Query Optimization and Performance with DB2 11 for z/OS

Central Ohio DB2 User Group (CODUG)

Troy Coleman, IBM DB2 Advisor zOS
Author: Terry Purcell, IBM Lead Architect DB2 Query Optimizer

May 13, 2016

Agenda

- Plan Management Usage
- Minimal intervention query performance
- In-Memory Data Cache (sparse index)
- DPSIs, page range & parallelism
- Misc Performance enhancements
- Optimizer externalization and statistics cleanup

Plan Management Usage

Static Plan Management - Target Usage

- Plan management provides protection from access path (performance) regression across REBIND/BIND
  - Access path fallback to prior (good) access path after REBIND
    - DB2 9 PLANMGMT(EXTENDED/BASIC) with SWITCH capability
  - DB2 10
    - Freeze access path across BIND/REBIND
      - **BIND/REBIND PACKAGE … APREUSE(ERROR)**
    - Access path comparison with BIND/REBIND
      - **BIND/REBIND PACKAGE… APCOMPARE(WARN | ERROR)**
  - DB2 11
      - **BIND/REBIND PACKAGE … APREUSE(WARN)**

DB2 11 Plan Management – APREUSE(WARN)

- DB2 10 delivered APREUSE(ERROR)
  - Allowed potential for reuse of prior plan to generate new runtime structure
  - Failure of reuse failed the entire package

- DB2 11 delivers APREUSE(WARN)
  - Upon failure of reuse, Optimizer will generate a new access path choice for that SQL
    - Thus failure of 1 SQL will not fail the entire package

APREUSE usage & implications

- Trade safety for potential CPU savings
  - Improved performance is one of the highlights of DB2 11
  - And the biggest gains often come from new access path choices
    - Example - one internal DB2 “query” workload had

      <2% CPU saving without REBIND (old runtime structure)

      <10% CPU savings with APREUSE (new runtime structure, old access path)

      >30% CPU saving without APREUSE (new access path)

      NOTE: this is NOT to demonstrate YOUR expected savings. Not all queries need new ap

- Migration is often a time when safety is desired
  - APREUSE(ERROR) in DB2 10 & 11 provides the most safety from change
  - May consider APREUSE(WARN) as 2nd step (after 1st step using ERROR)

Plan Management – Migration Preparation

- There is NO capability to FREE only an ORIGINAL copy
  - FREE PACKAGE PLANMGMTSCOPE(PLANMGMTINACTIVE)
    - FREEs both ORIGINAL and PREVIOUS

- ORIGINAL can become stale
  - The idea is to keep a “good and stable” backup in case of emergency
    - But it needs to be a recent good/stable backup

- Before migration to DB2 11
  - Perform FREE PACKAGE PLANMGMTSCOPE(PLANMGMTINACTIVE)
  - So that 1st REBIND in DB2 11 will save the pre-V11 CURRENT copy as ORIGINAL

  - BUT…..before doing that, read next slide……

DB2 11 and prior release package support

- DB2 11 supports packages from n-2 releases (DB2 9)
  - Pre-DB2 9 packages will be undergo AUTOBIND
    - AUTOBIND replaces the CURRENT which does NOT get saved as PREVIOUS/ORIGINAL
- REBIND all pre-V9 packages in V10 before DB2 11 migration
  - Any problems – REBIND SWITCH(PREVIOUS) in V10
    - ABIND=COEXIST will avoid AUTOBIND ping-pong in co-existence
  - Order of steps
    - REBIND all pre-V9 packages in V10. Once satisfied…….
    - FREE PACKAGE PLANMGMTSCOPE(PLANMGMTINACTIVE)
    - Migrate to DB2 11

Minimal Intervention Query Performance Improvements

Improve single matching index access options

- Improved predicate filtering – filtering rows earlier
  - Stage 2 predicate to indexable rewrites without “Index on Expression”
    - YEAR(DATE_COL)
    - DATE(TIMESTAMP_COL)
    - value BETWEEN C1 AND C2
    - SUBSTR(C1,1,10)
  - Single index access for OR IS NULL predicates
  - Indexability for IN/OR combinations
  - Push complex predicates inside materialized views/table expressions
  - Pruning (removing) “always true/false” literals (except “OR 0=1”)

