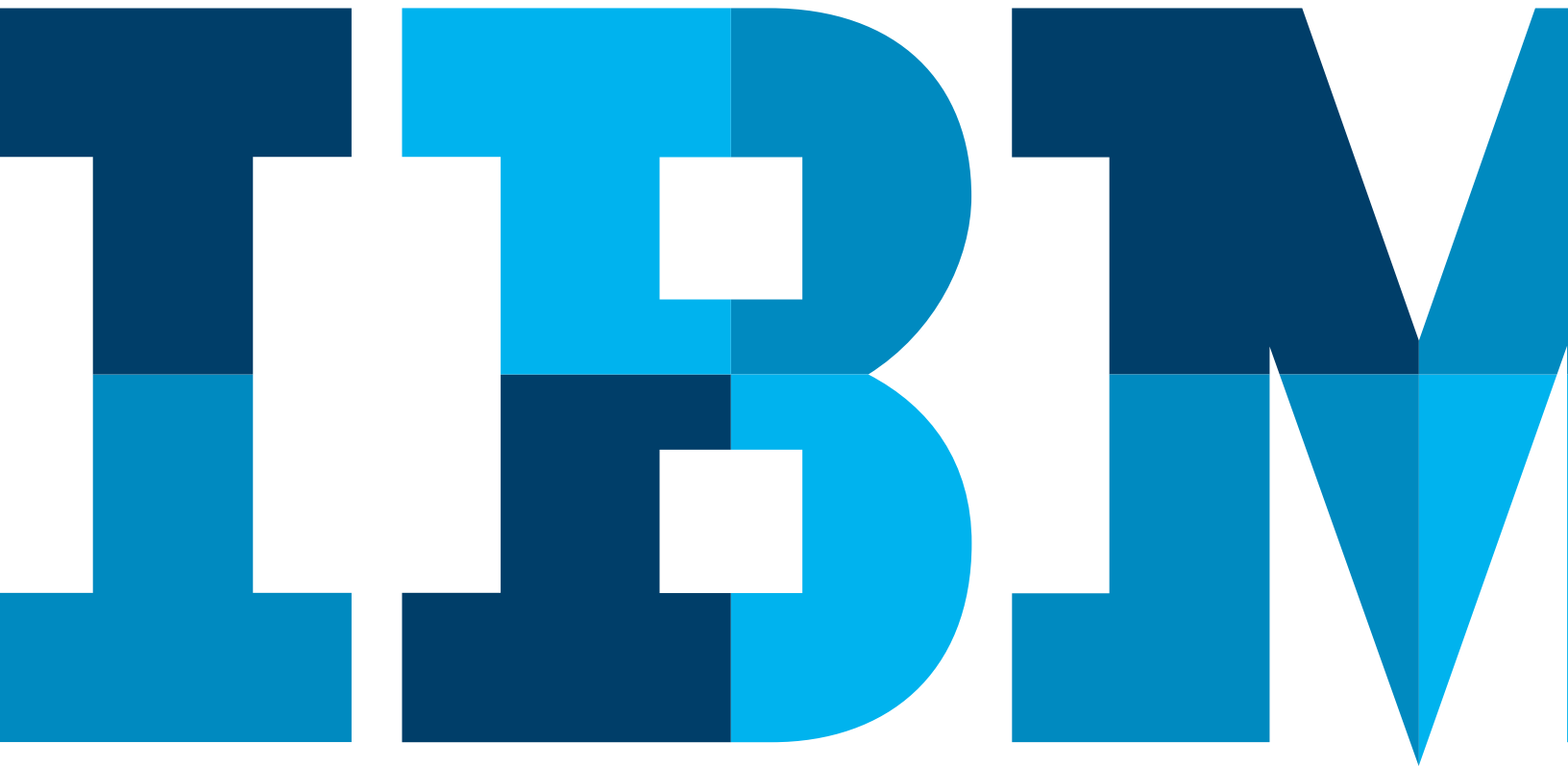


Migrating to IBM DB2 10 for z/OS—Proactive rebind strategies and tooling



Contents

2 Migration steps

- 3 Step 1: Establish key performance indicators (KPIs) and performance baseline.
 - Why rebind?
- 6 Step 2: Manage packages that will have automatic rebinds.
 - Using DB2 Path Checker in version-to-version migrations
 - Finding packages without plan table entries
- 8 Step 3: Identify packages that will have improved access path performance.
- 8 Step 4: Use triage techniques to manage problematic access paths.
- 10 Step 5: Handle remaining access path performance problems with SQL tuning tools.
 - IBM DB2 Administration Tool for z/OS
 - IBM DB2 SQL Performance Analyzer for z/OS
 - IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS
 - Environment capture
- 17 Step 6: Reexamine KPIs after migration.

17 Conclusion

17 For more information

18 Appendix A

19 Appendix B

There are many reasons companies worldwide are anxious to move to IBM® DB2® 10 as soon as possible. IBM clients that participated in the DB2 10 beta program and others that have already migrated to DB2 10 since it became available in 2010 have reported that they've achieved—and in many cases exceeded—the CPU-usage reductions promised. Some CPU-usage reductions will happen as soon as DB2 10 is brought up in conversion mode, but to achieve the maximum CPU-usage reduction requires some effort on your part.

Although achieving the maximum CPU benefit of DB2 10 is the goal, a necessary step to getting there—and the primary focus of this paper—is rebinding. You may be reluctant to rebind for one or more reasons. Perhaps you're concerned by the tedious and time-consuming nature of the task. More likely you're worried that Structured Query Language (SQL) performance will not be as good after a rebind; this concern will be addressed later in the paper.

Migration steps

There are six recommended migration steps to complete the rebind process within your DB2 10 migration. Several IBM products that you may already own can assist you with these tasks. The aim of this paper is to highlight the value provided by the IBM tools, while offering value to clients that don't currently have the IBM products.

1. Establish key performance indicators (KPIs) and performance baseline.
2. Manage packages that will have automatic rebinds.
3. Identify packages that will have improved access path performance.
4. Use triage techniques to manage problematic access paths.
5. Handle remaining access path performance problems with SQL tuning tools.
6. Reexamine KPIs after migration.

Step 1: Establish key performance indicators (KPIs) and performance baseline.

The use of a monitoring tool, whether you use IBM Tivoli® OMEGAMON® or a competitive product, is extremely important as part of your migration. The monitor can be used to help you identify the return on investment of moving to DB2 10. As we said at the beginning, we expect to see a nice reduction in CPU usage. The monitoring tool can also help you find overlooked performance opportunities, especially by reviewing new metrics that are available in DB2 10. And, most important, the monitor can capture data that will be needed in the event you experience any performance regression.

When planning your migration to DB2 10, be aware that you must use version 5.1.0 or higher of Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS. The older versions of OMEGAMON will not work with DB2 10. Version 5.1.0 or higher of Tivoli OMEGAMON supports DB2 8, DB2 9 and DB2 10. IBM recommends that you implement V5.1.0 before you start your DB2 migration.

Before beginning the migration to DB2 10, you need to establish a performance baseline that you can compare against as you move through the different modes of the migration. You need to identify the critical business applications and capture KPIs, such as CPU usage and elapsed time. The long-established IBM DB2 Performance Monitor software, with its batch reporting capabilities, are now part of the Tivoli OMEGAMON XE for DB2 software, reports from which will be examined later in the paper.

An invaluable function of Tivoli OMEGAMON XE for DB2 in a version-to-version migration is the ability to create a performance database. IBM provides the data definition language (DDL) to create the database and sample load statements that can be used to populate the tables. The load input data is generated out of the standard batch reporting process. Most clients taking advantage of this feature populate their performance database tables daily as part of their process of running accounting and statistics reports. This process is not just limited to System Management Facility (SMF) data. It can bring in data from other DB2 trace sources, in addition to the Tivoli OMEGAMON near-term history data sets. Figure 1 illustrates the topology of a typical performance database. Take a look at the batch reporting user guide if you're interested in learning more about using this capability.

