PLAN Migration

Setting up a DB2 10 Precheck Environment

Ulf Heinrich - SEGUS Inc
u.heinrich@segus.com
Agenda

- Death of plans, have no fear
  - What will happen to plans during the migration
  - How the DB2 automatic rebinds work and how you can avoid them!
- Pinpointing all areas of degradation risk
  - Plans containing DBRMS
  - Plans and packages bound prior to V7
  - Static and dynamic SQL tuning needs
- Predicting access path degradations
  - Setting up a DB2 10 environment
  - Prechecking access paths prior to migration
  - Looking for typical access path change patterns
Each new version of DB2 has advantages and disadvantages regarding performance and resources – because of enriched features and functions ...

- Synergy with z platform
- Safe query optimization
- Query/Access Path Performance Enhancements
- Performance improvements from distributed data facility
DB2 for z/OS version migration

Being well prepared reduces the risk of a bad migration

• Check prereq’s (z/OS, DB2 APARs, RACF,…)
  • fallback SPE APAR PK56922
    see also information APARs II14474/II14477
• Verify removed/deprecated features and functions
• Verify catalog statistics
• Pre-migration check DSNTIJPM discovers gotchas to lookout for
• Pre-migration job DSNTIJPA (delivered via APAR PM04968) discovers cleanup needs
DB2 for z/OS version migration

If you experience a severe application or performance error in DB2 10 you need to fallback to DB2 V8, or DB2 9.
DB2 for z/OS version migration

Beside the rare case where real errors force the need of a fallback, more often performance issues occur as a global REBIND is requested.

*Most performance improvements are implemented by migrating to DB2 10 and rebinding*

- If you migrate and rebind plans or packages, the CPU savings can be twice as much than without re-binding
- Some enhancements are implemented through normal DB2 activities through re-binding
- REBIND is needed to obtain the best performance and memory improvements
- Early DB2 10 performance benchmarking and customer experience has shown a 5 to 10% CPU reduction in transactions after re-binding
- REBIND is not required for migration to DB2 10, but REBIND is strongly recommended
DB2 for z/OS version migration

- The enhancements to aggressively merge views and table expressions to avoid materialization to work files is available in conversion mode after *rebind*.
- Getting the best performance improvements and eliminating regression does depend upon *rebind* in most situations.
- STMTID support (as well as 401 generation) requires *REBIND* in NFM.
- Eliminating performance regression may depend upon *REBIND*.
- To use the enhanced monitoring support functions, you must *rebind* or bind any existing pre-DB2 10 package in DB2 10 new function mode.
- Storage constraint relief depends upon *REBIND*.
- Changing to use RELEASE(DEALLOCATE) requires a *REBIND*.
- All plans containing DBRMs must be *rebound*.
- The residual predicate enhancements are available in conversion mode after a *rebind*.
- All packages that were last bound on V5 or lower must be *rebound*.
- Static SQL statements with DEGREE(ANY) for parallel processing should be rebound, or it will be serial.
- Improvements in access paths can be very significant, such as stage 2 predicates that can become stage 1.
"DB2 10 delivers by improving performance and reducing CPU usage. Most customers can achieve out-of-the-box CPU savings of 5 to 10 percent for traditional workloads and up to 20 percent for specific workloads."

REBIND is needed to obtain the best performance and memory improvements". 
DB2 for z/OS version migration

Measurements of IBM Relational Warehouse Workload (IRWW) with data sharing

Base: DB2 9 NFM REBIND with PLANMGMT EXTENDED
- DB2 9 NFM → DB2 10 CM without REBIND showed 1.3% CPU reduction
- DB2 10 CM REBIND with same access path showed 4.8% CPU reduction
- DB2 10 NFM brought 5.1% CPU reduction
- DB2 10 CM or NFM with RELEASE DEALLOCATE 12.6% CPU reduction from DB2 9

CPU Percent reduced from DB2 9
DB2 for z/OS version migration

DB2 9 has already started the process of moving the plan and package static statement storage above-the-bar

• Depending on the statement 5% - 90% moves
• To achieve the VSCR in the below-the-bar storage for the CT and PT, you need to rebind. The storage below the bar associated with CT and PT pages only occurs for plans and packages that are not rebound on DB2 10. Once they are rebound, this storage is allocated above-the-bar
• In addition, to get the most benefit from virtual storage relief, native SQL procedures created on DB2 9 need to be regenerated on DB2 10
DB2 for z/OS version migration

>> So please be kind. REBIND! <<
DB2 for z/OS version migration

Some gotchas to watch out for:
• Binding DBRMs directly into plans is no longer supported
  → For pre-existing plans you can use the COLLID parameter of the REBIND PLAN command to create packages.