Predicate Indexability & Plan management

- REBIND SWITCH takes you back to the prior runtime structure
  - If that is a pre-V11 plan, then that is pre-V11 predicate indexability improvements
- APREUSE/APCOMPARE occurs after query (predicate) transformations
  - May result in the prior plan NOT being available due to rewritten predicates
  - For example:
    - OR COL IS NULL rewritten to a single index plan – prior multi-index or range-list plan not available

      APREUSE(ERROR) would fail or APREUSE(WARN) would get a new plan
- Stage 2 to indexable rewrite may mean same index, but increase in matchcols
  - APREUSE(ERROR) would fail
    - No changes in plan are acceptable
  - APREUSE(WARN) would succeed with reusing prior plan
    - If only change is MATCHCOLS increase

Index skipping and Early-out

- Improvements to queries involving GROUP BY, DISTINCT or non-correlated subq
  - Where an index can be used for sort avoidance
    - By skipping over duplicates in the index
- Improvement to join queries using GROUP BY, DISTINCT (not apreuse friendly)
  - By NOT accessing duplicates from inner table of a join if DISTINCT/GROUP BY removes duplicates
- Improvement to correlated subqueries
  - Early-out for ordered access to MAX/MIN correlated subqueries
    - When I1-fetch is not available
  - Optimize usage of the “result cache” for access to subquery with duplicate keys from the outer query
    - 100 element result cache dates back to DB2 V2 as a runtime optimization

In-memory data cache / Sparse Indexing

Sparse index (in-memory data cache)

- Similar in concept to hash join in other RDBMSs
  - Controlled by zparm MXDTCACH (default 20MB)

- Improved optimizer usage and memory allocation in DB2 11
  - Each sparse index/IMDC is given a % of MXDTCACH
    - From optimizer cost perspective
    - At runtime (based upon cost estimation)
  - Runtime will choose appropriate implementation based upon available storage
    - Hash, binary search, or spill over to workfile

IMDC/Sparse index – Performance considerations

- DB2 11 provides simple accounting/statistics data for sparse index
  - Sparse IX disabled
    - Suggest reducing MXDTCACH or allocating more memory to the system
  - Sparse IX built WF
    - Increase MXDTCACH (if above counter is = 0) or reduce WF BP VPSEQT (if high sync I/O)

- Memory considerations for sparse index
  - Default DB2 setting for MXDTCACH is conservative
  - Customers generally undersize WF BP (compared to data BPs)
    - And often set VPSEQT too high (close to 100) for sort BP
  - If sync I/O seen in WF BP or PF requests & issues with PF engines
    - Consider increasing MXDTCACH given sufficient system memory
    - Consider increasing WF BP size and setting VPSEQT=90

DPSI, Page Range & Parallelism

DB2 11 Page Range Screening

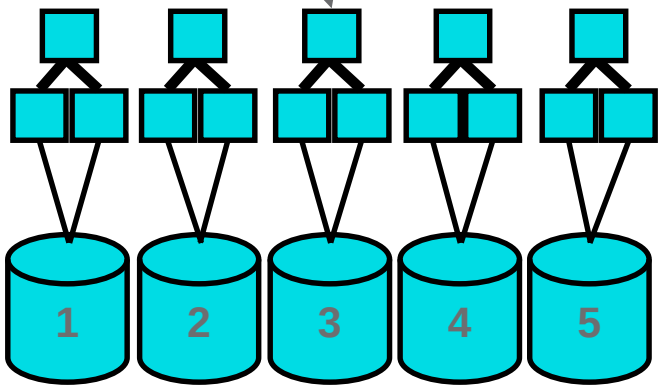
- Page range performance Improvements
  - Page Range Screening on Join Predicates
    - Access only qualified partitions
  - Pre-DB2 11, page range screening only applied to local predicates
    - With literals, host variables or parameter markers
  - Applies to index access or tablespace scan
    - Benefits NPIs by reducing data access only to qualified parts
    - Biggest benefit to DPSIs by reducing access only to qualified DPSI parts
- Only for equal predicates, same datatype/length

Page Range Join Probing (Join on Partitioning Col)

- Join recognizes page range screening
  - Only probe 1 part

YEAR	PARTNO
2011	1
2012	2
2013	3
2014	4
2015	5

T2
Partition by PARTNO
DPSI on ACCTNO




```
SELECT *
FROM T1, T2
WHERE T1.PARTNO = T2.PARTNO
AND T1.YEAR = 2013
AND T2.ACCTNO = 12345
```

Page range screening – who benefits?

