

Setting Up a Precheck Environment for DB2 Version Migration

Roy Boxwell
SOFTWARE ENGINEERING



Abstract:

This presentation will focus on how to pre-check the affect of a DB2 version migration on

application performance prior to migrating your production system. We will show you how to set up

a test environment for pre-checking applications in a Vnext system in order to examine access path

changes. Step-by-step, we will explain how to set up a test environment for pre-checking the access

paths of applications for both static and dynamic SQL. Examples of the findings of typical access

path changes are shown in detail along with DB2 9 specific patterns.

Housekeeping

- We value your feedback please fill out the session evaluations after each session and complete the overall IBM IOD EMEA 2009 evaluation form
- Please visit the IOD experience and Expo Solution Centre
- Remember this is a non smoking venue
- Please switch off your mobile phones
- Please remember to use your badge at all time including the party.



Agenda

- DB2 for z/OS version migration
- Pre-checking impacts on applications
- Tasks for setting up Precheck environment
- Customer case study
- Access path changes migrating to DB2 9



Objectives:

- Learn details of the DB2 for z/OS version migration
- Discover how to pre-check impacts on applications
- See the tasks for setting up an Precheck environment
- Exploit experiences from Customer case studies
- Get introduced to typical access path patterns of DB2 9

DB2 for z/OS version migration

DB2 9 for z/OS Performance Topics

IBM



„DB2 9 for z/OS is an exciting new version, with many improvements in performance and little regression“

Use the functions that guide reduced CPU

Discover improved scalability and availability

Reduce TCO with more z/HP eligibility

DB2 9 for z/OS is an exciting new version, with many improvements in performance and little regression. DB2 9 improves availability and performance with new SQL and IAM functions. Optimization improvements include new SQL functions to optimize, improved statistics for the optimizer, better optimization techniques, and a new approach to providing information for tuning. V8 SQL procedures were not eligible to run on the IBM System z9 Integrated Information Processor (z/PP), but changing to use the native SQL procedures on DB2 9 makes the work eligible for z/PP processing. The performance of varying length data can improve substantially if there are large numbers of varying length columns. Several improvements in disk access can reduce the time for sequential disk access and improve data rates.

The key DB2 9 for z/OS performance improvements include reduced CPU time in many utilities, deep synergy with IBM System z hardware and z/OS software, improved performance and scalability for inserts and updates, increased SQL selectivity, and improved execution for complex multi-SQL

INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION

BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE

IBM Redbooks are developed
by the IBM International
Technical Support



DB2 for z/OS version migration

Each new version of DB2 has advantages and disadvantages regarding performance and resources – because of enriched features and functions ...

- Synergy with new hardware: zIIP, MIDAW, DS8000
- Insert / Update / Delete enhance- and improvements
- Query/Access Path Performance Enhancements
- Improved virtual storage usage below bar



DB2 for z/OS version migration

Being well prepared reduces the risk of a bad migration

- Check prereq's (z/OS, DB2 APARs, RACF,...)
- Verify removed/deprecated features and functions
- Verify catalog statistics
- Pre-migration health check DSNTIJP9 (PK31841)



A couple of DB2 9 prereq's:

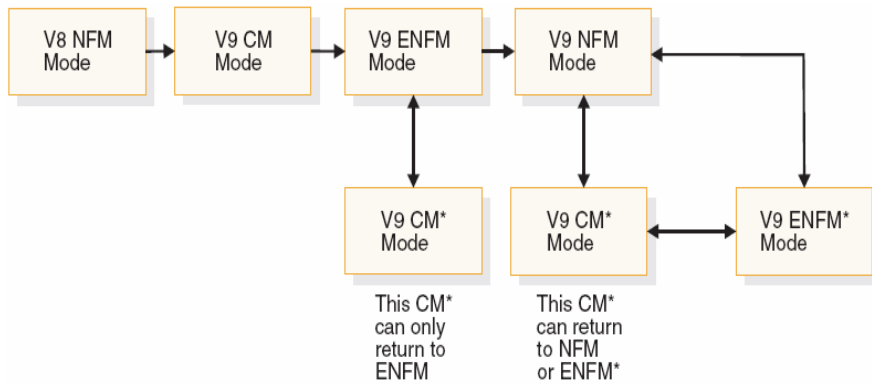
- z/OS >= 1.7
- DB2 V8 NFM
- BSDS already in expanded format
- Fallback prereq's SPE APAR PK11129 (PTF PK90008)

