DB2 11 for z/OS: Technical Overview

by John Campbell and Gareth Jones

This IBM® DB2® for z/OS® white paper provides a high-level overview of the changes introduced in DB2 11 for z/OS, including the following topics:

- DB2 11 performance expectations and improvements
- Availability and resilience enhancements
- Data sharing improvements
- Security enhancements
- Utility enhancements
- Analytics improvements
- New and enhanced application features
- Easier version upgrade, including the new Application Compatibility feature

The performance expectations section concentrates on improvements that can be expected with and without REBIND and describes which improvements require DBA or application programmer effort. We discuss other features in more detail in subsequent sections.[1,2]

---

[1] Information regarding potential future products is intended to outline general product direction, and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products might not be incorporated into any contract. The development, release, and timing of any future features or functionality described for IBM products remains at IBM's sole discretion.

[2] This document contains performance information based on measurements done in a controlled environment. The actual throughput or performance that any user experiences will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user can achieve throughput or performance improvements equivalent to the numbers stated here.

DB2 11 Performance Expectations

IBM recognizes that performance improvements can also result in cost savings for customers, making the IBM System z® platform more attractive and helping customer investment to deliver value.

DB2 10 for z/OS provided some significant application performance improvements by reducing CPU consumption for many online transaction processing (OLTP) applications running simple SQL queries. This theme continues in DB2 11 for z/OS, with the focus now on complex queries. In this paper, "simple queries" are queries that retrieve data using a primary key lookup; for the purposes of this discussion, you can regard all other queries as complex queries. While some performance regression for a small number of queries is possible, most DB2 11 customers can expect to see reduced CPU consumption for a significant proportion of their complex queries.

The performance improvements customers can expect from DB2 11 might vary depending on many factors. Changes to the DB2 Optimizer mean the access path chosen by the DB2 11 REBIND process could differ from that chosen by DB2 10. For example, DB2 11 can choose a nested loop join instead of a sort merge join, or vice versa. The read/write ratio is also important because DB2 11 reduces the logging overhead for write-intensive applications.

Other factors affecting performance expectations include the number of rows returned, the number and type of columns returned, the number of partitions touched, and the number of partitions defined. You are more likely to see performance improvements when using table-controlled partitioning and data-partitioned secondary indexes (DPSIs) because IBM has worked to make DPSI much more useful in this release. The BIND option RELEASE(COMMIT/DEALLOCATE) and the use of table compression are two other important factors influencing the kind of performance improvement you can expect.

Customers often want to know what enhancements they can anticipate from any new release of DB2, so it is important to be clear about which enhancements are immediately available. First, there are no Data Definition Language (DDL) changes, no Data Manipulation Language (DML) changes, and no application changes. However, this does mean that achieving the most significant performance gains for static SQL packages require a REBIND. Additional performance savings require user action in the form of DDL or DML changes or other DB2 changes.

Performance Expectations for OLTP and Batch

Table 1.1 shows the OLTP and batch CPU savings reported from IBM's own internal benchmarks. In these benchmarks, System z measures total CPU consumption—that is, the CPU consumption reported in the statistics trace as well as the accounting trace. This point is important because when you deploy DB2 11 and measure your own performance improvements, you'll need to make sure you look at the complete picture by including CPU consumption figures from both statistics and accounting.

Table 1.1: Sample CPU savings for OLTP and batch	
Workload description	**CPU savings (%)**
Batch running locally with a varied workload	10
Batch in a distributed environment with concurrent sequential insert processes	5
OLTP in a distributed environment executing simple SQL queries	4
OLTP running locally in a data sharing environment using the basic RBA format executing simple SQL queries	1
OLTP running locally in a data sharing environment using the extended RBA format executing simple SQL queries	4.7
Distributed OLTP running SAP banking in a data sharing environment	9
Distributed OLTP running complex queries and using SQL PL stored procedures	9.8
A set of IBM DB2 utilities	3

These figures demonstrate that positive CPU savings apply across a broad range of SQL query workloads. The modest 1 percent improvement seen for the local, non-distributed workload running simple SQL queries in a data sharing environment is expected, given that in this release IBM focused on reducing CPU consumption for complex queries. However, a healthy reduction in CPU consumption is reported for the other workloads, including the IBM utility set.

Performance Expectations for Queries

The figures in Table 1.2, representing a variety of industry-standard benchmarks and customer workloads executing complex queries, are certainly impressive. Note that these workloads include not only business intelligence but also complex OLTP and batch.

Table 1.2: DB2 11 Query workloads after REBIND without APREUSE	
Workload description	**CPU savings (%)**
TPC-H benchmark queries	37
TPC-H–like queries	21
Query customer workload 1	36
Query customer workload 2	36
Query customer workload 3	12
Query customer workload 4	46
Benchmark – SAP BW	33
Benchmark business intelligence, long running	47
Benchmark business intelligence, short running	14

Some of these workloads use static SQL, and for purposes of the test, the containing DB2 plans were rebound without APREUSE under DB2 11, opening up new and improved access path choices for these applications. Of course, those choices are automatically available to dynamic SQL at PREPARE time. Although most performance improvements are available even after a successful REBIND with APREUSE(ERROR) or APREUSE(WARN), you must rebind without APREUSE to get the new or improved access paths.

It is also important to understand that these are the sorts of workloads that are expected to benefit significantly from DB2 11. The savings you see might differ. Underlying the variation in CPU savings for these workloads is the fact that the functional usage of SQL varies from workload to workload; the savings you can expect will depend on the characteristics of the SQL requests issued by your applications and the design of your database schema.

DB2 11 Performance Expectations Summary

In summarizing the kind of CPU savings expected with DB2 11, you will notice we use the phrase "Up to"—which includes the value zero. For example, "Up to 10 percent for complex OLTP" should be understood as "From zero to 10 percent for complex OLTP." We discussed the reasons for this convention earlier, but, essentially, the savings you can expect are very workload-dependent.