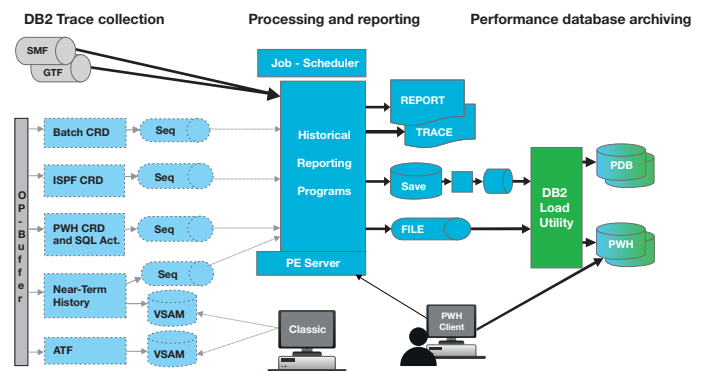


Figure 1: Analyzing application performance—performance database

As mentioned earlier, you need metrics from before and after the migration, especially if you need to open a problem management record (PMR) and engage IBM DB2 support in solving a performance regression. DB2 developers use Tivoli OMEGAMON batch reporting extensively, as they're developing each new release of DB2 and they're familiar with using the reports to solve problems. So, you should be sure to have basic statistics and accounting data available, either in the performance database or by retaining your DB2 SMF 100 and 101 records.

Example reports follow. Figure 2 shows a sample statistics long report, which gives you an idea of how well the DB2 subsystem is performing. One area you should examine is storage utilization in the DBM1 address space. You should see quite a bit of relief, especially if your existing subsystems are constrained by virtual storage.

```

LOCATION: DSNDAIB          OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (VSRI)          PAGE: 1-1
GROUP: N/P              STATISTICS REPORT - LONG                          REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P             TO: NOT SPECIFIED
SUBSYSTEM: DAIB        SCOPE: MEMBER
DB2 VERSION: V10
                                INTERVAL FROM: 07/06/11 03:39:00.00
                                INTERVAL TO: 07/06/11 04:10:00.00
                                TOTAL THREADS : 0.00
                                TOTAL COMMITS : 1192.8K
                                DATA SHARING MEMBER: N/A
----- HIGHLIGHTS -----
INTERVAL START : 07/06/11 03:39:00.00   SAMPLING START: 07/06/11 03:39:00.00   TOTAL THREADS : 0.00
INTERVAL END   : 07/06/11 04:10:00.00   SAMPLING END  : 07/06/11 04:10:00.00   TOTAL COMMITS : 1192.8K
INTERVAL ELAPSED: 31:00.000490          OUTAGE ELAPSED: 0.000000                DATA SHARING MEMBER: N/A
-----
SQL DML          QUANTITY /SECOND /THREAD /COMMIT   SQL DCL          QUANTITY /SECOND /THREAD /COMMIT
-----
SELECT           0.00      0.00    N/C      0.00    LOCK TABLE      0.00      0.00    N/C      0.00
INSERT          3537.7K   1901.97  N/C      2.97    GRANT             0.00      0.00    N/C      0.00
NUMBER OF ROWS  3537.7K   1901.99  N/C      2.97    REVOKE           0.00      0.00    N/C      0.00
UPDATE         4528.5K   2434.69  N/C      3.80    SET HOST VARIABLE 0.00      0.00    N/C      0.00
NUMBER OF ROWS  4528.5K   2434.69  N/C      3.80    SET CURRENT SOLID 0.00      0.00    N/C      0.00
MERGE           0.00      0.00    N/C      0.00    SET CURRENT DEGREE 0.00      0.00    N/C      0.00
DELETE         238.4K    128.17   N/C      0.20    SET CURRENT RULES 0.00      0.00    N/C      0.00
NUMBER OF ROWS  238.4K    128.17   N/C      0.20    SET CURRENT PATH  0.00      0.00    N/C      0.00
PREPARE        9337.8K   5020.32  N/C      7.83    SET CURRENT PRECISION 0.00      0.00    N/C      0.00
DESCRIBE       9337.8K   5020.32  N/C      7.83    CONNECT TYPE 1    0.00      0.00    N/C      0.00
DESCRIBE TABLE 0.00      0.00    N/C      0.00    CONNECT TYPE 2    0.00      0.00    N/C      0.00
OPEN           9918.7K   5332.64  N/C      8.32    RELEASE           0.00      0.00    N/C      0.00
CLOSE         3883.9K   2088.12  N/C      3.26    SET CONNECTION    0.00      0.00    N/C      0.00
FETCH         3882.7K   2087.49  N/C      3.26
NUMBER OF ROWS  15660.0K  8419.33  N/C     13.13  ASSOCIATE LOCATORS 0.00      0.00    N/C      0.00
TOTAL DML      44665.5K  24.0K    N/C     37.45  ALLOCATE CURSOR   0.00      0.00    N/C      0.00
TOTAL          0.00      0.00    N/C      0.00  HOLD LOCATOR      0.00      0.00    N/C      0.00
FREE LOCATOR    0.00      0.00    N/C      0.00
TOTAL          0.00      0.00    N/C      0.00

STORED PROCEDURES QUANTITY /SECOND /THREAD /COMMIT   TRIGGERS QUANTITY /SECOND /THREAD /COMMIT
-----
CALL STATEMENT EXECUTED 0.00 0.00 N/C 0.00 STATEMENT TRIGGER ACTIVATED 0.00 0.00 N/C 0.00
PROCEDURE ABENDED 0.00 0.00 N/C 0.00 ROW TRIGGER ACTIVATED 0.00 0.00 N/C 0.00
CALL STATEMENT TIMED OUT 0.00 0.00 N/C 0.00 SQL ERROR OCCURRED 0.00 0.00 N/C 0.00
CALL STATEMENT REJECTED 0.00 0.00 N/C 0.00

USER DEFINED FUNCTIONS QUANTITY /SECOND /THREAD /COMMIT   ROW ID QUANTITY /SECOND /THREAD /COMMIT
-----
EXECUTED 0.00 0.00 N/C 0.00 DIRECT ACCESS 0.00 0.00 N/C 0.00
ABENDED 0.00 0.00 N/C 0.00 INDEX USED 0.00 0.00 N/C 0.00
TIMED OUT 0.00 0.00 N/C 0.00 TABLE SPACE SCAN USED 0.00 0.00 N/C 0.00
REJECTED 0.00 0.00 N/C 0.00

USE CURRENTLY COMMITTED QUANTITY /SECOND /THREAD /COMMIT
-----
INSERT ROWS SKIPPED 0.00 0.00 N/C 0.00
DELETE ROWS ACCESSED 0.00 0.00 N/C 0.00
UPDATE ROWS ACCESSED 0.00 0.00 N/C 0.00
    
```