A couple of things you'll get with DB2 9:

- DB2 managed SPAS converted to WLM SPAS
- Increase Catalog TS and IS size
 - 33% more table spaces, 18% more indexes, 30% more columns, 13% more table check constraints (w/o XML repository)
- Temp DB no longer used – WORKFILE DB now used for system and user defined temporary table spaces
- Simple TS deprecated

DB2 for z/OS version migration

But if you experience a severe application or performance error in DB2 9 you need to fallback to DB2 V8.



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.

- **Note:** The DISTINCT and GROUP By enhancements are available in conversion mode. REBIND is required to obtain a sort avoidance benefit.
- **Note:** The dynamic prefetch functionality is available in conversion mode after a REBIND.
- **Note:** The global query optimization functionality is available in conversion mode and requires a REBIND.
- **Note:** The enhanced page-range screening functionality is available in conversion mode and requires REBIND.
- **Note:** Generalized sparse indexes and in-memory data caching are available in conversion mode after a REBIND.
- **Note:** Dynamic index ANDing for a star join query is available in conversion mode and requires a REBIND.
- **Note:** The histogram statistics over a range of column values are available in new-function mode with a REBIND required.



DB2 for z/OS version migration

DB2 9 has 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 your plans and packages in DB2 9 to move the relevant sections to 64-bit addressing



DB2 for z/OS version migration

Bottom Line:

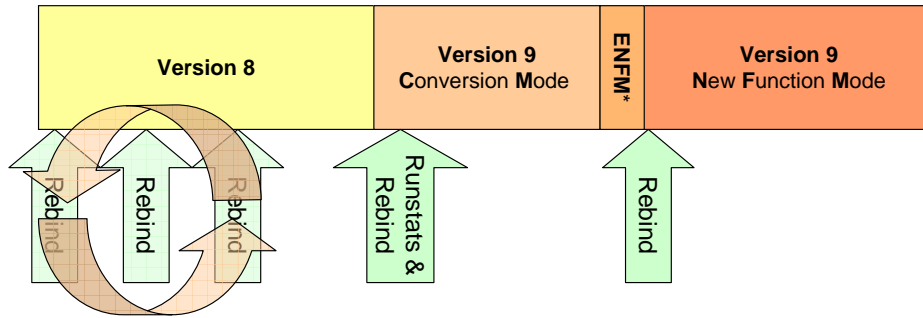
The DB2 9 Optimizer has improved algorithms and a rewritten approach to handle performance information for tuning and for exceptions.

Improved algorithms widen the scope of optimization.

→ Exploit the enhancements you paid for!



DB2 for z/OS version migration



* ENFM: Enabling New Function Mode



DB2 for z/OS version migration

... 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 this „exciting new version“ only, without the „little regressions“?



DB2 for z/OS version migration

How to secure access paths during version migration?

Options:

1. Trial and error – very 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

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 – use usermod for compression



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

- SET DB2 9 ZPARAM PLANMGMT to EXTENDED*

- * Watch out for the 64GB limit



Securing Access Paths – Option 1 Trial and error

- Migration using Package Stability:
 - Get DB2 9 CM stable
 - Set ZPARM STATSCLUS to ENHANCED (default)
 - this opens up the new clusterratio calculation
 - this opens up data repetition factor calculation (DRF)
 - Run extended RUNSTATS
 - Execute global REBINDs
 - Current access path is DB2 9
 - Previous access path is a prior DB2 9 access path or V8
 - Original access path will remain your V8 access path



Securing Access Paths – Option 1 Trial and error

- Fallback using Package Stability:

- Do a REBIND PACKAGE ... SWITCH(ORIGINAL)

- Cleanup:

- Do a FREE PACKAGE ... PLANMGMTSCOPE(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)

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 DB2 V8 production system with the access paths on a V9 test system (based on V8 production statistics)
- without affecting the production system
- at the earliest possible stage
- for static and dynamic SQL



By comparing the DB2 V8 production system against a DB2 9 test system, a potential access path degradation can be detected already BEFORE even starting with the migration.