- Page range screening enhancement is not workload dependent
  - Depends instead on a partitioning scheme
    - Where the partitioning column(s) include join columns,

      but an index supporting a join does NOT include the partitioning columns as leading columns

- Performance benefit?
  - No benefit if index is a PI
    - Since index columns match partitioning columns
  - No benefit if NPI and partitioning columns exist in index
    - Since predicates on partitioning columns would be index screening
  - Significant benefit up to 40% CPU reduction for DPSIs
    - NOT expected any customer is using DPSIs in this scenario today.
    - May allow switch to DPSIs for this scenario

DPSI – DB2 11 Enhancements

- DPSI can benefit from page range screening from join
  - Assuming you partition by columns used in joins (see previous slides)

- For DPSIs on join columns and partition by other columns
  - DB2 11 Improves DPSI Join Performance (using parallelism)
    - Controlled by ZPARM PARAMDEG_DPSI

- Sort avoidance for DPSIs (also known as DPSI merge)
  - Use of Index On Expression (IOE)
    - Ability to avoid sorting with DPSI IOE (already available for DPSI non-IOE)
  - Index lookaside when DPSI used for sort avoidance

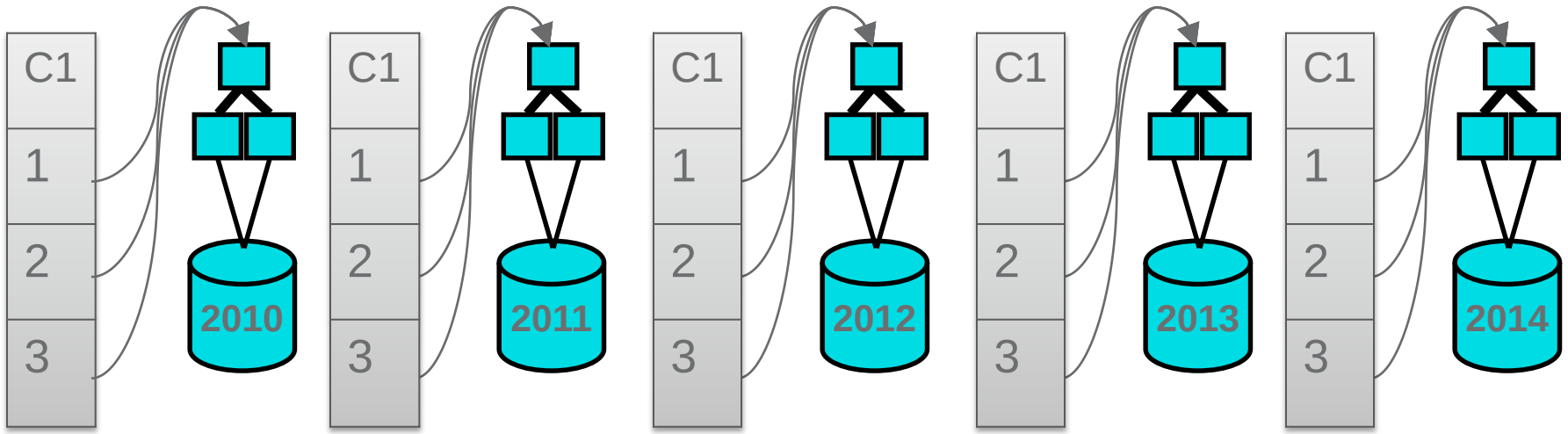
- Straw-model parallelism support for DPSI
  - Straw-model (delivered in V10) implies that DB2 creates more work elements than there are degrees on parallelism.

