many of the strongest trends in data warehousing and business intelligence address the key strengths of System z, with high quality of service, availability, resilience, scalability, security and the ability to share data and workloads.

Rather than being used as separate systems, business intelligence delivers more value by being integrated into the business processing, so that the business intelligence runs with more current information and the rest of the business can use the capabilities and data.

Integration and use of a wider range of data sources fits with the System z. Rather than needing to extract and duplicate the information, moving to another platform, processing power and disk can be shared. If data needs to be copied, the process is much simpler.
DB2 was designed to handle transactions and queries. In Version 1, the CPU times for transactions were roughly double those of IMS for similar transactions, so DB2 was positioned for the information center. Each release of DB2 has included substantial improvements for some types of queries: optimization improvements, SQL language, indexes, and avilit to manage the work. So DB2 has increased its ability to handle large, complex queries on large volumes of data over the years.

Customers have been using DB2 for queries over the time, and many do have warehouses on DB2 for z/OS. In most cases, the workload is shared with other work, rather than dedicated or standalone.

Current trends in business intelligence match the strengths of System z and DB2 for z/OS. Cost improvements with specialty engines and improvements in the latest versions make DB2 for z/OS even better for warehousing.
IBM announced a beta program for InfoSphere Warehouse on System z which will provide a highly scalable, lower cost way to design, populate and optimize a DB2 for z/OS data warehouse to support BI applications such as Cognos 8 BI. This new offering will simplify operational complexity with a single database for both operational and warehouse data - - reducing costs related to data duplication while providing more efficient access to DB2 data. Significantly improve query performance for users who want to drill down into specific data stored in DB2 for z/OS. (For example, a sales manager looking to build sales plans can look not only at revenue by year, but also data by quarter, month, and customer to make fast business decisions.) Give customers the ability to support near real-time decisions based on core business data managed in DB2 for z/OS, helping customers gain additional competitive advantage and value from their operational data. InfoSphere Warehouse on System z will further strengthen the IBM Data Warehousing and Business Intelligence software solution for System z that today includes: DB2 for z/OS
•InfoSphere Information Server for System z
•InfoSphere MDM Server for System z
•Cognos 8 BI for System z
DB2 delivers many tools in the Developer Workbench. Now those tools and many more come with Data Studio, delivering with DB2 and Informix database management systems. The versions which come with Data Studio are generally the latest tools, with many improvements.

IBM Data Studio is for analysts, architects, designers, developers, and administrators who work across the entire data life cycle and across the full set of IBM relational DBMS: DB2 for LUW, DB2 for z/OS, DB2 for i and IDS. Rational provides support for the application life cycle, while the mission for Data Studio is the data life cycle. Almost every customer needs the Data Studio tools included with DB2, and many need the additional ones in the Data Studio portfolio.

DB2 is changing to deliver interfaces for this tooling, usually new stored procedures. See the IBM Data Studio web page for all the changes in IBM Data Studio. Watch closely, as this area is changing fast.

http://ibm.com/software/data/studio
DB2 for z/OS V7 became generally available (GA) March 2001, and V8 delivered three years later. DB2 9 became generally available in March 2007, three more years. We expect the next version will be roughly 3 years from DB2 9 GA to DB2 X or whatever the name becomes.

The themes for future versions will continue to focus on core platform strengths of performance, scalability, reliability, stability, availability, resilience, and security. PureXML and Schema evolution or data definition on demand will be ongoing for a long time. In contrast, most of the 64 bit evolution should be completed in DB2 X.

The key interfaces for customers and vendors expand for both XML and for SQL. Information is a key leg of the SOA platform, and DB2 for z/OS provides many advantages for data management in SOA.

Standards, interoperability, portability and security along with secure access using the latest technologies are key touch points. Productivity improvements for application developers and for database administrators are very important as data grows in scale and complexity.
Reducing CPU from DB2 9 to DB2 X without significant administration or application changes is the primary thrust of the performance work. Most of the changes work with CPU caching and path lengths, so that applications are not changed. We can take advantage of new instructions without needing to have other techniques for older processors which do not have fast implementations of the new instructions. This work is preliminary, but the performance plan for DB2 X is much more aggressive than in any recent version. The last version which contained significant improvements for reducing CPU time in transactions and batch was Version 2 in 1988. Versions 3 to 9 made improvements in queries and in utility CPU time and provided many scalability improvements, but little reduction in transaction CPU time, other than in specific situations.

