IDUG
Leading the DB2 User Community since 1988

IDUG DB2 Australasian Tech Conference
Sydney, Australia | September 2016

#IDUGDB2

Real Time Analytics Query Performance with DB2 for z/OS

Terry Purcell
IBM Silicon Valley Lab

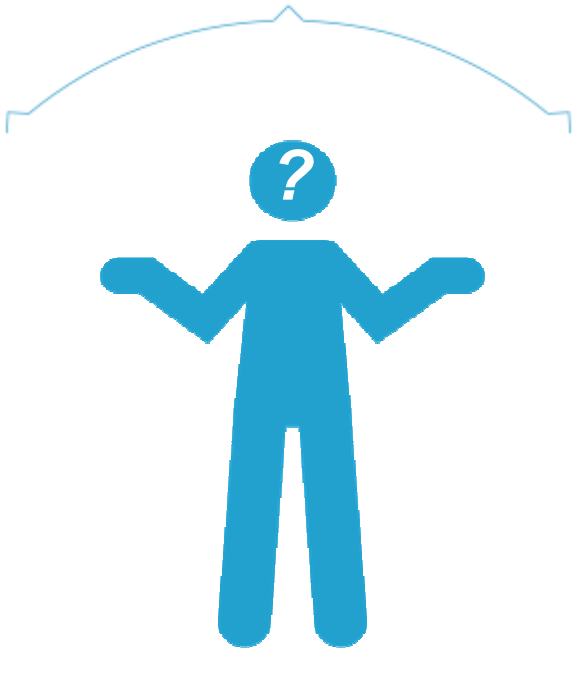
Session Code: Z9
Thursday Sept 15th 1:15pm –2:15pm| Platform: DB2 for z/OS

Where does DB2 z fit in the analytics space?

Where does DB2z fit into the Analytics landscape?

Where should each analytics workload reside?

What are the design considerations with Analytics on DB2 for z/OS?



How do I position IDAA and native DB2 for z/OS?

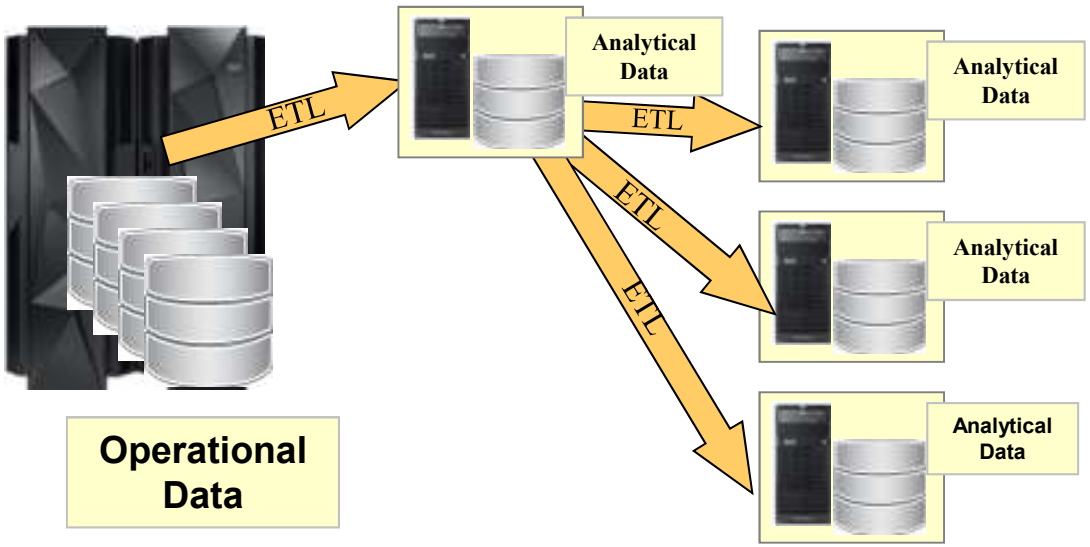
What is DB2's roadmap for Analytics?

