Introduction

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

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

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

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

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

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

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

1. Improved performance and lower costs

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

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

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

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

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

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

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

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

2. Ten times more concurrent users

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

3. Reduce contention in database administration

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

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

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

4. More administrative capabilities while the database is online

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

5. Improved security and auditing with more granularity

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

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

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

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

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

7. SQL enhancements to improve portability

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

Several other enhancements improve application portability, such as:

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

8. pureXML performance and usability enhancements

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

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

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

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

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

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

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

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

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

10. Faster OLTP performance – Hash Access

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

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

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

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

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

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

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