As customers move from DB2 V8 to DB2 9 CM, they generally find some CPU improvements, often in the utilities. As customers move to DB2 X CM, we anticipate a bigger reduction coming from transactions and batch work. REBIND will improve optimization. The largest improvements are expected for applications that can use the database changes, such as a hash for primary key access, and SQL improvements in DB2 X.

We expect DB2 X to run only on z10, z9, z890, z990, and later processors, and to provide CPU reductions from the beginning, with improvements in CM, but more dramatic reductions for applications that can take advantage of the improvements in application design.
Providing significant scalability and performance improvements is an important DB2 X objective. Synergy with the latest System z10 processors and follow-on machines provides part of the improvements. Being able to deliver high scalability for increasing numbers of processors is important for growth and costs. Being able to use large real memory effectively is required for scalability. Working with the hardware to improve CPU time by using new instructions and improving memory access and cache access is growing more important.

Synergy with z/OS 1.10 and later helps with managing larger volumes, and can help with memory, such as using 1 MB pages to manage the large amounts of memory.

The results are expected to be improved transaction times, with lower CPU usage for both large and small DB2 subsystems on transaction and batch workloads.

See the next pages for detailed changes.
Performance

- Hash access path
- Parallel index update at insert
- Faster single row retrievals
- Inline LOBs
- LOB streaming between DDF and rest of DB2
  - Faster fetch and insert, reduced virtual storage
- DEFINE NO for LOBs (and XML)
- MEMBER CLUSTER for UTS
- Efficient dynamic SQL statement caching with literals

Hashing is an faster alternative to index access for a key = value lookup, not needing to touch the index pages.

Indexes can be updated in parallel to speed the process when many indexes must be maintained.

A single row select is another common process that is speeded up in this release.

Inline LOBs resolve the common performance problem of having large objects where most LOBs are small, by keeping part of the LOB on the page with the rest of the data. Streaming for LOBs improves distributed processing. Define NO lets DB2 avoid data set definitions.

Member clustering is now allowed for data sharing situations with frequent updating for universal table spaces.

Even if the dynamic SQL has literal values, rather than host variables or parameter markers, DB2 can use the dynamic statement cache effectively.
Performance

- Index include columns
- Workfile spanned records, PBG support, and in-memory enhancements
- Buffer pool enhancements
  - Utilize z10 1MB page size
  - “Fully in memory” option
- Internal performance optimizations
  - Improved cpu cache performance
  - Use new hardware instructions
  - Streamline performance-critical paths
- Solid state disk

Being able to have a column that is unique, but includes extra columns can improve performance substantially in some situations, and makes the DB2 family more consistent.

If you want to have a table be completely in memory, then the new option can make the process easier. System z10 and z/OS 1.10 allow a 1 megabyte page size, instead of a 4 kilobyte page size, improving efficiency for large amounts of memory.

Many of the performance do not require changes in applications or administration. The CPU cache performance and new hardware instructions can reduce CPU time without customer action, other than moving to this version.

Solid state disk is working today with DB2, and integration can improve the performance and value. Solid state disk is expected to improve substantially in size and value in the next few years.
A wide range of optimization improvements will help with DB2 X query performance. New techniques for access data more quickly are the first step. The ability to have a unique index with additional columns delivers the ability to use indexes more effectively. Where the full key is equal to a value, a hash can reduce CPU and memory access substantially.

Optimization techniques include better ability to avoid bad access paths where an average cost is a little better, but some access will be much worse. Improved optimizations can reduce the query processing by a substantial amount.

Increasing parallel processing by lifting restrictions and improving efficiency will return results faster and enable more use of zIIP.
Virtual storage is most common constraint for large customers. Virtual storage can limit the number of concurrent threads for a single member or subsystem.

The DB2 9 virtual storage objective was 10-15% relief. The DB2 X target is 80% to 90% of the DBM1 address space. We expect the result to be the ability to run much more concurrent work, with an early guess of 3 to 5 times more threads.