Figure 2: A sample statistics long report from Tivoli OMEGAMON XE for DB2

Figure 3 shows an accounting long report, which also includes a tremendous amount of data at the application level. You'll have the detailed breakout of times by class 1, class 2 and class 3, and time spent in stored procedures and triggers. You'll also see how much CPU usage is being off-loaded to IBM System z® Integrated Information Processor (zIIP) engines.

```

ACCOUNTING
REPORT
LAYOUT (LONG)
ORDER (PRIMAUTH-PLANNAME)
SCOPE (MEMBER)
...

LOCATION: OMPD61          OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (VSRI)          PAGE: 1-1
GROUP: N/P              ACCOUNTING REPORT - LONG                          REQUESTED FROM: NOT SPECIFIED
MEMBER: N/P             TO: NOT SPECIFIED
SUBSYSTEM: DMS1        ORDER: PRIMAUTH-PLANNAME                          INTERVAL FROM: 07/28/10 08:45:00.15
DB2 VERSION: V10      SCOPE: MEMBER
                                INTERVAL TO: 07/28/10 08:45:00.15
PRIMAUTH: SKA PLANNAME: MULTIPKG

ELAPSED TIME DISTRIBUTION          CLASS 2 TIME DISTRIBUTION
-----
APPL  =====> 62%                CPU  =====> 7%
DB2   ==>>> 6%                     SECPU =====> 84%
SUSP  =====> 32%                NOTACC =====> 0%
                                SUSP  =====> 0%

AVERAGE  APPL(CL.1) DB2 (CL.2) IFI (CL.5)  CLASS 3 SUSPENSIONS  AVERAGE TIME  AV. EVENT  HIGHLIGHTS
-----
ELAPSED TIME 3:23.32193 1:18.09838  N/P LOCK/LATCH(DB2+IRLM) 0.155724 164.00 #OCCURRENCES : 4
NONNESTED 1:29.35995 1:15.25326  N/A IRLM LOCK+LATCH 0.076568 2.25 #ALLIEDS : 4
STORED PROC 1:02.49925 2:17.107  N/A DB2 LATCH 0.079156 161.75 #ALLIEDS DISTRIB: 0
UDF 51.982729 0.039014  N/A SYNCHRON- I/O 0.624019 3859.75 #OBATS : 0
TRIGGER 0.000000 0.000000  N/A DATABASE I/O 9.550911 3836.75 #OBATS DISTRIB.: 0
CP CPU TIME 5.769037 5.236138  N/P LOG WRITE I/O 0.073988 23.00 #NO PROGRAM DATA: N/P
AGENT 5.769037 5.236138  N/P OTHER READ I/O 20.382125 6804.25 #NORMAL TERMINAT: 0
NONNESTED 3.361427 2.937803  N/P OTHER WRITE I/O 29.592807 6323.50 #ABNORMAL ROLLUP: 1
STORED PROC 2.394543 2.979111  N/P SER. TASK SWITCH 5.192599 589.00 #ABNORMAL TERMIN: 0
UDF 0.013967 0.000425  N/A UPDATE COMMIT 2.381118 571.25 #CP% PARALLEL: 0
TRIGGER 0.000000 0.000000  N/A OPEN/CLOSE 0.085983 0.75 #IO PARALLELISM: 0
PAR.TASKS 0.000000 0.000000  N/A SYSLOGNG REC 0.019129 8.75 #INCREMENT. BIND: 4
                                EXT/DEL/DEF 2.696186 6.25 #COMMIT% : 2397
                                OTHER SERVCEP 0.010183 2.00 #ROW FRACKS : 49
    
```

Figure 3: A sample accounting long report from Tivoli OMEGAMON XE for DB2

Besides the traditional batch reporting functions, there is a significant new feature in Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS V5.1.0. If you're only licensed for Tivoli OMEGAMON XE for DB2 Performance Monitor, then you would need to upgrade to Performance Expert to use this feature, known as Extended Insight. In a nutshell, it enables you to begin monitoring distributed workloads coming into DB2 on System z at the application server. IBM supports applications written using Java, CLI or .Net. Extended Insight captures performance metrics from the call to DB2 on System z and back, and stores that data in its own repository. That repository can be another great source for before-and-after performance comparisons.

Figure 4 provides a couple of screen shots of Tivoli OMEGAMON XE for DB2 Performance Expert Extended Insight on z/OS. Through this web-based interface, you will open multiple browser windows to do the comparisons. The software has a history function which enables you to step back to a previous time and choose how large a time slice you want to see.

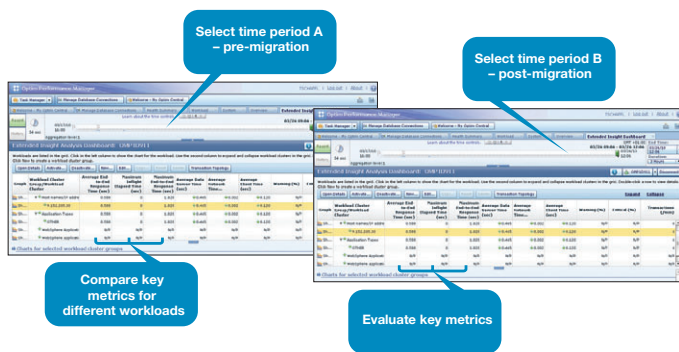


Figure 4: Two screen shots of Tivoli OMEGAMON XE for DB2 Performance Expert Extended Insight on z/OS

Why rebind?

The first question that usually arises is, “Why do I need to do a rebind when I move to a new version of DB2?” In actuality, the real question should be, “How can I perform the necessary rebinds in a controlled manner, and do so with confidence that performance will be equal or better after the rebind?” But let’s focus on the issues associated with rebind, and what to do when performance declines as a result of a rebind.

DB2 version 8 was the first release of DB2 for z/OS to use 64-bit addressing. Operating in 64-bit mode provides greater in-memory data storage and improved performance. Database Request Modules (DBRMs) that were bound before version 8 require a rebind to enable them to operate in 64-bit mode.

If you have packages that include SPROCS (select procedures), they will be disabled when you come up in conversion mode. If you’re not familiar with SPROCS, they provide a significant performance improvement for the internal handling of columns by DB2. You can expect to see degraded performance on packages with SPROCS until you re-enable them through a rebind.

An important point is that many performance enhancements introduced in DB2 10, such as improvements in virtual storage usage and parallelism, require a rebind to become activated. Finally, when you run a package that was bound in a previous version of DB2, that package must first be internally converted into the structure of a DB2 10 package. This is adding unnecessary overhead to the execution of your packages and represents CPU cycles that can be bought back through a rebind.

In order to eliminate some of the overhead just discussed—that is the overhead that occurs when executing very old packages—each new DB2 version now automatically rebinds packages that were bound prior to a certain version of DB2. DB2 10 will automatically rebind any package that has not been rebound since version 5. Because this rebind will occur automatically, you will not have an opportunity to try to proactively address any potential performance regression with the SQL in the packages. As a result, you’ll want to identify and analyze these packages before migrating to DB2 10.

Step 2: Manage packages that will have automatic rebinds.

Let's take a moment and talk about bind and rebind strategies. For some of the reasons mentioned above, many shops are reluctant to ever rebind, mostly out of fear that performance will not be as good. As just discussed, a migration to a new version of DB2 will require revisiting the idea of no rebinds.

Another common issue is that many shops do not bind their packages with "EXPLAIN YES" specified as an option on their bind or rebind commands. Without capturing the EXPLAIN data at the time of a bind or rebind, we have no way of seeing the details on the access path that the DB2 optimizer has chosen, and we have no way to compare that access path to the new path that would be chosen by DB2 in the event of a rebind. So, it is strongly recommended that you make it a practice to specify "EXPLAIN YES" on all binds and rebinds, at least in your production environment.

One of the first things you need to do in the migration to DB2 10 is run the pre-migration health-check job. This job will be placed in the DB2 sample data set by one of the program temporary fixes (PTFs) that you will be applying to your current DB2 subsystems to prepare for migration to DB2 10. Two of the reports produced by this job will identify for you all the plans and packages that will be automatically rebound by DB2. Make a note of the most recent timestamp in these reports for easy analysis of these plans and packages in IBM DB2 Path Checker for DB2 for z/OS.

Using DB2 Path Checker in version-to-version migrations

To deal with missing EXPLAIN data, a couple of queries can be used to generate DB2 Path Checker commands that will generate rebind cards—for packages where the access path

will remain the same, and for packages that have a lower cost estimate indicated by the optimizer. Then, the process of finding packages that have SQL that may present some challenges can begin.

Figure 5 shows the input sources used—and the outputs produced—by DB2 Path Checker. You can see on the left that the DB2 catalog and directory, your DBRMs and your plan tables are used as inputs. On the right, several user-selectable reports are produced. These range from a comprehensive EXPLAIN report that will appeal to those of you who are used to dealing with plan table data directly, to a report some prefer to use for quick access path comparisons (to be discussed later in this paper).

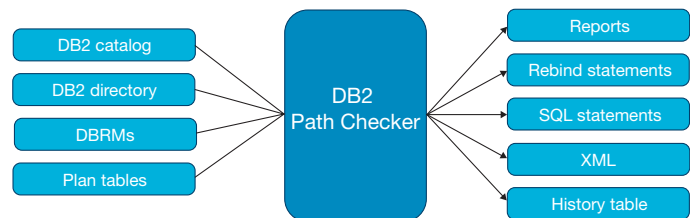


Figure 5: DB2 Path Checker inputs and outputs

Rebind statements need to be generated to two different files: one file for packages where there is no change in access path or where the optimizer indicates better performance, and another file for packages where the access path changes or the cost estimate is higher. Output files that may be used to interface with the EXPLAIN tools are also produced. One of these contains SQL statements in a plain text file; the other has the SQL in Extensible Markup Language (XML) format. You'll see later in this paper how to bring these files into the tools.