If you execute a plan that is bound from DBRMs, DB2 performs an automatic rebind that creates packages from the DBRMs and binds those packages into a plan.
(DSN_DEFAULT_COLLID_plan-name)

However, the recommendation is to use REBIND with the COLLID option.
DB2 for z/OS version migration

Some gotchas to watch out for:

• The COLLID option of REBIND PLAN binds the DBRMs to packages, and binds the packages to the specified plan. COLLID applies only to plans to which DBRMs are bound directly.

• If the installation uses the RACF access control module, owners of plans with DBRMs need to explicitly rebind the plans to convert the DBRMs to packages.

Check APARs PM62876 & PM79925
DB2 for z/OS version migration

Some gotchas to watch out for:

- Plans and packages created before DB2 10 containing static SQL statements using parallelism
  → DB2 incrementally rebinds those packages and plans after migration

*Incremental rebinds can cause performance degradation, so you should manually rebind*

You should consider rebinding those packages after migration, as soon as your Version 10 system is stable.
DB2 for z/OS version migration

Some gotchas to watch out for:

- Queries with the LIKE predicate often evaluated by DB2
  → The predicate evaluation enhancements are available in CM without rebinding, but rebinding avoids having to generate the machine code dynamically at execution time

You should consider rebinding those packages after migration, as soon as your Version 10 system is stable.
DB2 for z/OS version migration

Some gotchas to watch out for:

- DB2 10 allows an access plan to overflow to work files and continue processing RIDs, even when one of the RID thresholds are encountered at run time.

  → The RID pool work file overflow enhancement is available in conversion mode. A rebind of applications is not required

However, you’re advised to rebind or bind applications to reset the RID thresholds that are stored within a package.
Some gotchas to watch out for:

- Plans and packages from DB2 V5 or earlier need to be rebound
  
  → DB2 automatically rebinds them if you set DSNZPARM ABIND=YES or COEXIST (AUTO BIND field of panel DSNTIPO)

You might experience an execution delay the first time that such a plan is loaded. Also, DB2 might change the access path due to the autobind, potentially resulting in a more efficient access path.
DB2 for z/OS version migration

Some gotchas to watch out for:

The (out-of the-box) improvements do require a REBIND in most situations, and that does mean checking and testing, but DB2 version changes also take testing, so combining the work for a dramatic improvement will work for many customers.
DB2 for z/OS version migration

Bottom Line:

The DB2 Optimizer has improved algorithms and a rewritten approach to handle performance information for tuning and for exceptions.

Improved algorithms widen the scope of optimization.

→ Exploit the enhancements you paid for!
DB2 for z/OS version migration

When to REBIND?
• Avoid incremental/auto- (RE-)BINDs

* Consider the enhanced RUNSTATS when doing a skip migration (new clusterration, DRF)
** life the 3R’s in operating your DB2
DB2 for z/OS version migration

How to REBINND?
... so there are strong reasons to exploit the new features and REBIND, but a REBIND can be surprising!

How to get the many improvements in performance of DB2 10, without the risk of regressions?
DB2 for z/OS version migration

How to secure access paths during version migration?

Options:

1. Trial and error – may be risky, only possible when you have enough time/resources to play around
2. No REBINDs at all – no exploitation of enhancements already paid for
3. Pre-Check the results – most controllable and most efficient
Securing Access Paths

– Option 1 Trial and error

DB2 10 REBIND can compare the new and old access paths at bind or rebind processing and indicate that a warning or error is issued when an access path changes (APREUSE/APCOMPARE)*.

→ REBIND in DB2 10 takes more CPU and elapsed time, but more concurrent REBINDs are possible (NFM only)

*Requires APAR PM33767, PM25678
There is no guarantee to succeed
Requires package to be rebound at least under DB2 9
Securing Access Paths

– Option 1 Trial and error

Can Package Stability help?
   Package Stability is an access path backup!