Storage monitoring should be drastically reduced. Customers are consolidating LPARs. Sometimes they need to have more than one DB2 subsystem on an LPAR, costing real storage and CPU. With these changes, work can run in one DB2 subsystem, rather than needing more members.

The net for this change is expected to be reduced cost, improved productivity, easier management, and the ability to grow DB2 use much more easily.

Increasing the number of concurrent threads will expose the next tier of constraints. DB2 X will address a number of the next items, such as utility locking, catalog concurrency.
Customers are constrained by virtual memory to various degrees. This slide shows a relatively extreme situation experienced by some customers today. With a maximum of 500 threads (very dependent upon workload) in a DB2 subsystem, this customer is using two DB2 subsystems in the same data sharing group on a single LPAR. This is not efficient for memory of CPU, but avoids the memory constraints with fewer LPARs. Additional relief for virtual storage comes with IMS V11 and other products.

This example allows customers to run 10 times as many threads in a single DB2 subsystem, improving efficiency for storage and CPU. The biggest change is easier management and simpler growth. Most customers use data sharing for high availability, and that need still exists. Extreme scale continues to need data sharing, but fewer data sharing members can mean easier management and reduced resource consumption.
Other System Scaling Improvements

● Other bottlenecks can emerge in extremely heavy workloads
  ‣ several improvements planned to reduce latching and other system serialization contention
  ‣ new option to for readers to avoid waiting for updaters
  ‣ eliminate UTSERIAL lock contention for utilities
  ‣ Use 64-bit common storage to avoid ECSA constraints

● Concurrent DDL/BIND/Prepare processes can contend with one another
  ‣ restructure parts of DB2 catalog to avoid the contention

● SPT01 64GB limit can be a constraint, especially if “package stability” support is enabled
  ‣ relieve 64GB limit for SPT01

Increasing the number of concurrent threads will expose the next tier of constraints. DB2 X will address a number of the next items, such as utility locking and catalog concurrency.

The UTSERIAL lock means that scheduling 20 concurrent REORGs for hundreds of partitions in each one will result in deadlocks too often. Reducing the granularity by removing this lock means that the jobs run.

Improving the catalog structure to allow row level locking can improve concurrency substantially.

The DB2 catalog structure is changed to move most of the large fields with repeating rows of data into LOB columns, eliminating the 64 GB limit and making the information more readable by separating character from binary data. The LOB columns are inline for improved performance.
These are the changes in table space type in diagram form.

What is not done? Change from multi-table segmented table space. Change back to classic simple, segmented and partitioned. The strategic choice for table space type is the universal table space. Simple table spaces are deprecated, and this version provides a migration path.

If you need more improvements in table spaces, then universal table spaces – either partition by range or partition by growth should be your choice.
Continuous availability requirements continue to escalate. Large batch and maintenance windows are in the past. Those windows are being closed on the fingers of DBAs. DBAs increasingly need the ability to make all changes and to do all maintenance activities online or around the clock.

DB2 X allows more online schema changes with an ALTER for a PENDING change, then an online REORG to take effect. ALTER a simple or segmented table space containing a single table or a partitioned table space to a universal table space. Page size and member clustering can be altered. Index changes become less disruptive. Pending changes which have not been completed with a REORG can be dropped.

REORG is improved to allow SHRLEVEL(CHANGE) for LOBs. Consistent image copies can be provided without a quiesce. Inline copies to allow for dataset-level FlashCopy.

Online REORG usability and performance enhancements are provided.
Customers are being pressed for a wide range of improved security and compliance. Data retention is a growing need. Protecting sensitive data from the privileged users and administrators is required. Separation of authority for security, access, and some common tasks, like EXPLAIN will help. Auditing for privileged users can also make compliance simpler.

In DB2 X, we expect to have a form or temporal data or the ability for a table to contain both current and historical data, and to query the information as of a specific point in time.

Access control is refined in several ways with better granularity for the administrative privileges and with finer grained access control at the row and column level, including the ability to mask access to some fields. Auditing is also enhanced.
Some of the improvements come with Data Studio for application programming and administration – stronger cross-platform graphical interfaces, better integration with Java, improvements in the ability to develop and debug.