Finding packages without plan table entries

The first step is to find packages that do not currently have entries in your plan table. Figure 6 shows the use of a simple query to accomplish this task and to generate DB2 Path Checker EXPLAIN commands. Running these generated commands will yield entries in both the plan table and DSN_STATEMENT table. There is no guarantee that a new EXPLAIN will match the access path currently in use, so some packages may simply lack sufficient data to enable you to be proactive.

```
SELECT DISTINCT 'EXPLAIN DBRM '
  CONCAT RTRIM(NAME) CONCAT ' TO PACKAGE '
  CONCAT RTRIM(COLLID) CONCAT '.* IN P390I.PLAN_TABLE '
FROM SYSIBM.SYSPACKAGE A
  WHERE CREATOR = 'P390H' AND VALID = 'Y'
  AND OPERATIVE = 'Y'
  AND NOT EXISTS (SELECT 1 FROM P390H.PLAN_TABLE B
    WHERE A.COLLID = B.COLLID
    AND A.NAME = B.PROGNAME
    AND A.VERSION = B.VERSION );
```

Figure 6: DB2 Path Checker—first analysis

It's possible that some SQL statements may require some rewrite in order to maintain or improve performance, so hopefully you'll have application areas that are willing to invest time in helping you ensure the best possible DB2 performance.

The first pass using DB2 Path Checker will be to identify those packages that will keep their current access path when rebinding. Since the access path is not changing, the variation in the optimizer's cost estimate can be ignored. These are the safest rebinds you can do.

A sample of the query used to generate the DB2 Path Checker commands, and some sample test commands showing the simple syntax used by DB2 Path Checker, will be shown later.

First, let's look at a sample access path analysis report. Figure 7 depicts the report mentioned earlier that is very easy to read and provides just the minimum information that you need.

```
PCK218I PROGRAM TEST01      HAS A CHANGED ACCESS PATH
      QUERYNO 327
      OLD PROCSU 209
      NEW PROCSU 139
      COLLECTION TESTCOLL
      NEW VERSION 2005 - 04 - 11 - 18.34.53.754785
      OLD VERSION 2005 - 04 - 11 - 18.20.42.913225

PCK217I TOTAL DBRMS COMPARED 1
PCK217I TOTAL STATEMENTS COMPARED 2
PCK217I TOTAL ACCESS PATH CHANGES 1
PCK217I TOTAL ACCESS PATH SAME 1
PCK217I TOTAL NEW SQL STATEMENTS 0
PCK217I TOTAL DELETED SQL STATEMENTS 0
PCK217I TOTAL DSN_STATEMENT COST INCR 0
```

Figure 7: A sample DB2 Path Checker access path analysis report (SYSCHG)

In the report shown in Figure 7, two SQL statements were analyzed and only one of them has a changed access path. You'd need to go to one of the other reports to see the SQL statement and the old and new access path and plan table data, but from this report you can see that SQL statement 327 has a changed access path. With this new access path, the optimizer is indicating a significant decrease in its cost estimate. This package, in the indicated collection, is a safe candidate for rebind. One thing to note about this report is that the data seen here can also be produced in DB2 load format so you can place the data in a history table. Then you have a baseline to compare to for future rebinds.

Rebinding these safe packages will take care of anywhere from 60 - 80 percent of all packages in most shops. Perhaps these version-to-version rebinds aren't as tedious or time consuming as some people think.

Step 3: Identify packages that will have improved access path performance.

The next step is to look for packages which show an improvement in the optimizer's cost estimate. This query (see Figure 8) builds the DB2 Path Checker test commands that are needed, in this case based on collection ID and based on the timestamp, so you're looking for packages that have not been bound in some time. This example is just slightly different from what was used in the first analysis (see Figure 6).

```
-- SELECT BIND DATE OLDER THAN ...
-- CHANGE CPUPCT nn ITERATIVELY
SELECT DISTINCT 'TEST DBRM '
CONCAT RTRIM(NAME) CONCAT ' AS PACKAGE '
CONCAT RTRIM(COLLID) CONCAT '.*'
CONCAT ' IN TRAIN01.PLAN_TABLE '
FROM SYSIBM.SYSPACKAGE A
WHERE OWNER = 'TRAIN01'
AND COLLID = 'TST01COLLX'
AND TIMESTAMP < '2009 - 09- 04- 14.24.34.593683';
```

Figure 8: DB2 Path Checker—second analysis

Figure 9 shows an input stream with several DB2 Path Checker commands. There is an options statement that will be modified to make iterative passes through this process. Specifying a value greater than zero (0) allows the creation of rebind statements for packages where the increase in CPU is within some specified percentage. For example, if the value specified is five (5), then DB2 Path Checker would generate rebinds for a package in which the increase in CPU is five percent or less.

```
OPTIONS CPUPCT 00 CATALOGSQL
CONNECT TO GT8G
SET CURRENT SQLID = 'PUBLIC'
-- TEST COMMANDS ARE FROM 2ND ANALYSIS QUERY
TEST DBRM TEST01 AS PACKAGE TST01COLLX.* IN
TRAIN01.PLAN_TABLE
TEST DBRM TEST02 AS PACKAGE TST01COLLX.* IN
TRAIN01.PLAN_TABLE
```

Figure 9: An example of a DB2 Path Checker input stream

Running with CPU percent (CPUPCT) of zero on the OPTIONS statement will give us rebind cards for packages that have a lower cost. It's typical to have rebound an additional 5 - 10 percent of your packages once you've handled all of the ones that have a lower cost estimate. So you're now down to a manageable number of packages that might represent challenges during a rebind.

Step 4: Use triage techniques to manage problematic access paths.

Next, begin an iterative process to find packages that have SQL that may suffer mild to severe performance degradation. You can increase the test value for the CPU percentage as you deem appropriate. As you find the problem packages and SQL, you'll use a set of triage techniques.

The SYSCHG report (see Figure 7) is the fastest report to scan through, focusing on access path and cost changes. Note that if you change the CPU—for example, from IBM z10™ to IBM zEnterprise™—simultaneous with your DB2 migration, it's best to use the MSU value provided by the optimizer.

Following are five steps, or guidelines, to follow in determining why performance has degraded.

1. Check for changes in inner and outer joins. This is probably due to a difference in the statistics that were available to the optimizer at the time of the original and the present bind. The statistics advisor in the IBM InfoSphere® Optim™ Query Workload Tuner for DB2 for z/OS product can really help you quickly fix your statistics problems.
2. Look at the before and after values for the number of matching index columns. Again, a likely reason for this change is due to the difference in your statistics. Also note that this is not influenced by hints, so if you're using hints to try to coerce the optimizer in reselecting a certain access path, in this case they will be ignored.
3. In each new version of DB2, IBM strives to increase the performance of SQL; so, comparing the costs between a current bind and a very old bind may show what appear to be very large increases in the cost estimates. Take time to examine these and be sensitive to the amount of data in the accessed tables today versus what may have been there five, six or seven years ago. (This may not be as big an issue as it first appears to be.)
4. At this point the easy steps are completed, giving you confidence that moving to the new DB2 is safe and performance is good. Hopefully, you're seeing those CPU reductions mentioned at the beginning of the paper. Now you have to tackle the remaining packages and SQL statements that don't want to cooperate.

5. There are a lot of coding tricks, often seen presented at DB2 conferences and user group meetings, that can be used to try to trick the optimizer. These are great as temporary solutions, but should not be counted on to always work. As the optimizer evolves, some of these may no longer work for you. If you find these, you need to go back to your developers and ask them to review their SQL to determine whether they are coding using the best practices of the newest DB2 version.

If the cost of a SQL statement has increased dramatically, by 50 percent or more, then something in the environment has most likely changed. Always start with the RUNSTATS utility to be sure the optimizer has up-to-date and accurate information for it to choose an access path. Then, two key areas to look at are the configuration of the buffer pools and ensuring that some inadvertent change in the indexes being used has not occurred.