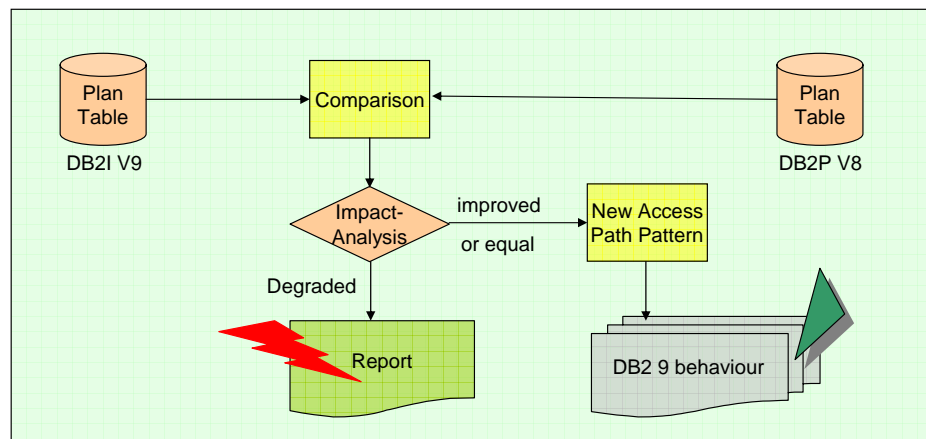
Setting up a Precheck system

- for Static SQL
- for Dynamic SQL



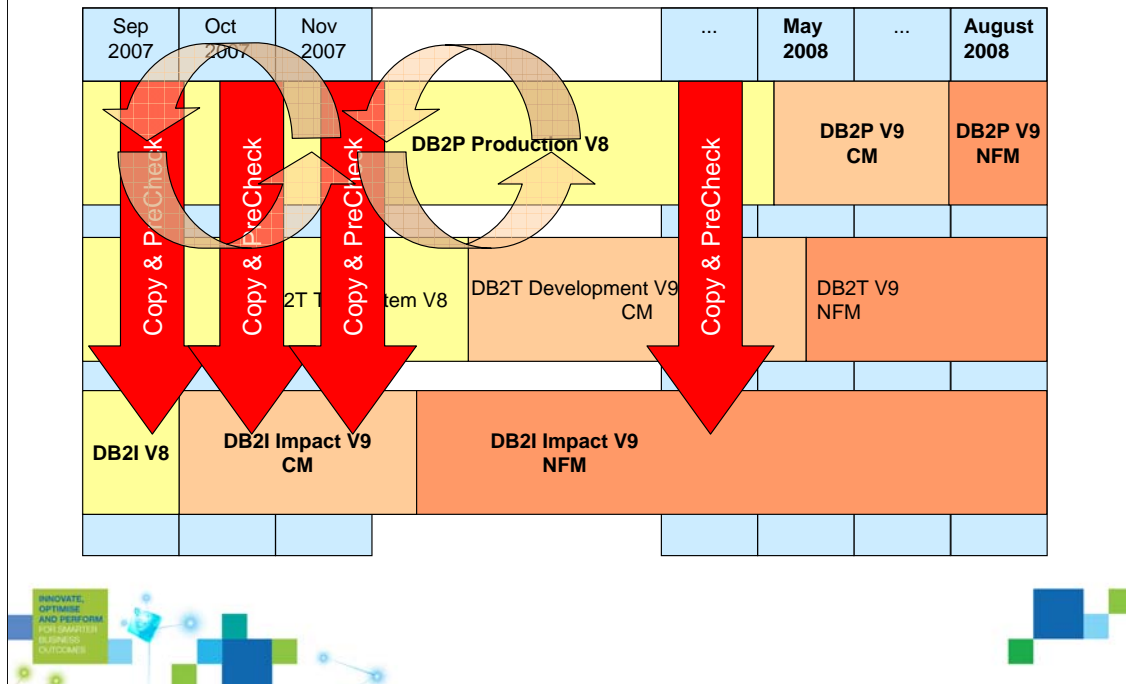
Title page for part 2 of presentation.