DPSI Join on Non-Partitioning Column

- DB2 11 DPSI part-level Nested Loop Join
  - Share composite table for each child task (diagram shows a copy)
    - Each child task is a 2 table join
    - Allows each join to T2 to access index sequentially (and data if high CR)

```
SELECT *
FROM T1, T2
WHERE T1.C1 = T2.C1
```

T2
DPSI on C1



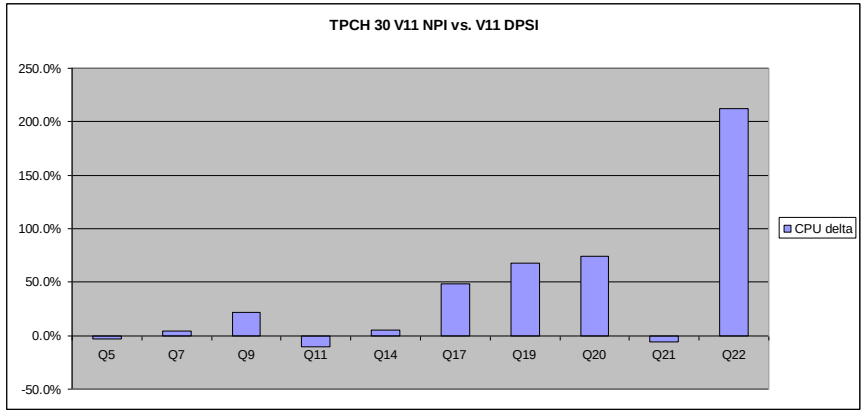
What does DB2 11 mean for DPSIs?

- A “partitioned” index means excellent utility performance
  - But historically there was one sweet spot ONLY for DPSIs
    - When local predicates in the query could limit partitions to be accessed

- Does DB2 11 allow me to switch all NPIs to DPSIs?
  - NO, but the sweet spot just got a little bigger
    - NPIs still are necessary in many workloads

- How do NPIs & DPSIs now compare?
  - Internal TPCH measurement
    - DPSIs increased CPU on avg by 8% vs NPIs

      But 1 query was 200% !!!!
  - DB2 11 ESP customer feedback
    - 2 customers reported > 75% CPU improvement for DPSIs (no other details provided)



Parallelism considerations

- Parallelism controls – default (‘1’) disabled
  - Static SQL – DEGREE bind parameter
  - Dynamic SQL – zparm CDSSRDEF or SET CURRENT DEGREE
- Number of degrees
  - Default PARAMDEG=0 which equals 2 * # of total CPs
    - Can be too high if few zIIPs
    - Conservative recommendation is 2 * # of zIIPs
- Parallelism requires sufficient resources

- DPSI performance can be improved with parallelism
  - Only DPSI part level join is controlled by zparm PARAMDEG_DPSI

Misc Performance items

CPU speed impact on access paths

- DB2 11 can reduce access path changes based upon different CPUs
  - CPU speed is one of the inputs to the optimizer
- Customers have seen CPU speed alter access paths
  - Across data sharing members
  - After CPU upgrade
  - Development vs production with different CPU speeds
- Less need to model production CPU speed in test in V11
  - Unless using Business Class machines
  - http://www-01.ibm.com/support/docview.wss?uid=swg21470440
    - Or google “DB2 production modelling”

Sort / Workfile Recommendations

- In-memory (from V9 to 11) is avoided if CURSOR WITH HOLD
  - Which is the default for ODBC & JDBC
- Ensure adequate WF BP, VPSEQT & datasets
  - Set VPSEQT=90 for sort (due to sparse index and/or DGTTs)
    - Evaluate sync I/Os in WF BP – may indicate sparse index spilling to WF
  - Provide multiple physical workfiles placed on different DASD volumes
  - Sort workfile placement example
    - 4-way Data Sharing Group
    - Assume 24 volumes are available
    - Each member should have 24 workfile tablespaces on separate volumes
    - All members should share all 24 volumes (i.e. 4 workfiles on each volume)

RID processing enhancements

- Pre-DB2 11
  - DB2 10 added RID failover to WF
    - Did not apply to queries involving column function
  - A single Hybrid Join query could consume 100% of the RID pool
    - Causing other concurrent queries to hit RID limit if > 1 RID block needed

- DB2 11
  - RID failover to WF extended to all scenarios when RID limit is hit
  - Hybrid join limited to 80% of the RID pool