As you migrate your production environment, you'll likely find that most of your challenges are related to new access paths. You would want to first check the order of joins to see if that is causing the performance decrease. You may need to do a column-by-column comparison of the old and new plan table entries. In most cases we expect to see access path changes result in improved performance. But one cannot overstate the importance of having up-to-date statistics before binding.

Step 5: Handle remaining access path performance problems with SQL tuning tools.

Now let’s look at tools that include EXPLAIN functionality, and discuss ways to improve SQL performance.

IBM DB2 Administration Tool for z/OS

Primarily a DBA productivity tool and not really a SQL tuning tool, the main focus of the IBM DB2 Administration Tool for z/OS is catalog browsing and generating complex work lists to carry out migrations and alterations. Let’s look at some of the included EXPLAIN functions.

Figure 10 shows Option E from the main menu. This is sometimes overlooked by users, because it’s not within the catalog navigation panels. Take note of the ability to use the tool to create a new set of tables used by the DB2 EXPLAIN function. You can use the listed commands to upgrade the columns in the listed tables to the structure that includes all of the added DB2 10 columns, and you can perform a simple EXPLAIN from this screen and list the contents of your plan table.

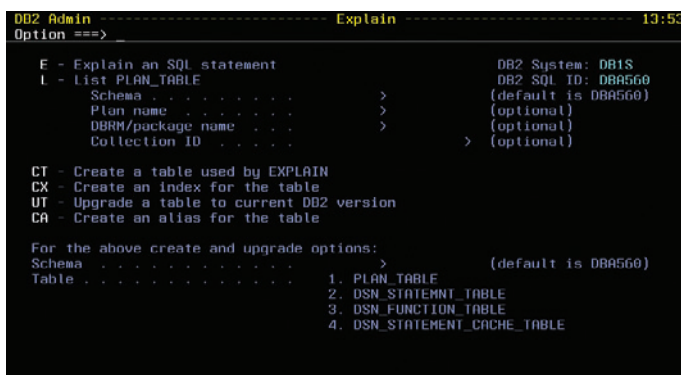


Figure 10: DB2 Administration Tool for z/OS—Explain panel (“Option E” from main menu)

Figure 11 shows a view of the raw plan table data. You can also access this view when looking at a list of packages, and you would only see the entries for the selected package. The first line command listed in the upper left is the interpret command, which will show the full plan table data for a selected entry and the basic access path data is also shown.

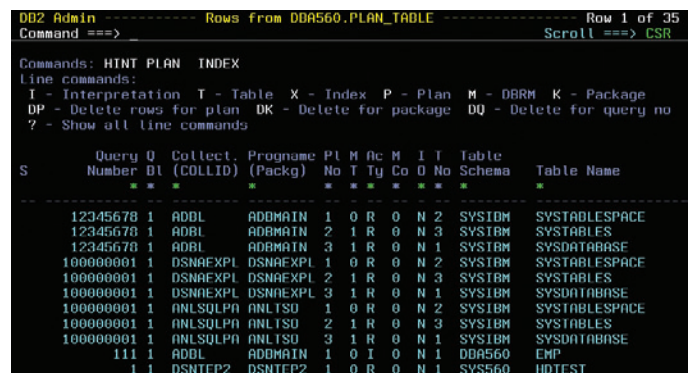


Figure 11: Plan table selection list—Option L

Basic access path information for the table can be obtained, but the DB2 Administration Tool does not offer advice on improving SQL. It is possible to invoke the next tool, while using the DB2 Administration Tool to view the SQL within a package.

IBM DB2 SQL Performance Analyzer for z/OS

The IBM DB2 SQL Performance Analyzer for z/OS is a traditional batch and Interactive System Productivity Facility (ISPF)-based tool that accepts many forms of input, such as DBRMs, source programs and text files. There are several other tools in the DB2 portfolio that provide an interface to SQL Performance Analyzer. One of these is DB2 Path Checker.

Figure 12 shows the input options. While doing an analysis with DB2 Path Checker, you would probably use option 2 or option 4. To use option 2, you would get the collection and package name from the DB2 Path Checker reports. Or, you can use the SQL text file created by DB2 Path Checker with option 4.

```

SQLPA410 ----- Basic Processing Options ----- 13:51
Option ==>

Options:                                DB2 system: DSNA
                                           DB2 SQL ID : DBA560

 1 Process plans from the DB2 system catalog
 2 Process packages from the DB2 system catalog
 3 Process application DBRM
 4 Process SQL from a sequential data set or PDS member
 5 Process SQL from program source
 6 Process a query number from the plan table
 7 Process a QMF statement
 8 Process tables from the DB2 system catalog

```

Figure 12: DB2 SQL Performance Analyzer for z/OS—Basic Processing Options panel

Figure 13 shows a panel that enables you to explain the statements or a subset of the statements in a currently bound package. Again, the collection and package name and query numbers are in the DB2 Path Checker reports.

```

SQLPA410 ----- Process Packages ----- 13:57
Command ==>                                Scroll ==> CSR

Commands: EXPLAIN SQL TABLES              DB2 system: DSNA
                                           DB2 SQL ID : DBA560

Enter the package information:
*Collection ID . . . ANLSQLPA >
*Package name . . . ANLS* >
Version ID . . . >
Current degree . . . 1 (1 or Any) >
Generations . . . 1 >
Force . . . N (Y or N) >
Statements accessing the following table:
Table owner . . . >
Table name . . . >
Range of statements:
Beginning QUERYNO . . . >
Ending QUERYNO . . . >
Location
Remote location . . . >
*Compare the old package to the new package? N (Y or N)
(An * indicates a required field.)

```

Figure 13: DB2 SQL Performance Analyzer for z/OS—Process Packages panel

The panel in Figure 14 simply enables you to bring in a sequential file or partitioned data set member that contains nothing but SQL statements. This particular file has 4 statements in it, so choose the EXPLAIN command at the top of the panel and continue.

```

SQLPA410 ----- Process SQL ----- 13:57
Command ==>                                Scroll ==> CSR

Commands: EXPLAIN SQL TABLES EDIT         DB2 system: DSNA
                                           DB2 SQL ID : DBA560

Enter the input data set name:
*Data set name . . . 'DBA560.DEMO.SQL(SULDEMO)'

Enter optional parameters:
Table qualifier . . . >
Synonym creator . . . >
Current degree . . . 1 (1 or Any)
(An * indicates a required field.)

```

Figure 14: DB2 SQL Performance Analyzer for z/OS—Process SQL panel

The first thing you see in Figure 15 is a summary of the SQL statements that were in the file. For each statement you are shown the estimated CPU usage and elapsed times, the number of physical disk I/Os we expect, an internally calculated value known as QUNITS and the actual dollar cost that can be used with your values for CPU and I/O if you bill your users. You can scan these metrics, looking for the most costly statements to focus on, or use the highlighted indicators on the left. The first two statements have a value of Y in the column under Q. The columns represent the CPU Time, Elapsed Time, Physical I/O, QUNITS and Cost columns to the right. You set user thresholds for each of these values, so that any statement that exceeds any of the thresholds will be flagged. From here you can navigate to the detailed EXPLAIN report for any of the statements.

```

SQLPARA10 ----- Query Limits Report ----- Row 1 of 4
Command ==>                                     Scroll ==> CSR

DB2 system: DSNR
DB2 SQLID : DRA560
Time      : 13:59

Commands: PR - PRINT

Line commands:
C - Cost Report  E - Explain Report  S - SQL  T - Trace Report  R - EEE Report
W - What-if?

S CCIQ$ Error Query No Type Time      Elapsed  Physical  QUNITS  Cost
Time      I/O
-----
--Y- 0      10000003 NB   0.00800 0.194    6         412     0.0018
--Y- 0      10000004 ST   0.00700 0.018    1         349     0.0016
----- 0      10000001 UE   0.00400 0.633    6         189     0.0009
----- 0      10000002 ST   0.00300 0.005    1         153     0.0007
***** Bottom of data *****
    
```