To summarize, the total CPU savings you can expect for your SQL applications in DB2 11 are as follows:

- Up to 10 percent for complex OLTP
- Up to 10 percent for update-intensive batch
- Up to 25 percent for reporting queries without compressed tables
- Up to 40 percent for complex queries with compressed tables

Performance Highlights

In this section, we highlight some of the most significant performance improvements in DB2 11 for z/OS. Some of these might require user action, while others might not.

Two of the performance improvements can particularly help write-intensive batch applications—that is, applications that use INSERT, UPDATE, and/or DELETE intensively.

The first improvement in this area does not require REBIND because it is related to the log output buffer, which has been moved from the MSTR address space to the 64-bit common area. The advantage of this change is that DB2 now avoids a cross-memory call to the MSTR address space to update the log buffer. Avoiding these cross-memory calls reduces CPU time in particular for write-intensive applications.

The second improvement can benefit data sharing users with certain kinds of extremely write-intensive applications. It requires you to be in new-function mode (NFM). It reduces the need to spin on the CPU to obtain a new log record sequence number (LRSN). A lot of work was done in DB2 10 to reduce LRSN spins for INSERT processing, but further enhancements are available in DB2 11 if you take the necessary user actions to exploit the extended log relative byte address (RBA)/LRSN once in NFM. DB2 11 provides additional LRSN spin avoidance for UPDATE and DELETE processing and continued improvement for INSERT by greatly reducing the need for LRSN spins when updating space map pages.

Several enhancements benefit query workloads. Improving query workload performance was a primary focal point for SQL performance enhancements in DB2 11.

Applications that access compressed tables, especially those where the selected columns are clustered together or where predicates to be applied are clustered together, and where many rows are scanned, should see reduced CPU consumption because of an improved decompression algorithm.

There is also assistance for sort-intensive queries. DB2 11 reduces the need to use physical work files, generates custom machine code to use in sort processing, and improves the processing of in-memory work files. These changes not only reduce CPU consumption but also result in fewer I/O requests.

DB2 11 improves performance for queries accessing multiple DPSI partitions in a join operation where there are additional join predicates on the columns making up the partitioning key. It does this by using page range screening to ensure that only the necessary DPSI parts are accessed during the join operation.

DB2 11 includes a number of enhancements to reduce the number of data moves required and the amount of code executed when returning rows from the IBM DB2 Analytics Accelerator (IDAA). These changes are targeted mainly at queries accessing IDAA to retrieve large result sets.

Before DB2 11, queries bound with RELEASE(COMMIT), which accessed one or a small number of partitions, were sensitive to the number of partitions defined rather than the number of partitions accessed. For queries accessing tables with a large number of partitions—say more than 200—the CPU cost starts to become significant, meaning performance scales poorly as the defined number of partitions grows. However, in DB2 11, DB2 is only sensitive to the number of partitions accessed in a single COMMIT scope. The result is that the larger the number of defined partitions, the greater the performance improvement.

DB2 11 continues the theme of large real memory exploitation, delivering a further enhancement for customers using large buffer pools when running on an IBM zEnterprise® zEC12 CEC. The zEC12 provides support for page-fixed 2 GB page frames, helping to improve throughput and reduce CPU consumption when you have very large buffer pools (i.e., larger than 2 GB).

Readers might be familiar with DB2's use of runtime optimizations, or customized procedures for operations frequently used by SELECT, UPDATE, and so on, called xPROCs. DB2 11 introduces a new customized procedure for column processing that can reduce CPU consumption for queries that select a very large number of columns. The greater the number of columns selected, the greater the reduction in CPU consumption.

A significant number of DRDA, or Distributed Data Facility (DDF), applications are often described as "chatty." These long-running DDF transactions issue multiple simple SQL statements, causing a lot of send/receive TCP/IP processing in the DIST address space. Before DB2 11, DB2 used a technique called asynchronous receive in the DIST address space, which required extra supervisor request block (SRB) dispatching. With z/OS 2.1 Communications Server, or z/OS 1.13 Communications Server with APAR PM80004 applied, DB2 11 DDF replaces all asynchronous calls with synchronous calls to eliminate the SRB dispatching overhead. This change results in reduced network latency and a significant CPU reduction in the DIST address space for chatty DDF applications. No rebind is required to benefit from this change.

The enhancements we've discussed so far provide cost savings through reduced CPU consumption. This next enhancement is a simple cost saving benefit. In DB2 10, the prefetch engines and the deferred write engines became zIIP-eligible. DB2 11 extends this support by allowing all other system agents, with the exception of the P-lock negotiation agent, to become zIIP-eligible.

Two cases in particular benefit from this enhancement. First, in data sharing, DBM1 chargeable CPU time can be reduced because castout processing is eligible for zIIP offload. Second, MSTR address space chargeable CPU can also be reduced for update-intensive workloads because log read and log write are also zIIP-eligible.

ESP Customer Feedback

Several Early Support Program (ESP) customers compared the performance of DB2 11 with that of DB2 10 and sent SMF data to the DB2 Lab at IBM's Silicon Valley Lab for analysis. The measurements made using the provided data consistently indicated that performance improved once these customers had migrated to DB2 11 and rebound their static SQL packages. This fact is important because the DB2 Lab cannot possibly reproduce all customer workloads as workloads vary so widely. Due to the difficulty driving comparable online workloads, most of the customer workloads were batch. This is another significant point because, for the findings to be useful, the workloads must be comparable in terms of the SQL profile and the data being processed. It is worth noting that some customers were also able to send in SMF data for OLTP workloads, but the variety of these workloads was smaller.