Agenda

- Analytics – some perspective
- Analytics and DB2 for z/OS
- DB2 for z/OS Tuning Considerations for Analytics

BI/DW and Analytics Landscape with DB2 z

- Many customers have "off-platform" BI/DW environments
  - Moving DB2 z data regularly to these platforms
  - Often quoted that there are "7 copies on average" of OLTP data on other platforms used for analytics or other purposes
    - www.redbooks.ibm.com
    - "Reducing Data Movement Costs in z Systems Environments"
      - Focused on benefit of IDAA in reduced data movements
      - Also applicable to DB2 for z/OS directly



Tactical vs Strategic Analytics

- Excerpts - Blog post by Nin Lei (CTO, IBM Big Data & Analytics for zSystems hardware)
  - Strategic analytics
    - Year-over-year or month-to-month sales analysis.
    - Data mining to explore how to categorize customers' behaviors.
    - Fraud detection models to reduce losses.
    - Workloads consuming large amount of data, requiring a system architecture with massive parallelism.
  - Tactical analytics (or operational analytics)
    - More single-customer oriented and thus smaller/targeted data.
    - Identify items purchased by current customer in the past year to recommend the appropriate products for that customer.
    - A phone company call center rep wants to know the customers phone call patterns - to upsell the customer with a more profitable plan.
    - Fraud detection analysis of prior purchase history

Where does Strategic Analytics belong?

- Strategic analytics (traditional BI/DW)
  - The realm of "shared nothing" or highly parallelized architectures.
  - For DB2 for z/OS – IDAA is an excellent candidate to
    - Reduce data movement compared with other off-platform solutions
    - Benefit from zSystems quality of service
  - Native DB2 for z/OS requires targeted partitioning and indexing (and potentially MQTs) to deliver acceptable performance
    - Likely to discourage many customers due to skill/effort
  - IDAA has brought (strategic) analytics back to DB2 for z/OS

Where does Tactical Analytics belong?

- Tactical analytics
  - "In-transaction Analytics"
  - Operational analytics

  - Basically - If the analytics is part of the transaction
    - Then analytics should be performed on or close to the operational data.
      - IDAA is closer than other competitors solutions

  - "Tuned" analytics on DB2 for z/OS can meet transactional concurrency
    - IDAA typically better with poorly indexed workloads
    - DB2 for z/OS provides transaction concurrency
      - Depending on degree of tuning and resources

What is your definition of Real-time Analytics?

- Is the Analytics in real-time on near-real-time data?
  - Tactical analytics example:
    - Analysis of historical transactions for one customer
- Or are you performing analytics on (current) real-time data?
  - Any off-platform solution may promise “near real-time”
  - For analytics on real-time data
    - If you copy data from your OLTP system – its not real-time anymore
    - So only the OLTP system can be truly real-time
- Not all applications require “real-time” data

Agenda

- Analytics – some perspective
- Analytics and DB2 for z/OS
- DB2 for z/OS Tuning Considerations for Analytics

What do analytics queries look like?

- What are the attributes of an Analytics Query?
  - Analytics queries often involve
    - More tables in a query – joins, subqueries, table expressions etc
    - More complex expressions (in WHERE clause or select list)
    - More rows being processed when compared with OLTP
    - More rows joined, sorted etc
    - Views that contain joins or UNION ALL
      - Views may contain more columns/tables than query requires
- Today's OLTP workloads have similar complexity
  - CICS/COBOL applications were written with efficiency as part of the design
  - Todays workloads are developed for speed of deployment, not performance

DB2 Analytics Enhancements

- Recent DB2 releases have significant functional and performance focus on Analytics
  - DB2 9 through 11
  - Continuing in DB2 12
    - Major focus due to HTAP and also new SAP analytics workload requirements

- From DB2 9 thru 11 OLAP SQL function
  - Rank/Dense rank/Row number
  - Moving sum/average
  - Grouping sets, ROLLUP/CUBE
  - More to come in DB2 12

DB2 Analytics/Query Performance Focus

DB2 9

- Index on expression
- GROUP BY sort enhancements
- Histogram statistics
- Dynamic Index ANDing
- In-memory for small sorts and FETCH FIRST
- Parallelism cut on inner table
- Sparse index/In-mem cache
- Global subquery optimization