Figure 15: DB2 SQL Performance Analyzer for z/OS—Query Limits Report panel

Let’s quickly step through and highlight a few things in the reports. On this first screen (Figure 16) you can see the SQL text and, at the bottom, an easy-to-read box of text detailing something that might affect performance. Some of these will be SQL coding issues and others may be environmental issues. You can suppress some of these by setting user options for the analysis. For example, you may not want to see a box that tells you why select * is bad.

```

EDIT ----- DRA560.ANL410.EXP ----- COLUMNS 00001 00072
COMMAND ==>                                     SCROLL ==> CSR

000516
000517 SQL PA Analysis for Queryno 100000003
000518
000519 SELECT NAME, NTABLES, PARTITIONS, SEGSIZE
000520 FROM SYSIBM.SYSTABLESPACE
000521 WHERE CREATOR = 'SYSIBM'
000522 AND NAME IN (
000523 SELECT TSNAME
000524 FROM SYSIBM.SYSTABLES
000525 WHERE PARENTS > 0
000526 OR CHILDREN > 0)
000527
000528 Queryno: 100000003 QBlk: 1 Pln: 1 Mix: 0 Prnt QBlk: 0 Prnt Pln: 0
000529 Process ->
000530
000531 +----- Queryno 100000003 -----+
000532 |ANL70021 *** GUIDELINE:
000533 |This plan step has not selected any Sequential/List Prefetch I/O.
000534 |If the SQL processes just a few rows that is OK, but if many rows
000535 |are involved, you can help promote Sequential Detection by both
000536 |accessing the data in sequential order (presort?) and by binding
000537 |with Release (Deallocate) to avoid resetting counters at Commit.
    
```

Figure 16: DB2 SQL Performance Analyzer for z/OS—EXPLAIN report

The screen depicted in Figure 17 shows the access path information for an object, in this case an index scan. You also see the columns defined for that index and catalog statistics. If there are other available indexes defined for the object, you’ll see those listed, as well, and you could consider rewriting your query in such a way that it might make use of one of the alternate indexes.

```

EDIT ----- DRA560.ANL410.EXP ----- COLUMNS 00001 00072
COMMAND ==>                                     SCROLL ==> CSR

000570 RANDOM NONMATCH IX SCAN
000571
000572 IX Creator: SYSIBM
000573 Index Name: DSNDSX01
000574
000575 Vers: 1 Key Len: -1 Padded: N C-red: N C-ing: Y CtoRatio: 62.0000
000576 Fullkey card: 391 Firstkey card: 28
000577 Type: 2 Nleaf pages: 4 Nlevels: 2 Unique: P PRIMARY KEY
000578 0 of 2 columns are matched Close: N Lock mode: IS Bpool: DP0
000579 Ex Type: Key Target Cnt: 0 Unique Cnt: 0 Compress: N Owner Typ:
000580 Sparse index: - Hash overflow: -
000581
000582 Key Column Name Type Len DDNPGMH Colcard Dist# Hist#
000583 -----
000584 1 DBNAME VARCHAR 24 ANN--- 28 10 0
000585 2 NAME VARCHAR 24 ANN--- 276 0 0
000586
000587 +----- Queryno 100000003 -----+
000588 |ANL60521 *** NOTE:
000589 |This index is specified as "Not Padded", allowing storage of a
000590 |varying length index key. Padded indexes use blanks to fill out
000591 |their fixed length keys and are not eligible for Index Only scan.
    
```

Figure 17: SQL Performance Analyzer for z/OS—EXPLAIN report

This last screen (see Figure 18) shows, for tables that are defined with DB2 referential integrity, all of the related tables. This can be very useful to show a SQL developer that the simple update statement they wrote—updating a single column in a table—is actually causing updates to a lot of tables.

```

EDIT      DDA560.ANL410.EXP      COLUMNS 00001 00072
COMMAND ==>                      SCROLL ==> CSR
000290
000299 Table is a Parent to SYSIBM .SYSRELS      in Relation: DSNDT0DR
000300 Primary key has 2 columns and Delete rule is: CASCADE
000301
000302 Table is a Parent to SYSIBM .SYSTABSTATS_HIST in Relation: DSNTI0HB
000303 Primary key has 2 columns and Delete rule is: CASCADE
000304
000305 Table is a Parent to SYSIBM .SYSTABCONST in Relation: DSNDT0CN
000306 Primary key has 2 columns and Delete rule is: CASCADE
000307
000308 Table is a Parent to SYSIBM .SYSCONSTDEP in Relation: DSNDT0CC
000309 Primary key has 2 columns and Delete rule is: CASCADE
000310
000311 Table is a Parent to SYSIBM .SYSCHECKS      in Relation: DSNDT0SC
000312 Primary key has 2 columns and Delete rule is: CASCADE
000313
000314 Table is a Parent to SYSIBM .SYSTABLES_HIST in Relation: DSNDT0HD
000315 Primary key has 2 columns and Delete rule is: CASCADE
000316
000317 Table is a Parent to SYSIBM .SYSTRIGGERS   in Relation: DSNDT0UT
000318 Primary key has 2 columns and Delete rule is: CASCADE
000319

```

Figure 18: DB2 SQL Performance Analyzer for z/OS—EXPLAIN report, referential integrity

IBM InfoSphere Optim Query Workload Tuner for DB2 for z/OS

Similar to the DB2 SQL Performance Analyzer for z/OS product on the host, IBM InfoSphere Optim Query Workload Tuner for z/OS can pull SQL statements from a lot of different sources—see Figure 19. A key source for SQL statements in routine product usage is the DB2 dynamic statement cache. You can easily pull in statements from the cache based on filtering criteria you specify, so as to quickly focus on the

most expensive dynamic SQL you find. A recent enhancement allows you to pull in data from SQL monitoring tools, such as the IBM DB2 Query Monitor for z/OS. Clients have also used this capability with input from other vendors' monitoring tools. Remember that DB2 Path Checker can write SQL statements that have different access paths or higher costs to an XML file.

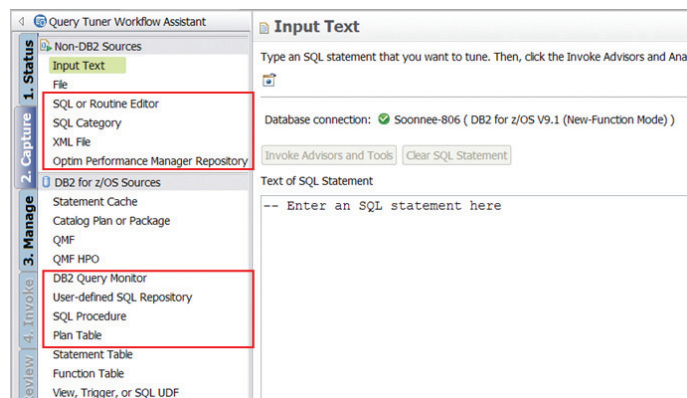


Figure 19: InfoSphere Optim Query Workload Tuner for z/OS—Capturing workloads from various sources

You may be using the EXPLAIN tools that IBM provides to you at no charge, such as IBM Data Studio or IBM Optimization Service Center for DB2 for z/OS (OSC), so a little of what you'll see on some of the screenshots will look similar to what you've been using. If you're still using OSC, you will need to switch to IBM Data Studio with DB2 10.

The screen in Figure 20 is the query formatter. This can take a very complex SQL statement and make it more easily readable. It also adds annotations to the query with data from the catalog. You'll notice also some highlighted icons on the lower left that are indicators that these tables have skewed data, and may benefit from better statistics.

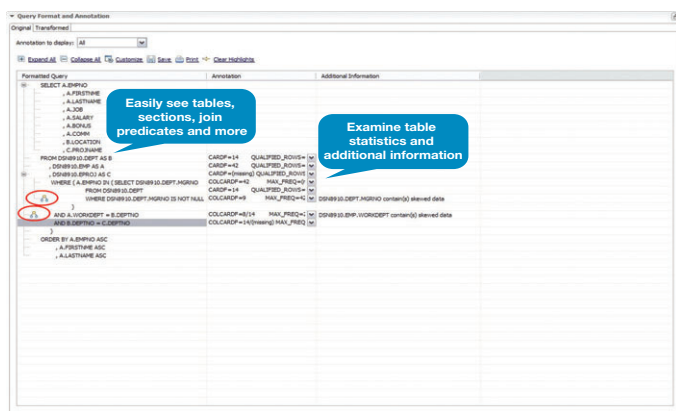


Figure 20: InfoSphere Optim Query Workload Tuner for z/OS—Query Formatter

Figure 21 shows the workload statistics advisor screen. Shown are all of the tables in the analyzed workload that have statistics recommendations. You can choose which of the tables you want to get updated stats on and actually invoke RUNSTATS directly in the graphical interface or take the generated control cards and submit them in a batch job.