Some of the improvements come within DB2 for z/OS. Improvements in SQL and XML improve productivity for those who develop new applications and for those who are porting from other platforms. Some of the improvements remove complexity from application tasks.

DB2 has a strong focus on making DB2 easier to use by automating tasks and eliminating tasks where possible. Avoiding the manual invocations can also help avoid problems for running the function too often or not often enough. Where the task cannot be eliminated, the frequency and monitoring can be reduced, such as the need to reorganize. The improvements for virtual storage and for availability also help DBA productivity.
Continuous availability requirements continue to escalate. Large batch and maintenance windows are in the past. Those windows are being closed on the fingers of DBAs. DBAs increasingly need the ability to make all changes and to do all maintenance activities online or around the clock.

DB2 X allows more online schema changes with an ALTER for a PENDING change, then an online REORG to take effect. ALTER a simple or segmented table space containing a single table or a partitioned table space to a universal table space. Page size and member clustering can be altered. Index changes become less disruptive. Pending changes which have not been completed with a REORG can be dropped.

REORG is improved to allow SHRLEVEL(CHANGE) for LOBs. Consistent image copies can be provided without a quiesce. Inline copies to allow for dataset-level FlashCopy.

Online REORG usability and performance enhancements are provided.
The pureXML improvements are noted two slides later. Improvements for LOBs include the ability to placed inline in the data page when they are small, and better ability to unload and load to a sequential file, rather than needing a separate file for each LOB.

Instead of needing to explicitly cast each data type, more flexible data typing improves productivity.

Time stamps have the option to include a time zone and to have more precision (nanoseconds, rather than microseconds).
Versioned data or Temporal Data

- Table-level specification to control the management of data based upon time
- Two notions of time:
  - System time: notes the occurrence of a data base change
    - “row xyz was deleted at 10:05 pm”
    - Query at current or any prior period of time
    - Useful for auditing, compliance
  - Business time: notes the occurrence of a business event
    - “customer xyz’s service contract was modified on March 23”
    - Query at current or any prior/future period of time
    - Useful for tracking of business events over time, app logic greatly simplified
- New syntax in FROM clause to specify a time criteria for selecting historical data

User queries request historical data from base table.
SQL procedure language is now allowed in scalar user-defined functions. Applications can use data above the bar in the new ODBC structures.

Optimistic locking can be managed better in dynamic SQL. Dynamic statement caching has new techniques to use access path hints and to use the dynamic statement cache when literals are used.

Some applications require to access part of a result set based on a certain position. SQL pagination delivers an efficient way either to skip unused part of result set or to retrieve the interesting part of result set only. Numeric-based pagination allows applications to access part of a result set based on an absolute position. Data-dependent pagination lets applications access part of a result set based on a logical key value.
This chart shows the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS. This step in the process is DB2 X for z/OS. DB2 X for z/OS moves more of the LUW unique items into the common set and adds a little more that is unique to the z platform. DB2 9.5 for LUW, delivered in 2008 and 9.7 in 2009. We are able to move more from the unique z list to the common list with DB2 9.5 and 9.7 for LUW, while bringing in some new unique function. There are three sets of SQL noted above, with some that is unique to DB2 for z/OS in the first group, SQL that is common across DB2 for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 for Linux, Unix and Windows in the bottom group. The changes in a specific version are not consistent. As we introduce new function, sometimes it will be on one platform first, but movement from unique lists into the common list continues to be the strongest trend. The Cross-Platform SQL Reference Version 3.1 documents the prior combination, with DB2 for i V6R1.

A range of XML improvements delivers a strong release 2 of the pureXML function. Customers use of DB2 9 pureXML shaped this delivery of improved performance and usability.

Multi-versioning: During the execution of a SQL statement, a row with an XML column can be kept in a work file. The row in the work file does not contain the actual XML document. Instead, the information needed for DB2 to retrieve the XML document from the XML table is cached in the work file. The problem occurs if the XML document in the XML table is deleted or updated. When the row in the work file is fetched, DB2 cannot find the expected XML document in the XML table, and the SQL statement fails with an error SQLCODE.