DB2 10

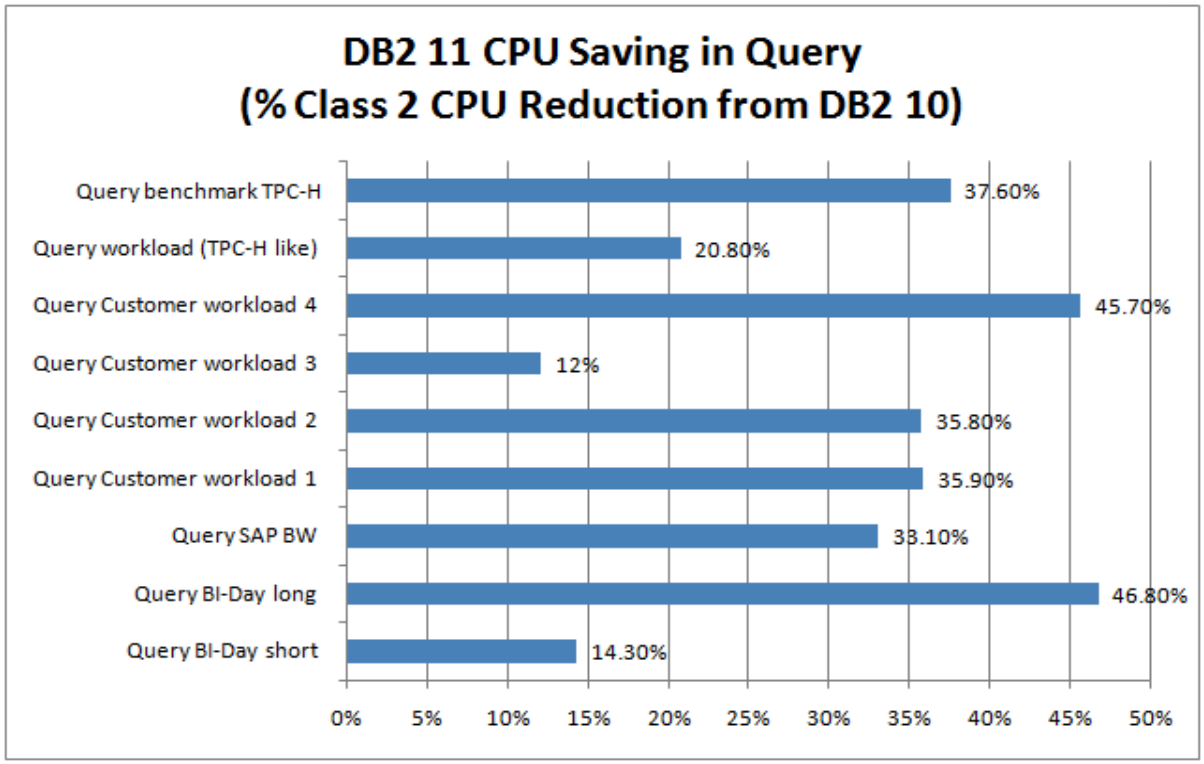
- IN-list and OR optimizations
- Multi IN-list matching and list PF
- Range-list access for OR predicates
- Simple outer join merge
- Prune simple always false predicates
- Pruning of redundant outer joins
  Stage 2 predicate pushdown
- More in-memory sorts
- Straw model Parallelism
- Dynamic record range
- Index INCLUDE columns

DB2 11

- Rewrite common stage 2 predicates to indexable
- Indexability for simple OR/IN
- Pruning compound true/false
- Push compound predicates to views/table expressions
- Generalize spidx/hash joins
- Index duplicate skipping
- Early-out join
- Optimized subquery cache
- Page range screening for join
- DPSI parallel join
- Optimizer externalization of missing statistics
- Not logged DGTTs

DB2 11 for z/OS Analytics Performance

- DB2 11 Internal Workload measurements
  - Best results of any prior DB2 release