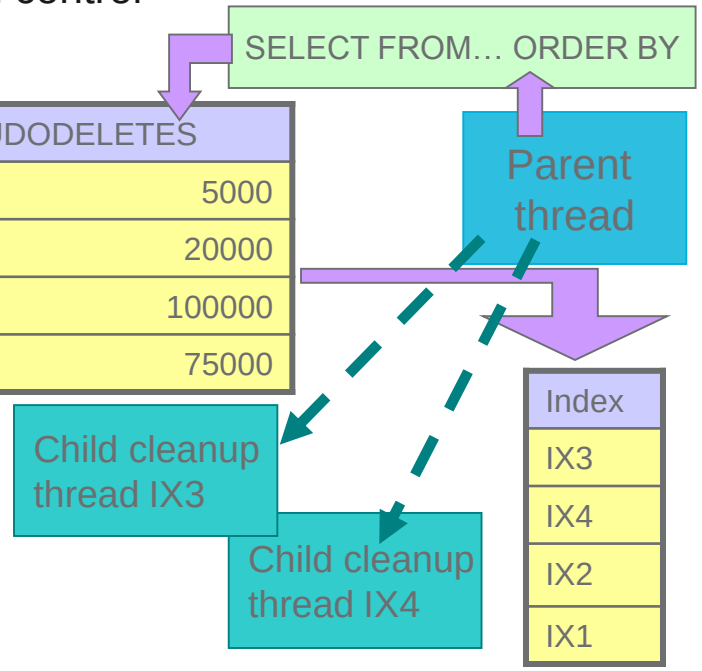
  - ZPARM MAXTEMPS_RID recommendation (DB2 10 & 11)
    - Set to NONE if failover to WF results in regressions

Reorg minimization enhancements – Indexes

- Pseudo-deletes
  - Index keys deleted/updated are marked pseudo-deleted and remain until REORG or when leaf page is full of pseudo-deletes
    - These degrade index scan performance
- DB2 11 adds automated clean up of pseudo-deletes
  - Cleanup is done under zIIP eligible system tasks
    - ZPARM INDEX_CLEANUP_THREADS to control # of concurrent tasks (default 10)
    - Catalog SYSIBM.SYSINDEXCLEANUP for table level control

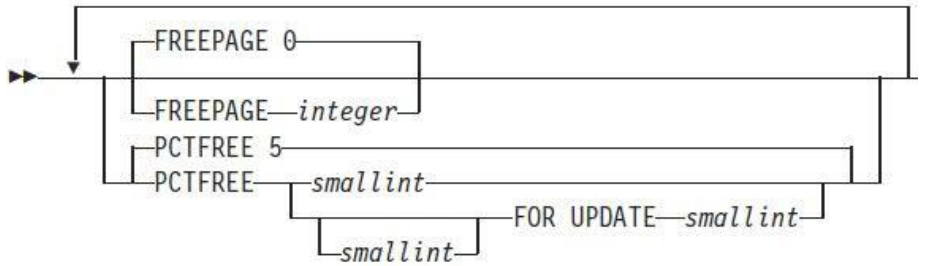
NAME	…	NPAGES	…	REORGPSEUDODELETES
IX1	nn	100	xx	5000
IX2	nn	1000	xx	20000
IX3	nn	500	xx	100000
IX4	nn	2000	xx	75000

SYSIBM.SYSINDEXSPACESTATS



Reorg minimization enhancements – TS Updates

- Indirect references
  - Update to var length or compressed row where row cannot fit in original location – DB2 will relocate row but leave original RID
    - Degrades data access since access to row requires extra getpage
- DB2 11 adds capability to allocate % free for updates
  - Leaves % space available during INSERTs or utilities
    - Utilities (LOAD/REORG) allocate the space, INSERT will not consume this
  - Zparm PCTFREE_UPD default 0, values
    - 0-99 (but may not want allocate value as system default)
    - Auto – uses RTS to determine %
  - Tablespace level control
    - 0-99, -1 means start with 5%, then RTS adjusts at REORG