Method of SQL Performance Precheck



1. Take the actual access paths as they exist in the PLAN_TABLE under V8
2. As a basis for comparison, do an EXPLAIN to get the access path information for the V9 precheck system.
3. Compare the V8 and V9 access paths to find which ones will worsen under V9. We found a number of access path changes that will occur under V9 that can be expected (V9 access path rule set).

Repeatable SQL Performance Precheck



At the same time we migrated to V9, we were also coming out with a new release of the application. Development was ongoing while we prepared the migration so we had to build up a repeatable process to monitor access path changes also for application parts where development was in progress.

Of course, we wanted to repeat the process also in NFM in addition to the process in CM since we they stayed in CM for quite a while and application changes took place while being in CM - regardless it makes sense to check both stages of V9 migration, because some access path changes happen between CM and NFM.

Precheck Scenario for Static SQL

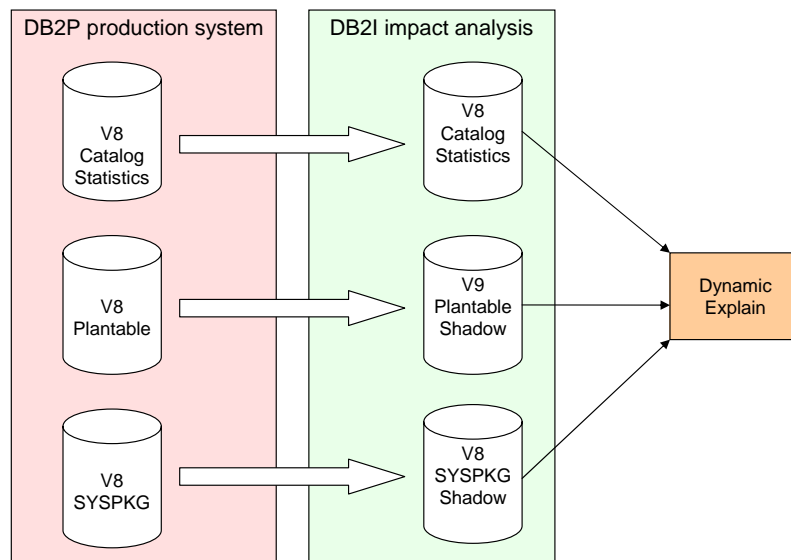
- **DB2P: Production**
 - DB2 z/OS Version 8
 - Central PLAN_TABLE - EXPLAIN(YES) (V8)
 - SYSIBM.SYSPACKAGE (V8) package statements
- **DB2I: Impact Analysis**
 - DB2 z/OS Version 9 (CM, NFM)
 - Homogeneous System Copy (without user VSAM cluster)
 - Object changes for DB2P also in DB2I
 - Update of DB2 Catalog (V9) with production statistics (V8)
 - Copy of central PLAN_TABLE (V8)
 - Copy of SYSIBM.SYSPACKAGE (V8) package statements
- V8 / V9 impact analysis can be redone whenever needed



This foil lists a precheck list for the V8 production system (DB2P) and the V9 analysis system (DB2I) for checking static SQL. From the V8 system, you will need the PLAN_TABLE and SYSIBM.SYSPACKAGE.

For the V9 analysis system, you will need V9 in CM or NFM mode, along with the production statistics from V8, a shadow of the V8 PLAN_TABLE, and a shadow of the V8 SYSIBM.SYSPACKAGE.

Copying catalog statistics, packages and access path information (static SQL) from DB2P



For prechecking static SQL, we simulated a production V9 system using our existing V9 test system (DB2I). We did this by transferring the V8 production catalog data, PLAN_TABLE data, and package information for static SQL from our production V8 system (DB2P).