Pro’s
• Easy fallback

Con’s
• 15% - 20% CPU overhead for each REBIND
• 2 – 3 times more storage requirements for DSNDB01.SPT01

DB2 10 changes the default for PLANMGMGT from OFF (in DB2 9) to EXTENDED.
   → setting PLANMGMGT=OFF in DB2 10 falls back to pre 10 default
Securing Access Paths

- Option 1 Trial and error

Preparation using Package Stability:

- Make sure you have plan table data for all your access paths REBIND EXPLAIN(YES)

- Make sure you have APAR PK52522 applied to your DB2 V8 (if doing skip migration)

- Use ZPARM PLANMGMT default (EXTENDED)
Securing Access Paths

- Option 1 Trial and error

Migration using Package Stability:
- Get DB2 10 CM stable
- Set ZPARM STATSCLUS to ENHANCED (default)*
  - new clusterratio calculation + DRF
- (Run extended RUNSTATS)
- Execute global REBINDs
  → Current access path is DB2 10
  → Previous access path is a prior DB2 10 access path or prior version
  - Original access path will remain your prior version access path

*if doing skip migration
Securing Access Paths

– Option 1 Trial and error

Fallback using Package Stability:
- Do a REBIND PACKAGE … SWITCH(ORIGINAL)

Cleanup:
- Do a FREE PACKAGE … PLANMGMTCSCOPE(INACTIVE)

If you are satisfied with the resulting access paths move on to NFM
Securing Access Paths

- Option 2 No REBINDs at all

>> To rebind or not to rebind, “when?” is the question: Whether ‘tis nobler in the mind to suffer the slings and arrows of outrageous access paths, Or to take arms against a sea of troubles, And by rebinding, end them? To die: to sleep; No more; and by a sleep to say we end the heart-ache and the thousand natural shocks that old plans are their to. ‘Tis a consummation devoutly to be wished to die, to sleep. To sleep: perchance to dream: aye, there’s the rub; For in that sleep of death what dream may come when plans have shuffled off this mortal coil, must give us pause: There’s the respect that makes calamity of so long life for old plans and packages.<< by Roger Miller
Securing Access Paths

– Option 2 No REBINDs at all

No REBINDs at all is **NOT** the solution!

You’ll degrade performance by not REBINDing! (e.g. fast column processing – SPROCs, VSCR, etc. etc.)

Make sure you do
- Global REBINDs after DB2 version migration
- Global REBINDs after DB2 system maintenance
- Practice the 3Rs – REORG → RUNSTATS → REBIND
Securing Access Paths

Option 3 Pre-check

- Goal: Detection of potential worsened access paths before migration of the production system by comparing the access paths of a prior DB2 production system with the access paths on a DB2 10 test system based on the original production statistics, CPU, # of CPs, BPs, rid & sort pool,...)*
- without affecting the production system
- at the earliest possible stage
- for static and dynamic SQL

*just became possible by production modeling V9 APAR PM26475 & V10 APAR PM26973
Securing Access Paths

– Option 3 Pre-check

<table>
<thead>
<tr>
<th>SYSCOLDIST / SYSKETGTIDIST</th>
<th>SYSCOLSTATS</th>
<th>SYSROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARDF</td>
<td>COLCARD</td>
<td>CARDINALITY*</td>
</tr>
<tr>
<td>COLGROUPCOLNO / KEYGROUPKEYNO</td>
<td>HIGHKEY</td>
<td>INITIAL_INSTS*</td>
</tr>
<tr>
<td>COLVALUE / KEYVALUE</td>
<td>LOWKEY</td>
<td>INITIAL_IOS*</td>
</tr>
<tr>
<td>FREQUENCYF</td>
<td></td>
<td>INSTS_PER_INVOC*</td>
</tr>
<tr>
<td>HIGHVALUE</td>
<td></td>
<td>IOS_PER_INVOC*</td>
</tr>
<tr>
<td>LOWVALUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMCOLUMNS / NUMKEYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUANTILENO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSCOLUMNMS / SYSKEYTARGETS</th>
<th>SYSINDEXES</th>
<th>SYSTABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLCARDF / CARDF</td>
<td>CLUSTERING*</td>
<td>CARDF</td>
</tr>
<tr>
<td>HIGHKEY</td>
<td>CLUSTERRATIOF</td>
<td>EDPROC*</td>
</tr>
<tr>
<td>LOWKEY</td>
<td>DATAREPEATFACTORF</td>
<td>NPAGES</td>
</tr>
<tr>
<td>STAT_FORMAT</td>
<td>FIRSTKEYCARDF</td>
<td>NPAGESF</td>
</tr>
<tr>
<td></td>
<td>FULLKEYCARDF</td>
<td>PCTROWCOMP</td>
</tr>
<tr>
<td></td>
<td>NLEAF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NLEVELS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSCOLUMNMS / SYSKEYTARGETS</th>
<th>SYSINDEXPART</th>
<th>SYSTABLESPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LIMITKEY*</td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NACTIVEF</td>
</tr>
</tbody>
</table>