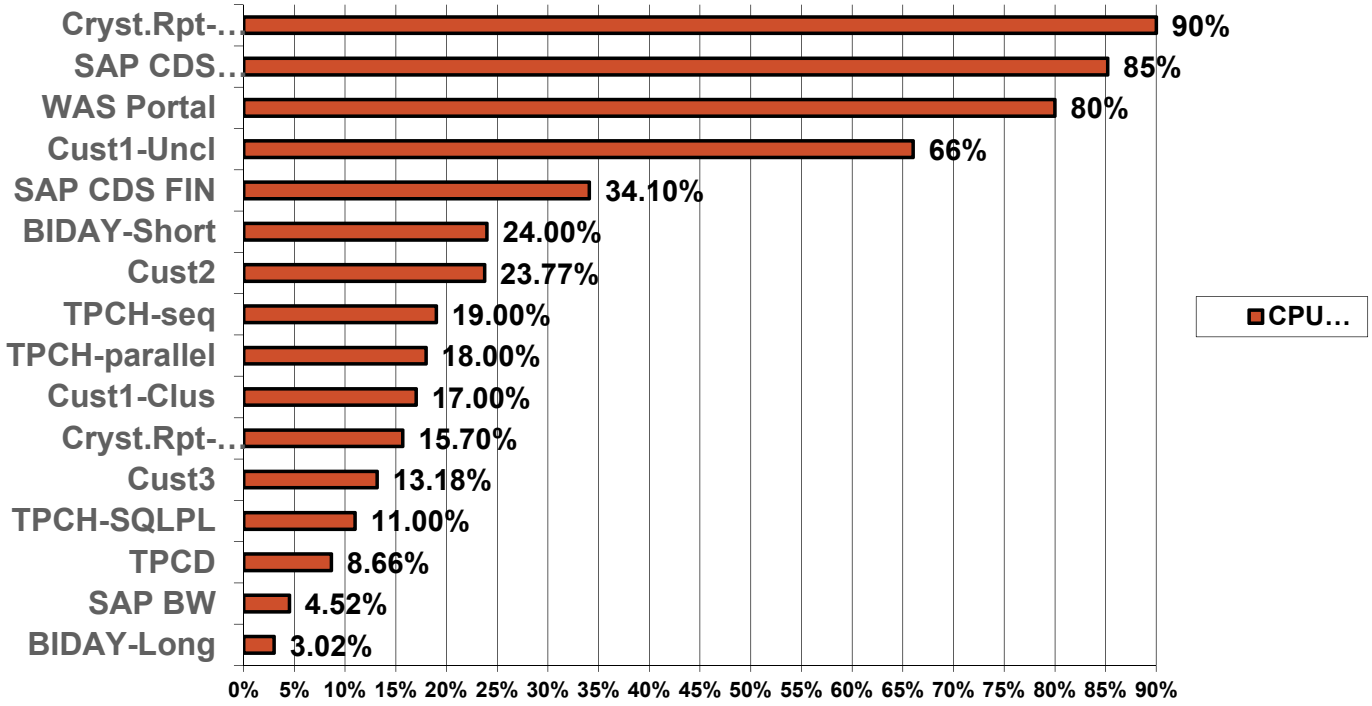
- NOTE: IBM results were obtained in isolated testing for internal measurement purposes only
  - Customer results cannot be predicted due to variability of workloads

DB2 12 (High Level) Performance

- Query focus based upon new(er) workloads
  - Complex views or table UDFs
    - UNION ALL
    - Outer joins
    - Join predicates with (stage 2) expressions
    - CASE expressions, CAST functions, scalar functions
  - Query - General Bottlenecks
    - Sort/workfile reductions
    - Reducing prepare cost and frequency
    - I/O performance
    - Reduce unnecessary prefetch scheduling
  - OLTP (engine) focus
    - Index tree traversal
    - Reduced getpage cost for pinned objects (PGSTEAL(NONE))
    - Reduced latch contentions, remove scalability limitations

DB2 12 for z/OS Analytics Performance

- Initial performance measurements showing promising results
  - 2-3 times acceleration for new analytics or poorly clustered workloads
  - Up to 25% CPU saving for traditional query workloads
  - Minor improvement for IDAA targeted (scan based) workloads



- NOTE: IBM results were obtained in isolated testing for internal measurement purposes only
  - Customer results cannot be predicted due to variability of workloads

Agenda

- Analytics – some perspective
- Analytics and DB2 for z/OS
- DB2 for z/OS Tuning Considerations for Analytics

Tuning for Analytics on DB2 for z/OS

- Strategic (traditional BI/DW) Analytics
  - If near-real time is sufficient (as it generally is)
    - Perfect candidate for DB2 z with IDAA
  - For real-time
    - DB2 z requires sufficient resources (CPU, memory, WF space/DASD)
    - Targeted tuning
      - Partitioning, clustering, indexing (out of scope for this presentation)
      - Potentially MQTs (although DB2 z does not support incremental update)
        - Must use “current” UNION ALL with MQT/history for real-time

- Tactical (transactional) Analytics
  - Since analytics is against the scope of the transaction
    - Existing OLTP indexing may suffice

Tuning for Analytics on DB2 for z/OS

- Many recent DB2 performance enhancements may require some degree of tuning to:
  - Ensure optimal performance
  - Minimize analytics impact on OLTP workloads

- Some configuration discussed
  - Sparse index
  - DPSIs
  - Parallelism
  - Compression
  - Sort/Workfile (incl RID overflow, DRDA impact)
  - Optimizer statistics recommendations