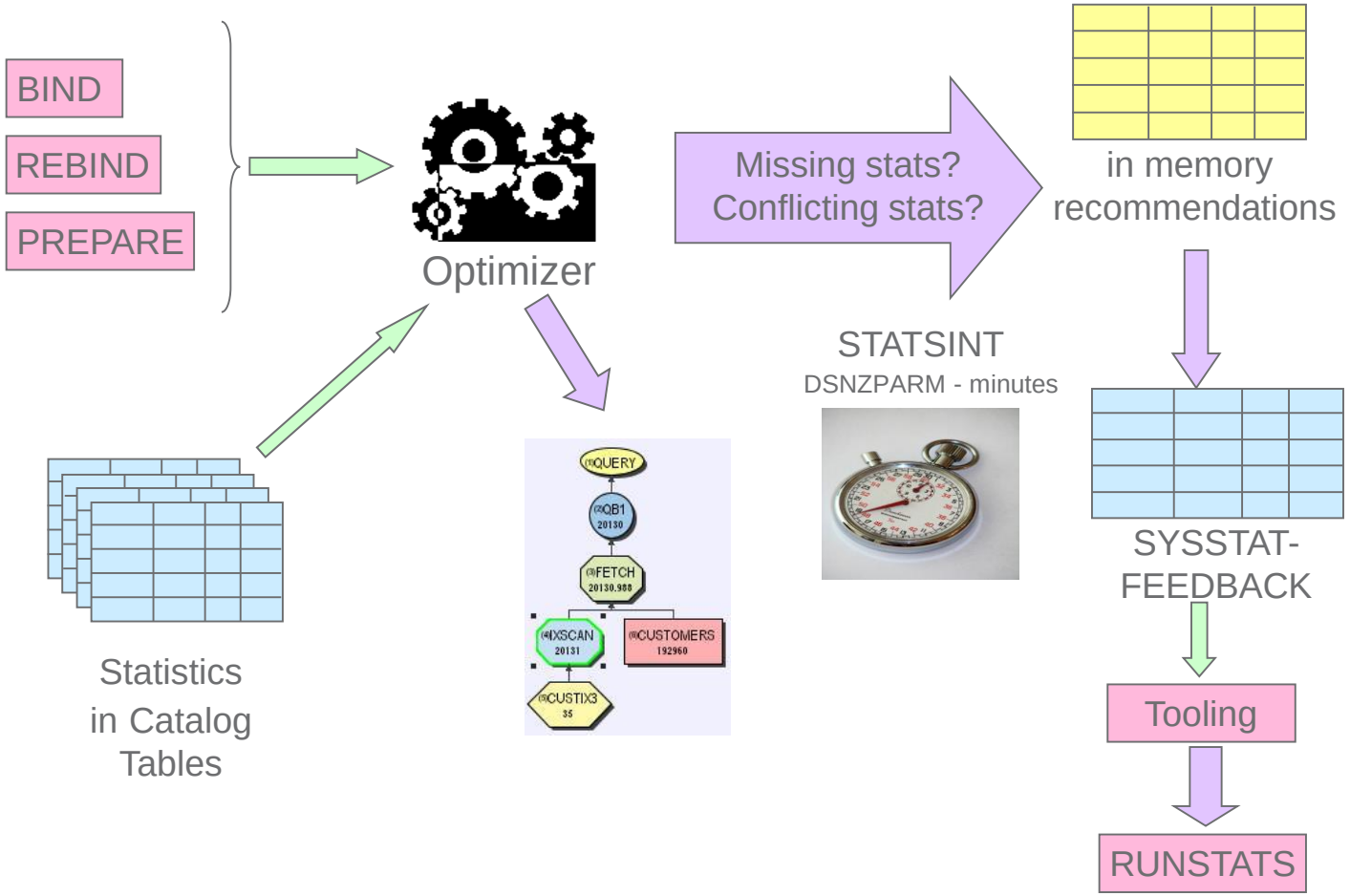


Optimizer externalization of missing statistics

DB2 Optimizer and Statistics - Challenge

- DB2 cost-based optimizer relies on statistics about tables & indexes
- Customers often gather only standard or default statistics
  - E.g. RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD
- Queries would often perform better if DB2 optimizer could exploit more complete statistics
- Customers have difficulty knowing which statistics are needed