* Columns are not updated by RUNSTATS
  Columns are not updatable

© 2011 SOFTWARE ENGINEERING GMBH and SEGUS Inc.
Production Modelling
- Supports optimizer overrides for optimizer relevant system settings
  - New zparms
    - SIMULATED_CPU_SPEED
    - SIMULATED_COUNT
  - New SYSIBM.DSN_PROFILE_ATTRIBUTES
    - SORT_POOL_SIZE
    - MAX_RIDBLOCKS
    - For bufferpools
      - Same as the BP names shown at DSNTIP1
        KEYWORDS value of 'BP8K0' corresponds to BP BP8K0.
Setting up a Precheck system

for Static SQL &
for Dynamic SQL
Method of SQL Performance Precheck

Plan Table (DB2I V10) → Comparison → Impact-Analysis

- Worsened

Report

Improved or equal

New Access Path Pattern

Plan Table (DB2P V8 or 9) → DB2 10 behaviour
Repeatable SQL Performance Precheck

- DB2P Production
- DB2T V10
- DB2T Impact V10
- DB2P V10 CM
- DB2P V10 NFM
- Copy & PreCheck

© 2011 SOFTWARE ENGINEERING GMBH and SEGUS Inc.
Precheck Scenario for Static SQL

- **DB2P: Production**
  - original DB2 z/OS Version
  - Central PLAN_TABLE - EXPLAIN(YES)
  - SYSIBM.SYSPACKAGE consisting the package statements

- **DB2I: Impact Analysis**
  - DB2 z/OS Version 10 (CM, NFM)
  - Homogeneous System Copy (without user VSAM cluster)
  - Update of DB2 Catalog (10) with original production statistics
  - Copy of central PLAN_TABLE (prod)
  - Copy of SYSIBM.SYSPACKAGE (prod) package statements

- Vold / Vnew impact analysis can be redone whenever needed
Precheck Scenario for Static SQL - setup

DB2P production system

old Catalog Statistics

old Plantable

old SYSPKG

DB2I impact analysis

DB2 10 Catalog Statistics

DB2 10 Plantable Shadow

DB2 10 SYSPKG Shadow

Dynamic Explain
Compare the access paths, not costs

ImpactExpert for DB2 z/OS ----- Comparison --------- LINE 00000077 COL 001 080
Command ===> Scroll ===> CSR
Mode: Precheck Static DB2: DB2I
Primary cmd: END, C(atalog data), D(etails on/off), S(tatement text)

Collection . RTDX0510
Package . . DSMORPB
StmtNo . . 332
Timestamp . . 2011-04-24-08.13.50.890000
Contoken . . 184177A60269695E
Bindtime . . 2010-10-23-14.41.12.401132

Access path before REBIND -------------- ! Access path with REBIND --------------

<table>
<thead>
<tr>
<th>TABLE</th>
<th>QB</th>
<th>PN</th>
<th>AC</th>
<th>MA</th>
<th>ME</th>
<th>IX</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTABLESPACE</td>
<td>1</td>
<td>1</td>
<td>R</td>
<td>0</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>SYSTABLEPART</td>
<td>1</td>
<td>2</td>
<td>I</td>
<td>2</td>
<td>1</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DSNDPX01</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATABASE</td>
<td>2</td>
<td>1</td>
<td>R</td>
<td>0</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE</th>
<th>QB</th>
<th>PN</th>
<th>AC</th>
<th>MA</th>
<th>ME</th>
<th>IX</th>
<th>PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTABLESPACE</td>
<td>1</td>
<td>1</td>
<td>R</td>
<td>0</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>SYSTABLEPART</td>
<td>1</td>
<td>2</td>
<td>I</td>
<td>2</td>
<td>4</td>
<td>N</td>
<td>L</td>
</tr>
<tr>
<td>DSNDPX01</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDATABASE</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>0</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Milliseconds: 32 ! Milliseconds: 28
Serviceunits: 125 ! Serviceunits: 110