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)

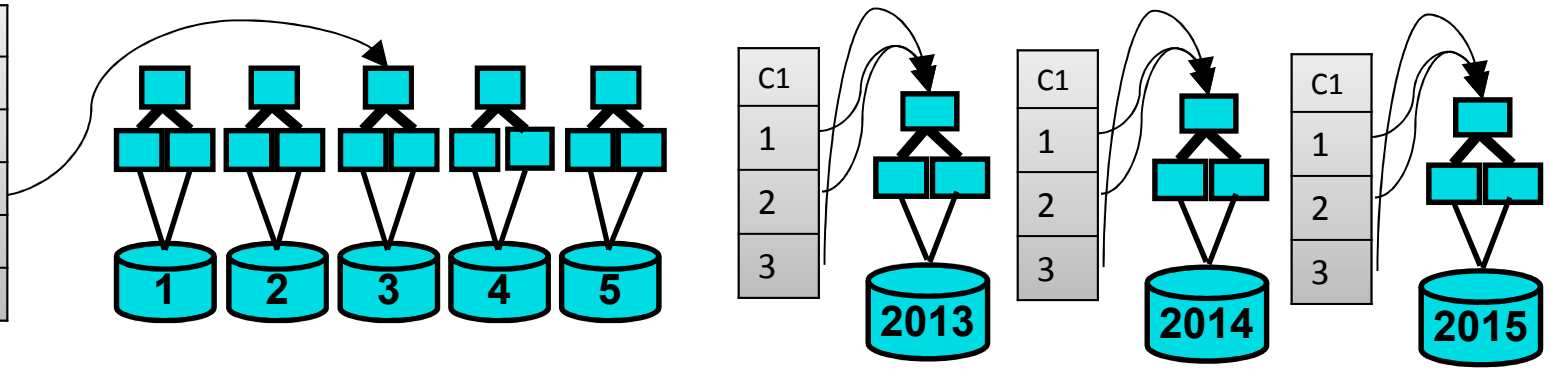
- Memory considerations for sparse index
  - Default DB2 setting for MXDTCACH is conservative
  - Customers generally undersize WF BP (compared to data BPs)
    - And often set VPSEQT too high (close to 100) for sort BP

  - 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

Recent DPSI performance enhancements

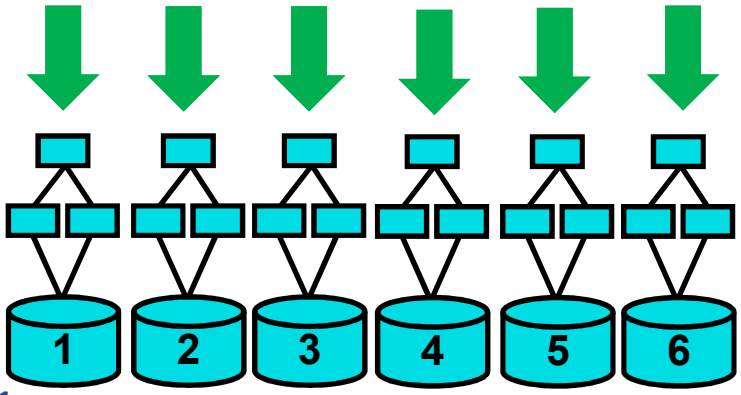
- A "partitioned" index means excellent utility performance
  - But historically there was one sweet spot ONLY for DPSIs
    - When local predicates in the query could limit partitions to be accessed
  - Outside of the DPSI sweet spot – performance often suffers compared with NPIs
- DB2 11 improves join performance for DPSIs
  - By page range screening for join predicates (when join by partitioning columns)
  - By exploiting parallelism when partitioned by non-join columns

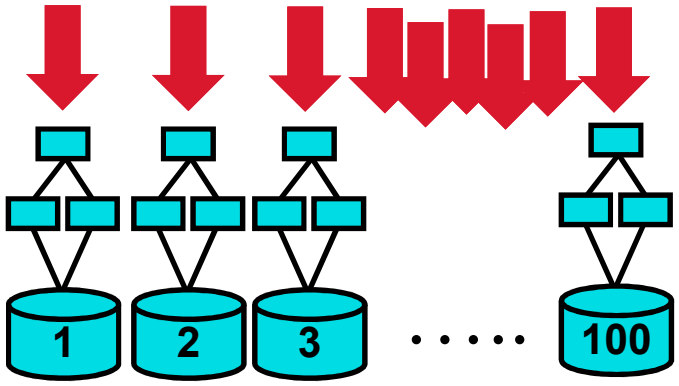
YEAR	PARTNO
2011	1
2012	2
2013	3
2014	4
2015	5