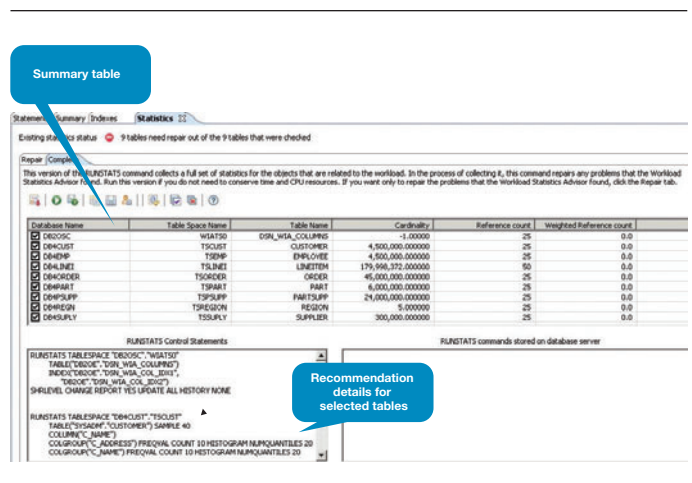


Figure 21: InfoSphere Optim Query Workload Tuner for z/OS—Workload Statistics Advisor

One important point to note about the recommended statistics is that they are based on the entire analyzed workload. So, if more than one statement accesses the same object you'll only get one generated RUNSTATS command.

Figure 22 illustrates a further look at the statistics advisor. You can see additional information in the report on the right, so you see when stats were last gathered and indications that there are conflicting or missing stats. You also get a detailed explanation regarding the conflicting statistics. Remember the indication of skewed data on the previous screenshot; this

advisor will build the necessary RUNSTATS commands to provide data on those columns with skewed data. The importance of having up-to-date, accurate stats available, so that the optimizer has the data it needs to choose the best possible access path, cannot be overemphasized. This feature of InfoSphere Optim Query Workload Tuner can save you a lot of time during and after your migration.

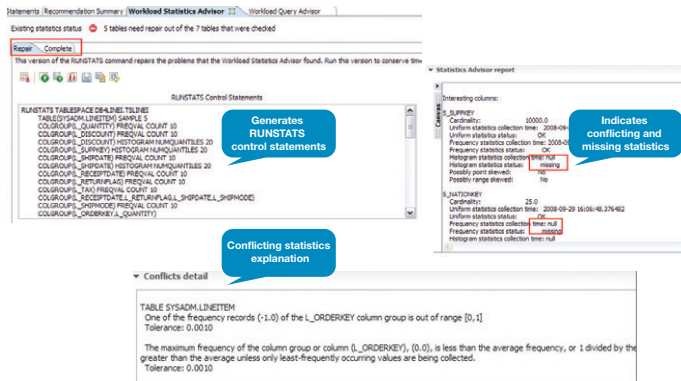


Figure 22: InfoSphere Optim Query Workload Tuner for z/OS—Statistics Advisor details

Figure 23 might look familiar to you. This is a visual depiction of the access path that is being used for a query. You can quickly see the flow of processing through DB2 and, as you move your pointer around the screen, pop-ups offer detailed information and advice. The image on the left contains a warning about statistics.

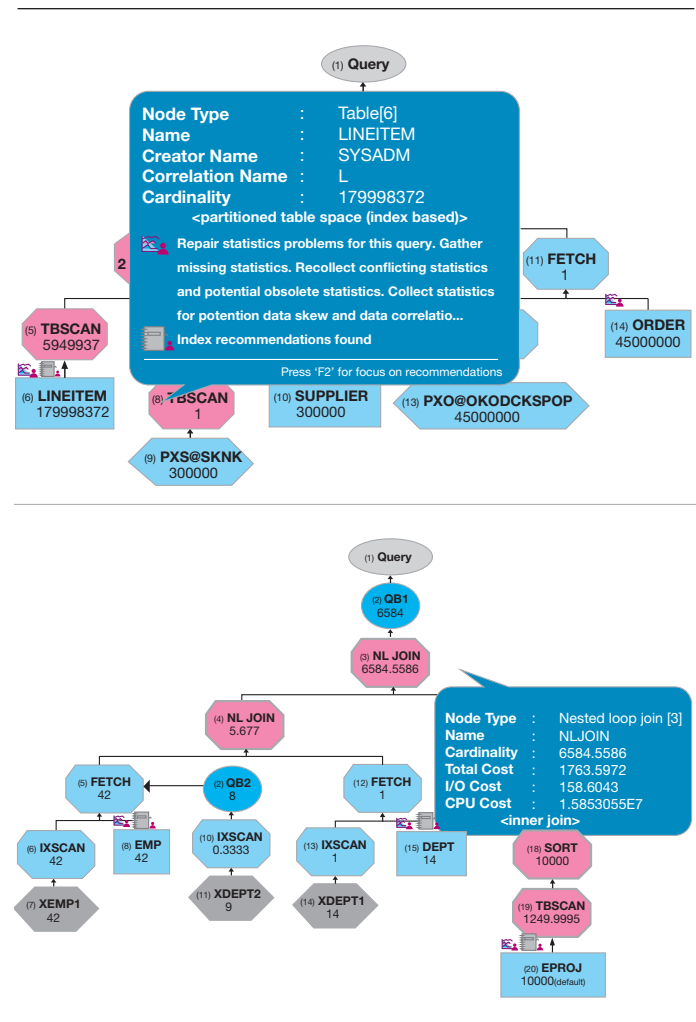


Figure 23: InfoSphere Optim Query Workload Tuner for z/OS—Visual EXPLAIN

There is also an access plan advisor (Figure 24), which can help you understand how you might influence the optimizer’s choice of an access path, and quickly pinpoint queries that are doing time-consuming processes such as tablespace scans and large sorts.

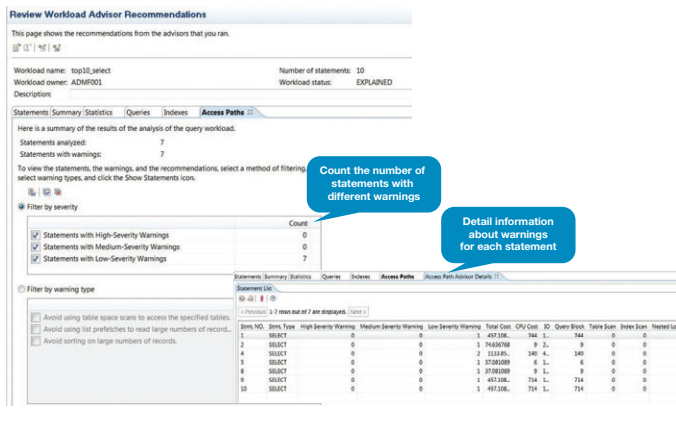


Figure 24: InfoSphere Optim Query Workload Tuner for z/OS—Access Plan Advisor

The last advisor is the index advisor, shown in Figure 25. This advisor can recommend new indexes that could improve performance. As with the other advisors, you can drill into considerable detail regarding the recommendations, and you can choose to create the index or use the virtual index capability to have DB2 do an EXPLAIN as though the index already exists, which will validate the tool’s recommendation.