DB2 11 – Optimizer externalization of missing statistics



DB2 11 Solution: Optimizer Externalization

- During access path calculation, optimizer will identify missing or conflicting statistics
  - On every BIND, REBIND or PREPARE
    - Asynchronously writes recommendations to SYSIBM.SYSSTATFEEDBACK (NFM)
  - DB2 also provides statistics recommendations on EXPLAIN
    - Populates DSN_STAT_FEEDBACK synchronously (CM if table exists)

- Contents of SYSSTATFEEDBACK or DSN_STAT_FEEDBACK can be used to generate input to RUNSTATS
  - Contents not directly consumable by RUNSTATS
  - Requires DBA or tooling to convert to RUNSTATS input

Optimizer Feedback - Controls

- Explain capability is available regardless of zparm value
  - Only requires existence of DSN_STAT_FEEDBACK table

- ZPARM STATFDBK_SCOPE
  - NONE – Disable collection of recommended RUNSTATS
  - STATIC – Collect for static queries only
  - DYNAMIC – Collect for dynamic queries only
  - ALL – Collect for all SQL (default)

- SYSTABLES.STAT_FEEDBACK updateable column (table control)
  - Y | N - indicates whether to externalize recommendations for this table
    - Yes is default. N means DB2 will not externalize for this table

- SYSSTATFEEDBACK.BLOCK_RUNSTATS updateable column (individual statistic control)
  - blank | Y – blank means okay to collect
    - Y(es) indicates to tooling or user that statistic should not be collected;
  - DB2 does not use this column as input, only tooling does

Recommendation to focus on

- Suggest focusing on these “FREQVAL” reasons
  - BASIC
    - Basic statistics are missing (TABLE(ALL) INDEX(ALL))
  - CONFLICT
    - There is a conflict between table & index statistics, or frequency & cardinality
    - Implies that statistics were run on different objects at different times
  - LOWCARD
    - Low cardinality column (often skewed)
  - NULLABLE
    - NULL is often the most frequently occurring value
  - DEFAULT
    - Implies column value “looks” like a default value (zero, blank, etc)
- Other reasons are targeted and may require further investigation

Further notes about interpreting recommendations

- DB2 is only recommending that a statistic could have been used
  - This is not a guarantee that the statistic is needed.
  - There is still a benefit to try to 1st determine whether collecting the statistic may add value
    - For a TYPE='F' recommendation – is the data really skewed?
    - What value to use for "COUNT integer"?

      10 is a good default

      If COLCARDF<=10, then use COLCARDF-1
  - REASON should also be considered
    - For example - TYPE='F',REASON='NULLABLE'

      If NULL is most frequently occurring, then you only need COUNT 1 (not 10)

Clearing out old statistics

- Old (stale) statistics
  - Customers often run “specialized” statistics as a once-off to try to solve an issue or as a prior default.
    - These old statistics can become stale and cause access path issues
    - Simplest way to find these is to look for tables with rows having different STATSTIMEs in SYSCOLDIST

- DB2 11 delivers
  - RUNSTATS reset option
    - Sets all relevant catalog values to -1, and clears tables such as SYSCOLDIST

    - Recommend running “regular” RUNSTATS after RESET

```
RUNSTATS TABLESPACE db-name.ts-name
TABLE table-name RESET ACCESSPATH
```

DB2 12: Taking DB2 to a New Level #DB2z

Redefining enterprise IT for digital business

DB2 12 Early Support Program Announced 6th Oct 2015

- **Scale and speed for the next era of mobile applications**
  - **Over 1 Million** Inserts per second
  - **256 trillion rows** in a single table, with agile partition technology

- **In Memory database**
- **23% CPU reduction** for lookups with advanced in-memory techniques
- **Next Gen application support**
- **360 million transactions** per hour through RESTful web API

- **Deliver analytical insights faster**
- **Up to 100 times speed** up for **targeted queries**

DB2 for z/OS On –line Communities #DB2z

- http://www.worldofdb2.com/
- IBMDB2 twitter **@IBMDB2**
- What's On DB2 for z/OS
- IDUG International DB2 User Group
- Facebook - DB2 for z/OS
- You Tube

Notices and Disclaimers

Notices and Disclaimers (con’t)

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, 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: www.ibm.com/legal/copytrade.shtml.

Thank You

Troy Coleman

Email: tlcolema@us.ibm.com