Compare the access paths

```
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 Timestamp. . 2009-04-24-08.13.50.890000
Package. . . DSMORPB Contoken . . 184177A60269695E
StmntNo . . . 332 Bindtime . . 2008-10-23-14.41.1
```

Verify the access path changes

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

TABLE INDEX	QB	PN	AC	MA	ME	IX	PR	!	TABLE INDEX	QB	PN	AC	MA	ME	IX	PR
	TY	CO	TH	ON	FT			!		TY	CO	TH	ON	FT		
SYSTABLESPACE	1	1	R	0		N	S	!	SYSTABLESPACE	1	1	R	0		N	S
SYSTABLEPART	1	2	I	2	1	N		!	SYSTABLEPART	1	2	I	2	4	N	L
DSNDPX01								!	DSNDPX01							
SYSDATABASE	1	3		0	3	N		!	SYSDATABASE	1	3		0	3	N	
	2	1	R	0		N	S	!	DSNDDX02	2	1	I	0		N	
	2	2		0	3	N		!		2	2		0	3	N	
Milliseconds:				32				!	Milliseconds:				28			
Serviceunits:				125				!	Serviceunits:				110			



Precheck Scenario for Dynamic SQL

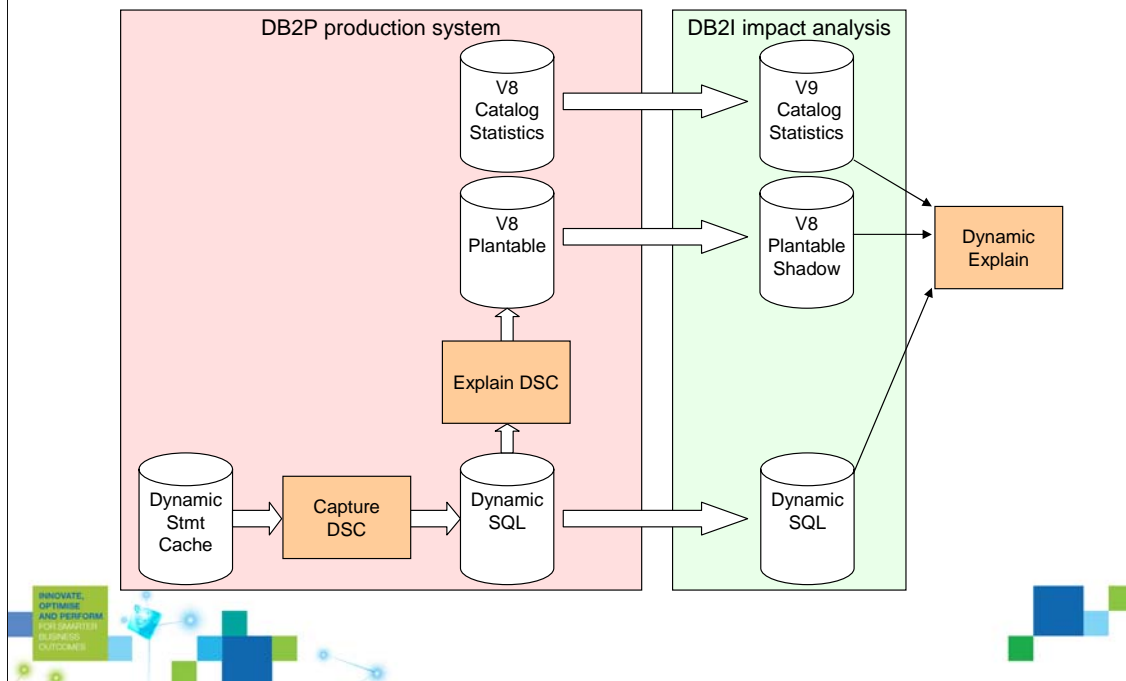
- **DB2P: Production**
 - DB2 z/OS Version 8
 - Snapshot of Dynamic Statement Cache (V8)
 - Explain of all captured statements to central PLAN_TABLE (V8)
- **DB2I: Impact Analysis**
 - DB2 z/OS Version 9 (CM, NFM)
 - Homogeneous System Copy (without user VSAM cluster)
 - Object changes for DB2P also in DB2I
 - Update of DB2 Catalog (V9) with production statistics (V8)
 - Copy of central PLAN_TABLE (V8)
 - Copy of snapshot of Dynamic Statement Cache (V8)
 - Explain of all extracted statements to PLAN_TABLE (V9)
- V8 / V9 impact analysis can be redone whenever needed



This foil lists a precheck list for the V8 production system (DB2P) and the V9 analysis system (DB2I) for prechecking dynamic SQL. From the V8 system, you will need the a snapshot of the dynamic statement cache and an explain of all captured statement in order to create a PLAN_TABLE with V8 access paths.