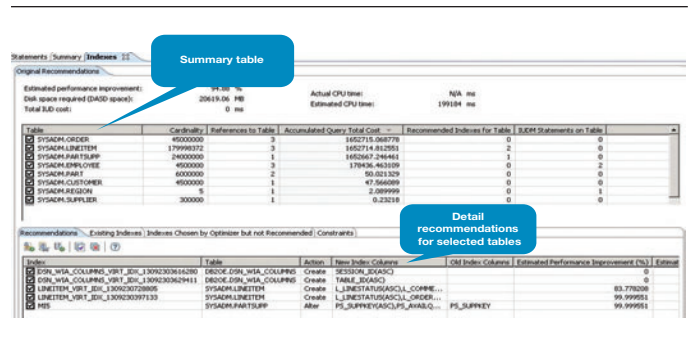


Figure 25: InfoSphere Optim Query Workload Tuner for z/OS—Index Advisor

If you’ll recall the discussion on binds, there may simply be some SQL that you will need to take back to your developers to help solve the problem. One very nice feature of the tool is the ability to generate an HTML report (see Figure 26) that you can use to send the information in the various advisors to a user. The users don’t need to have InfoSphere Optim Query Workload Tuner installed to see your analysis.

IBM Query Tuner Workload Summary Report

This report contains a summary of the recommendations from the workload advisors. Examine the recommendations and any corresponding scripts, and take any recommendations that were generated when the workload advisors analyzed the query workload, not when this report was generated. Navigate to the different sector [top of page](#).

Recommended Actions Statistics Indexes Access Paths Queries Run Log

Overview

Recommendation generation timestamp: 2011-06-27 22:19:14
 Database server configuration: jdbc:db2://dtec630 vmec.svl.ibm.com:446/STLEC1 (DSN10015)
 Owner of the query workload: ADMF001
 Default schema: ADMF001
 Workload name: top10_select
 Number of statements in the query workload: 10
 Number of statements analyzed in the query workload: 10
 Workload description:

Advisor	Warnings			Recommendation		Timestamp
	High	Medium	Low	RUNSTATS Commands	INDEX DDL Statements	
Workload Statistics Advisor	1	0	0	5	0	0 2011-06-19 19:30:23
Workload Index Advisor	0	0	0	0	0	0 2011-06-19 11:59:30
Workload Query Advisor	0	0	1	0	0	0 2011-06-19 11:59:22
Workload Access Path Advisor	0	0	0	0	0	0 2011-06-19 12:00:23

Figure 26: InfoSphere Optim Query Workload Tuner for z/OS—Workload Summary Report

Environment capture

If you haven't been successful in solving a performance problem and the developers haven't been able to help, either, then you will need to open a PMR to engage DB2 support to help you figure out what you can do to get a SQL statement back to its prior level of performance. The Environment Capture function can be used to automate the gathering of data the support team will request, such as object DDL and statistics. You'll need to indicate the PMR number you have open and the tool will gather the data and attach it to the PMR for you. This feature can save you a lot of time and effort.

Step 6: Reexamine KPIs after migration.

Once you have completed your migration to DB2 10, you should reexamine the KPIs by going back and performing steps 1 - 5 again.

Conclusion

Let's review the steps you need to take to ensure a successful migration.

1. Establish KPIs and performance baseline.
2. Manage packages that will have automatic rebinds.
3. Identify packages that will have improved access path performance.
4. Use triage techniques to manage problematic access paths.
5. Handle remaining access path performance problems with SQL tuning tools.
6. Reexamine KPIs after migration.

For more information

To learn more about migrating to IBM DB2 10 for z/OS, please contact your IBM marketing representative or IBM Business Partner, or visit the following website:

ibm.com/software/data/db2/zos

About the authors

John Casey, Consulting Technical Sales Specialist, IBM Software Group, authored this white paper. John's work focuses on IBM DB2 Tools for System z. He has more than 30 years of IT experience and has provided pre- and post-sales support to clients on systems, performance and data management products for more than 20 years. John currently works with medium to large accounts in the southeastern US, helping them to understand, install and configure IBM DB2 tools.

Mary Petras, Consulting Product Design Professional, IBM Software Group, contributed to this white paper. Mary works with the DB2 Tools Technical Specialists team at the IBM Silicon Valley Lab, and is responsible for supporting clients that have used IBM DB2 for z/OS for the past 11 years. Prior to that, she worked for the DB2 Development Group as a DB2 Lab Advocate and supported many DB2 for z/OS clients with their data-sharing implementation. Before joining IBM, Mary worked as a developer, technical support team leader, application programmer and mathematics professor. Her areas of expertise include DB2 data sharing, DB2 utilities, and performance and tuning. She has co-authored six IBM Redbooks®, including *DB2 9 for z/OS: Using the Utilities Suite*, *DB2 UDB for z/OS Version 8 Performance Topics*, *IBM DB2 Performance Expert for z/OS Version 2* and *DB2 UDB for OS/390 Version 6 Performance Topics*. Mary has also presented at the DB2 Technical Conference, at IBM Information On Demand (IOD), at the International DB2 Users Group (IDUG) and to local user groups.

Appendix A

Because there is overlap between InfoSphere Optim Query Workload Tuner and DB2 SQL Performance Analyzer, IBM clients often ask which they should choose. You can scan the points under each (see the table that follows), but the answer depends largely on how you respond to the following questions: Do you prefer working in a traditional ISPF and batch environment? Or, do you prefer graphical tools? Some

IBM clients choose both and give their users the choice. One thing to keep in mind if you also support DB2 on Linux, UNIX or Windows (LUW) is that InfoSphere Optim Query Workload Tuner is also available for DB2 on those platforms. And, you use the exact same workstation client for IBM System z® as for LUW. The newest version of QWT is at a different level of Eclipse than RDz, so they're not currently compatible.

Comparison of query tuning tools for IBM z/OS®

InfoSphere Optim Query Workload Tuner	DB2 SQL Performance Analyzer
<ul style="list-style-type: none"> • Better support for modern development and dynamic SQL • Eclipse-based, GUI and stored procedure-driven • Integrates with OMEGAMOM and Query Monitor GUI interfaces • Provides virtual index capabilities for single queries and across workloads • Recommends rewriting queries only when improvements result • Has limited support to create statistics in Query Environment Capture and Workload Environment Capture service support • Strategic investment spans platform 	<ul style="list-style-type: none"> • Better support for classic z/OS environment, DBRMs, batch analysis and z/OS libraries • ISPF and batch-driven • Integrates with OMEGAMON, IBM VTAM® and Query Monitor 3270 interfaces • Creates real indexes for “what-if” scenarios for single queries • Offers a best practices Query Advisor with approximately 150 rules • Has full support for cloning statistics • Continued z/OS specific investment

Appendix B

This paper has focused on a very specific area related to ensuring good SQL performance, but you might benefit from other tools as you move to DB2 10. The tools listed in the table that follows can be beneficial during the migration and

after, to help you take advantage of new features of DB2 10. There is substantial exploitation of DB2 10 in all of IBM's tools, and a representative would be happy to talk to you about any areas where you feel you have a gap in your current tools.

Migration task/Capability	DB2 for z/OS tool
Create preproduction test environment (subsystem and object clones) to test DB2	DB2 Cloning Tool for z/OS
Convert LOBs to in-line; convert existing tables to Hash Access	DB2 Administration Tool for z/OS
Measure and track SQL performance before and after migration	DB2 Query Monitor for OS/390
Identify and free unused packages; reduce bind impacts	DB2 Bind Manager for OS/390
Undo and Redo Temporal Data	DB2 Log Analysis Tool for OS/390
Ensure access to critical DB2 10 core functions (during and after migration) with cost reductions and performance	DB2 Utilities Suite for OS/390, DB2 Sort for z/OS
Establish and maintain DB2 10 utility syntax as company standards	DB2 Utilities Enhancement Tool for z/OS
Take advantage of DB2 10 utility improvements that reduce CPU usage and elapsed time for increased productivity	DB2 Automation Tool for OS/390
Safeguard DB2 10 data	InfoSphere Guardium Data Encryption for DB2 and IMS Databases



© Copyright IBM Corporation 2012

IBM Corporation
Route 100
Somers, NY 10589

Produced in the United States of America
May 2012

IBM, the IBM logo, ibm.com, DB2, InfoSphere, OMEGAMON, Optim, Rational, System z, Tivoli, VTAM, z10, z/OS, and zEnterprise are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



Please Recycle