Getting the best query performance in DB2 11 for z/OS

Terry Purcell
DB2 for z/OS Optimizer Development
Welcome

Terry Purcell is a Senior Technical Staff Member with the IBM Silicon Valley Lab, where he is lead designer for the DB2 for z/OS Optimizer.

Surekha Parekh - WW Marketing Program Director – DB2 for z/OS
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
  - APREUSE(ERROR) EXPLAIN(ONLY) failure may not represent a valid plan in DB2 10
    - Failed access path is written to PLAN_TABLE

- DB2 11 delivers APREUSE(WARN)
  - Upon failure of reuse, Optimizer will generate a new access path choice
    - Thus failure of 1 SQL will not fail the entire package
  - PLAN_TABLE output will represent a valid plan
    - For both ERROR or WARN
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)
Minimal intervention
Query Performance
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 single index plan – prior multi-index or range-list 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 – DB2 11 Enhancements

• 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 outer query
  • 100 element result cache dates back to DB2 V2 as a runtime optimization
In-memory Data Cache/Sparse Indexing
In Memory Data Cache vs. Sparse Index

- **IMDC (hash or binary search)**
  - Hash or Binary Search to look up exact location of qualified key (Hash used if sufficient memory)
  - IMDC sorted in t2.c order

- **Sparse Index**
  - When insufficient memory for IMDC
  - Binary Search of sparse index to look up “approximate” location of qualified key
  - Sparse Index sorted in t2.c order
  - Workfile sorted in t2.c order
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 Screening
& Parallelism
Page Range Screening – DB2 11 Enhancements

• 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
V11 Page Range Join Probing (Join on Partitioning Col)

- Join recognizes page range screening
  - Only 1 partition needs probing.

SELECT *
FROM T1, T2
WHERE T1.PARTNO = T2.PARTNO
AND T1.YEAR = 2013
AND T2.ACCTNO = 12345
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
```
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)
DB2 11 Parallelism Negotiation if insufficient memory

- Parallelism now checks system storage and reduces degree if insufficient
  - Utilizes storage consumption based on LPAR values like common storage (ECSA, ESQA), paging, and aux consumption.

```plaintext
SQL Query & DEGREE ANY

Parallelism at BIND/PREPARE

QW0316AVGESTI or QW0401AVGESTI

ACCESS_DEGREE or JOIN_DEGREE in PLAN_TABLE

Parallelism at host variable/parameter marker substitution

QW0316AVGPLAN or QW0401AVGPLAN

New: Memory negotiation

#DEGREE

Existing: Buffer pool negotiation

Run

QW0316AVGACT or QW0401AVGACT
```
IFCID fields for Parallelism negotiation

- Older fields like QXDEGBUF and QXREDGRP are still valid and reflect specific DB2 internal limits, e.g.
  - Buffer pool negotiation: Calculate the buffer pools needed for this query and check the availability of these pools.
- New additional IFCID fields in DSNDQXST:
  - QXSTOREDGRP: Number of parallelism reduced due to system negotiation.
  - QXSTODGNGRP: Number of parallelism degenerated (sequential mode) due to system negotiation.
  - QXMAXESTIDG: Maximum parallelism estimated at bind/prepare time.
  - QXMAXPLANDG: Maximum parallelism after host variable/parameter marker substitution.
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

• Prior to DB2 11
  – May want to consider steps on next slides
Pre-DB2 11 - Stabilizing CPU models

- Mismatched CPU speeds can result in different access paths depending on which member the query is bound/prepared

- Use similar approach as production to test modelling
  - STEP 1: Connect to each member and issue SELECT for CPU speed only
    - CPU speed is likely to be the most influential difference

```sql
EXPLAIN ALL SET QUERYNO=6475 FOR
SELECT * FROM SYSIBM.SYSDUMMY1;

SELECT HEX(SUBSTR(IBM_SERVICE_DATA,69,4)) AS CPU_SPEED
FROM PLAN_TABLE WHERE QUERYNO=6475
```
Pre-DB2 11 - Stabilizing CPU models

- Continued
  - STEP 2: Convert the CPU speed from HEX to decimal
  - STEP 3: Choose the highest number from all of the results returned from the queries executed on each member
  - STEP 4: Set ZPARM SIMULATED_CPU_SPEED for each member
  - STEP 5: Recycle DB2 (ouch!)
  - STEP 6: Reissue the SQL below – showing the before and after CPU speed

```
EXPLAIN ALL SET QUERYNO=6476 FOR
SELECT * FROM SYSIBM.SYSDUMMY1;

SELECT QUERYNO, HEX(SUBSTR(IBM_SERVICE_DATA, 69, 4)) AS CPU_SPEED
FROM PLAN_TABLE WHERE QUERYNO IN (6475, 6476)
```
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
    – Example - Sort workfile placement
      • 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

<table>
<thead>
<tr>
<th>NAME</th>
<th>...</th>
<th>NPAGES</th>
<th>...</th>
<th>REORG_PSEUDO_DELETES</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX1</td>
<td>nn</td>
<td>100</td>
<td>xx</td>
<td>5000</td>
</tr>
<tr>
<td>IX2</td>
<td>nn</td>
<td>1000</td>
<td>xx</td>
<td>20000</td>
</tr>
<tr>
<td>IX3</td>
<td>nn</td>
<td>500</td>
<td>xx</td>
<td>100000</td>
</tr>
<tr>
<td>IX4</td>
<td>nn</td>
<td>2000</td>
<td>xx</td>
<td>75000</td>
</tr>
</tbody>
</table>

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 = start with 5%, then RTS adjusts at REORG
Optimizer externalization of missing stats
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

Statistics in Catalog Tables

Optimizer

Missing stats? Conflicting stats?

in memory recommendations

STATSINT
DSNZPARM - minutes

SYSSTAT- FEEDBACK

Tooling

RUNSTATS
DB2 11 Solution: Optimizer Externalization

- During access path selection, optimizer will identify missing or conflicting stats
  - 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
Interpreting Statistics Recommendations

- DB2 is only recommending that a statistic could have been used
  - This is not a guarantee that the statistic is needed.

- Still 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
    - But when is the data no longer skewed?
    - If COLCARDF<=10, then use COLCARDF-1

- REASON should also be considered
  - For example - TYPE='F',REASON='NULLABLE'
    - If NULL is most frequent, 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

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

• Recommend running “regular” RUNSTATS after RESET
Upcoming Webcasts

- **Webcast 1** - DB2 11 Migration Planning & Customer Experiences - PART 1  June 16th 11am EST - John Campbell  https://ibm.biz/BdXZvM


- **Webcast 3**. Opportunities of using large real memory with DB2 10 and 11 to reduce CPU resource consumption 30th June 11 am EST  John Campbell
  https://ibm.biz/DB2LargeMemory

- **Webcast 4** - Unleash Critical Information Locked up in DB2 for z/OS: 7th July Gain Insight, IMPROVE YOUR COMPETITIVE EDGE. with Namik Hrle IBM Fellow, Eberhard Hechler & Martin Dirk Schneider, https://ibm.biz/IBMIDAAWebcast
Thank-you & Final Logistics

• Thank-you for attending this webcast. The slides are available in the Resource Section on the Right Hand side

• Q&As during the webcast will be compiled in a document and added to resources

• This webcast will be available for at least 12 months on replay

• Feedback, comments on how we can improve this service via “The World of DB2”
Thank you!

Terry Purcell
tpurcel@us.ibm.com