For the V9 analysis system of dynamic SQL, you will need V9 in CM or NFM mode, along with the production statistics from V8, the V8 snapshot of the dynamic statement cache along with their V8 access paths. On the V9 analysis system the V9 access paths will be determined for the comparison.

Copying catalog statistics, SQL and access path information (dynamic SQL) from DB2P



For prechecking dynamic SQL, we simulated a production V9 system using our existing V9 test system (DB2I). We did this by transferring the V8 production catalog data, V8 access path data for the dynamic SQL in the form of a PLAN_TABLE, and the snapshot of the dynamic SQL from our production V8 system (DB2P).

New Access Path Pattern V8

We encountered access path pattern of the following categories:

- UNCHANGED – Statements without Access Path Change
- IMPROVED – Statements with Improved Access Path
- DB2 V8 special 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 Access Path (not classified)
- WORSENER – Statements with Degraded Access Path



New Access Path Pattern V9

We encountered access path pattern of the following categories:

- **UNCHANGED** – Statements without Access Path Change
- **IMPROVED** – Statements with Improved Access Path
- **DB2 V9 special 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
 - V9 pattern 4 – TS scan to IX scan for small tables
 - V9 pattern 5 – Usage of index with 0 pages (bad stats)
 - V9 pattern 6 – Prefetch S to D
 - V9 pattern 7 – MIX to Hybrid Join
 - V9 pattern 8 – Join sequence change (outside smaller)
 - V9 pattern 9 – Join sequence change (outside bigger)
- **CHANGED** – Statements with Changed Access Path (not classified)
- **WORSENERD** – Statements with Degraded Access Path



SQL Performance Precheck Workflow

- Identifying changed access path by repeatable SQL PERFORMANCE PRECHECK impact analysis
- Categorizing all access path changes as „improved“, „unchanged“, „changed“ and „Degraded“
 - additionally classifying typical access path changes for a V8 / V9 transition to special V9 rule categories (rule 1, ..., rule 8)
- Manual checking of all SQL statements and access path changes of the categories „changed“ and „worsened“



1. Identify access path changes.
2. Categorize ALL access paths into 5 simple categories:
 - Unchanged
 - Degraded
 - Improved
 - Changed
 - V9 Specific patterns of access path changes
3. Categorization significantly reduces the number of statements that have to be manually checked. Only changed and Degraded categories need to be manually checked.

DB2 9 Optimizer Characteristics

- 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 V9 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 9 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 9 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



V9 Special Rules

- Other changes

- Change Prefetch=S to D
- Changed join sequence (new random function?)
 - Bigger table outside
 - Smaller table outside
- MIX -> Hybrid Join



Statistics

- V7 → V8 2005, no distribution stats
 - Improved: 4 %
 - Degraded: 6,1 %
 - Unchanged: 79,9 %
 - Changed: 10 %
- V7 → V8 2008, distribution stats on everything
 - Improved: 13,2 %
 - Degraded: 0,9 %
 - Unchanged: 79,9 %
 - Changed: 6 %



Statistics

- V8 → V9 2008 (critical online application, well tuned in V8)
 - Improved: 4,9 %
 - Degraded: 1,3 %
 - Unchanged: 70,6 %
 - Changed: 23,2 % (not categorized)
- V8 → V9 2008 (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 we have seen that the DB2 Optimizer makes as good a decision as it's knowledge base allows.

If you provide proper statistics, the DB2 Optimizer provides a good access path to the data.



Conclusion

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

V7 Recommendation:

Keep your Catalog Statistics current

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



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 V9 (CM, NFM) are different from DB2 z/OS V8 – we found certain patterns
- Requirements regarding quality and efficiency to maintenance procedures (RUNSTATS and REBIND) are higher for DB2 Version 9 and upcoming versions
 - **Growth of tables**
 - **Column Distribution Statistics**
 - **Histogram Statistics**



Session 1783A

IBM Information
>>> On Demand
2009

Setting Up a Precheck Environment for DB2 Version Migration

Roy Boxwell

SOFTWARE ENGINEERING

r.boxwell@seg.de