XML UPDATE: Applications which require parts of XML documents to be modified need to break apart the XML document into modifiable pieces, make the modification to a piece, and then construct the pieces back into an XML document.

SP/UDF/Trigger support: XML variables inside SQL PL, XML arguments, transition variables.
DB2 warehousing capabilities continue to be enhanced at a rapid pace. Improvements in the SQL and XML were noted earlier. Improvements in optimization and additional parallel processing allow faster query processing.
Key details about DB2 X

- CM, ENFM, NFM modes
- Prerequisites
  - z/OS V1.10
  - DB2 9 for z/OS in NFM
  - System z10, z9, z890, z990, and above (no z800, z900)
- Items deprecated in earlier versions eliminated:
  - Private protocol → DRDA (new help in DSNTP2DP)
  - Old plans and packages V5 or before → REBIND
  - Plans containing DBRMs → packages PK62876
  - ACQUIRE(ALLOCATE) → ACQUIRE(USE)
  - Old Plan table formats → DB2 V8 or 9 format (59 columns)
  - XML Extender → XML type
  - DB2 MQ XML user-defined functions and stored procedures → XML functions
  - DB2 Management Clients feature (DB2 Administration Server, Control Center, & Development Center) → IBM Data Studio application & administration services
  - msys for Setup DB2 Customization Center → install panels
  - BookManager use for DB2 publications → Info Center, pdf

Here are a couple of thoughts about what might be required in hardware and software to run DB2 X. Much will depend upon the timing of the deliveries and market acceptance. Moving forward as quickly as possible means that some of the past must be left behind. See the list of deprecated functions from prior versions.

The above features are still included in DB2 9 and may be dropped from future versions. Note the direction indicated to the right of the arrows, as these are the functions provided to replace the existing function. If you are using any of these functions, you are advised to move to the new function.

See the Installation Guide section, “Functions that are deprecated” and the announcement material for more information on these changes.

### DB2 X for z/OS At a Glance

<table>
<thead>
<tr>
<th>Application Enablement</th>
<th>RAS, Performance, Scalability, Security</th>
<th>Simplification, Reduced TCO</th>
<th>Dynamic Warehousing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Versioned data or Temporal</td>
<td>• Wide range of performance improvements</td>
<td>• 5 – 10 times more threads per DB2 image</td>
<td>• Moving sum, moving average</td>
</tr>
<tr>
<td>• pureXML enhancements</td>
<td>• Hash access to data</td>
<td>• Auto statistics</td>
<td>• Many query optimization improvements</td>
</tr>
<tr>
<td>• Last Committed reads</td>
<td>• More online schema changes</td>
<td>• Data compression on the fly</td>
<td>• Query parallelism improvements</td>
</tr>
<tr>
<td>• SQL improvements that simplify porting</td>
<td>• Catalog restructure for improved concurrency</td>
<td>• Query stability enhancements</td>
<td>• Advanced query acceleration</td>
</tr>
</tbody>
</table>

DB2 X builds upon many themes: core platform strengths of performance, scalability, reliability, stability, availability, resilience, and security. All of this work helps with total cost. PureXML and Schema evolution or data definition on demand enhancements are included. Most of the 64 bit work should be completed in DB2 X. XML, SQL, web services and other programming interfaces extend for usability. Information is a key leg of the SOA platform, and DB2 for z/OS provides many advantages for data management in SOA. Standards, interoperability, portability and security along with secure access using the latest technologies are key touch points. Productivity improvements for application developers and for database administrators are very important as data grows in scale and complexity. Warehousing continues to evolve, with key trends matching DB2 for z/OS strengths.
You can get most of the books from the Information Center or from the DB2 Library web page. The books continue to be updated, so get the latest ones. Some of the IBM Redbooks publications will be helpful. You may need books from the z/OS Library as well.

http://publib.boulder.ibm.com/infocenter/imzic/

http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011656


Be sure to use the latest information to save time and problems. Some of the IBM Redbooks publications have added lately or updated (next page).
DB2 9 in IBM Redbooks Publications

31. Watch for titles on DB2 serialization & concurrency, utilities
Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

• CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
• ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us. Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in the operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/400, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, System z,
The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, ODBC
Thanks for coming.