Verify the access path changes
Precheck Scenario for Dynamic SQL

• **DB2P: Production**
  - original DB2 z/OS Version
  - Snapshot of Dynamic Statement Cache
  - Explain of all captured statements to central PLAN_TABLE

• **DB2I: Impact Analysis**
  - DB2 z/OS Version 10 (CM, NFM)
  - Homogeneous System Copy (without user VSAM cluster)
  - Update of DB2 Catalog (DB2 10) with original production statistics
  - Copy of original, central PLAN_TABLE
  - Copy of the production snapshot of Dynamic Statement Cache
  - Explain of all extracted statements to PLAN_TABLE (DB2 10)

• *old / new impact analysis can be redone whenever needed*
Precheck Scenario for Dynamic SQL

DB2P production system:
- Old Catalog Statistics
- Old Plantable

DB2I impact analysis:
- DB2 10 Catalog Statistics
- DB2 10 Plantable Shadow

Dynamic Explain

Dynamic Stmt Cache
- Capture DSC
- Explain DSC
- Dynamic SQL

DB2 10
- Plantable Shadow

© 2011 SOFTWARE ENGINEERING GMBH and SEGUS Inc.
Compare the access paths, not the costs

ImpactExpert for DB2 z/OS ----- Comparison --------- LINE 00000001 COL 001 080
Command ==> Scroll ==> CSR
Mode: Precheck Dynamic
Primary cmd: END, C(atalog data), D(etails on/off), S(tatement text)

RunID old . DSCSNP01
Created TS. 2010-10-24-09.23.39.408107
StmtID old. 355
ExplainID . 1

RunID new . DSCSNP01
Created TS. 2010-10-24-09.23.39.408107
StmtID new. 355
ExplainID . 2

Access path OLD -----------------------! Access path NEW ----------------------

TABLE             QB PN AC MA ME IX PR
INDEX             TY CO TH ON FT
--------------------- -------------------
IDUGY001           1  1 R   0 N  S
IDUGY0011
IDUGY002           1  2 I  1 1 N    ! IDUGY002           1  2 R   0 1  N  S
IDUGY0021 !
IDUGY008           1  3 I  1 1 Y    ! IDUGY008           1  3 I  1 1 Y
IDUGY0081 ! IDUGY0081
Milliseconds:          119 ! Milliseconds:            1
Serviceunits:          465 ! Serviceunits:            2
-------------------------------------------------------------------------------
Verify the access path changes
New Access Path Pattern V8

We encountered access path pattern of the following categories:

- **UNCHANGED** – Statements without AP changes
- **IMPROVED** – Statements with improved AP
- **DB2 V8 specific patterns** (examples)
  - V8 pattern 1 – SORT first QBLOCK
  - V8 pattern 2 – Tablespace scan instead of non matching index scan
  - V8 pattern 3 – usage of smaller index
  - ...
- **CHANGED** – Statements with changed AP (not classified)
- **WORSENED** – Statements with degraded AP
New Access Path Pattern V9

We encountered access path pattern of the following categories:

- **UNCHANGED** – Statements without AP changes
- **IMPROVED** – Statements with improved AP
- **DB2 9 specific patterns (examples)**
  - V9 pattern 1 – Usage of smaller index
  - V9 pattern 2 – Usage of bigger index, sort avoided
  - V9 pattern 3 – Usage of smaller index, index only get lost
  - ...
- **CHANGED** – Statements with changed AP (not classified)
- **WORSENED** – Statements with worsened AP
New Access Path Pattern V10

We encountered access path pattern of the following categories:

- UNCHANGED – Statements without AP change
- IMPROVED – Statements with improved AP
- DB2 10 specific patterns (examples)
  - V10 pattern 1 – Range list index scan
  - V10 pattern 2 – In memory in list
  - V10 pattern 3 – SQL pagination
  - ...
- CHANGED – Statements with changed AP (not classified)
- WORSENED – Statements with worsened AP
Using a different index

- Comparing DB2 V7 and V8, the latter showed a tendency to use smaller indexes (no. of pages)
- In general DB2 9 shows the same tendency, but ..
- In a lot of cases DB2 9 changed access paths in the way to prefer a bigger index to avoid a physical sort (similar to V7)
DB2 Optimizer Characteristics

Small tables

- With DB2 V8 a lot of matching or non-matching index scans changed to a tablespace scan for small tables

- In a lot of cases this behavior changed back to an index usage with DB2 9 even for tables with only 1 page
DB2 Optimizer Characteristics

Bad stats:

- With DB2 V8 we saw many bad access paths caused by bad statistics. For example, missing statistics for a new index lead to a tablespace scan, even if another index was used before.

- With DB2 9 Indexes with 0 pages seem to be always used.
DB2 Optimizer Characteristics

Index size (part 2)

- DB2 9 tends to use the smallest index
- even if index only access gets lost
- in most cases the smaller ix was the partitioning index
DB2 Optimizer Characteristics

Other changes

- Change Prefetch=S to D

- Changed join sequence (new random feature?)
  - Bigger table outside
  - Smaller table outside

- MIX -> Hybrid Join
DB2 Optimizer Characteristics

Other changes

- DB2 10 Range-List IX scan

  Taking affect when SQL has multiple OR, IN or other predicates that reference the same index

  Improves SQL processing against an index when multiple WHERE clauses can all reference the same index.
DB2 Optimizer Characteristics

Other changes

- DB2 10 Optimizer uses more parallelism
  - index reverse scan for a table
  - SQL subquery is transformed into join
  - multiple column hybrid join with sort composite
  - leading table is sort output and the join between the leading table and the second table is a multiple column hybrid join
The DB2 10 Optimizer avoids further sorts

- Multi-index access, list prefetch and sort

→ Single matching index access with sort avoided
Statistics

- REBINDs without distribution stats
  - Improved: 4 %
  - Degraded: 6,1 %
  - Unchanged: 79,9 %
  - Changed: 10 %

- REBINDs with distribution stats on everything
  - Improved: 13,2 %
  - Degraded: 0,9 %
  - Unchanged: 79,9 %
  - Changed: 6 %
Statistics

- REBINDs in a critical online application, well tuned
  - Improved: 4,9 %
  - Degraded: 1,3 %
  - Unchanged: 70,6 %
  - Changed: 23,2 % (not categorized)

- REBINDs in an uncritical application, batch
  - Improved: 4,2 %
  - Degraded: 5,8 %
  - Unchanged: 82 %
  - Changed: 8 %
Conclusion

*It's all about statistics... – Garbage in, garbage out!*

From DB2 V8 on we have seen significant impact on the quality and scope of statistics on the optimizers decision

If you provide proper statistics, the DB2 Optimizer provides a good access path
Conclusion

*It’s all about statistics… – Garbage in, garbage out!*

**V8 Recommendation:**
Keep your Catalog Statistics current AND
determine DISTRIBUTION STATS

**V9 Recommendation:**
Keep your Catalog Statistics current AND
determine DISTRIBUTION STATS AND
determine HISTOGRAM STATS AND
capture enhanced clusterratio/DRF

**V10 Recommendation:**
Keep your Catalog Statistics current AND
determine DISTRIBUTION STATS AND
determine HISTOGRAM STATS
Summary

- IBM significantly improved the DB2 Optimizer with general as well as specific enhancements
- „Usual“ RUNSTATS doesn’t provide detailed statistics to enable the Optimizer to find the best access path
- Access path decisions of the Optimizer in DB2 z/OS 10 (CM, NFM) are different from prior DB2 z/OS – we found certain patterns
- Requirements regarding quality and efficiency to maintenance procedures (RUNSTATS and REBIND) are higher for DB2 10 and upcoming versions
  - Growth of tables
  - Column Distribution Statistics
  - Histogram Statistics
Ulf Heinrich

SEGUS Inc
u.heinrich@segus.com