Parallelism to alleviate DPSI query performance

- Outside of sweet spot(s) - Parallelism can improve DPSI performance
  - You must have fewer (larger) partitions – rather than many smaller partitions
  - Think 12 partitions or less – rather than 100s
    - Parallelism cannot save DPSI performance with 100s of partitions
      - Unless your CPU resources can support 100s of parallel degrees
    - This may seem counterintuitive for utility performance
      - Except……if larger/fewer parts mean DPSIs rather than NPIs – net is utility improvement
    - DPSI part-level join parallelism controlled by zparm PARAMDEG_DPSI





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
    - Very conservative – set PARAMDEG=2 & PARA_EFF=100
- Parallelism requires sufficient resources
  - Specifically – zIIP processors
    - Since 80% of child tasks are zIIP eligible
    - Parallelism may increase CPU consumption up to 20%

22

Compression considerations

- DB2 11 enhances decompression
  - Including partial row decompression
- Most benefit achieved if compression results in I/O savings
  - For in-memory objects, compression adds CPU overhead
- What about index compression?
  - Instead – consider larger index leaf page sizes
  - Especially for duplicate indexes
    - Where FULLKEYCARDF < NLEAF

23

DRDA performance implications

- Isolation level has an impact on performance
  - Default isolation level is CS(2).
    - Dynamic SQL's will use SYS*200 packages for default isolation level(CS) .
    - Other isolations will use different packages (Eg:SYS*300 for RS)
    - In CLI, application can set isolation level using connection attribute, db2cli.ini or db2dsdriver.cfg file.

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.swg.im.dbclient.config.doc/doc/r0054653.html

    - Best performance (of course) is ISOLATION UR

- Disable default of CURSOR HOLD used by ODBC/JDBC
  - As WITH HOLD disables DB2 in-memory sorts
  - Can be changed by setting CURSORHOLD=0 in db2cli.ini file

Sort / Workfile Recommendations

- In-memory (from V9 to 11) is avoided if CURSOR WITH HOLD
  - See previous slide

- FETCH FIRST n ROWS
  - Recommended to use if less that FULL result set required
    - Better to tell DB2 you will only fetch a maximum of 500 rows
  - If FETCH FIRST value is < FULL result set
    - Will reduce sort workfile usage

Sort / Workfile Recommendations

- Ensure adequate WF BP, VPSEQT & datasets
  - Sort requirements can increase with Analytics
    - Goal is to minimize impact Analytics has on OLTP work
  - Set VPSEQT=90 for sort (due to sparse index, tag sort and/or DGTTs)
    - Evaluate sync I/Os in WF BP
      - may indicate sparse index spilling to WF
      - Or, usage of tag sort (row length > 4k)
  - Provide multiple physical workfiles placed on different DASD volumes
    - To avoid WF spacemap contention
      - 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-failover to WF

- DB2 10 added RID failover to WF
  - If a RID threshold was reached
    - query would write RIDs to workfile rather than reverting to tablespace scan
  - DB2 11 removed prior limitation for queries with column functions

- DB2 10 also increased default RID pool size from 8 – 400 MB.
  - Requires customer DBA/sysprog to update zparm job
  - Smaller RID pool are more likely to cause overflow to workfile
    - RID overflow to workfile can add up to 25% CPU compared with RID pool

  - Downside of overflow to workfile
    - If the optimizer mis-estimates the filtering, query may qualify 90% of table
      - Previously in V9 at 25% of table or RID limit (1/2 of RID pool)
        - » If single index list prefetch or multi-index OR – revert to tablespace scan
        - » For multi-index AND – terminate that leg
      - V10 and later
        - » Overflow to WF
  - Zparm MAXTEMPS_RID=NONE if failover to WF causes perf issues

DB2 Optimizer and the 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
  - What to collect?
    - May be less critical for simple OLTP queries
    - Becomes more important as statement complexity or number of objects increase
- DB2 11 added externalization of missing statistics by the optimizer

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

DB2 11 exploitation by OQWT

- Optimal performance on DB2 often requires
  - Good base of indexes applicable to the queries
  - Good base of statistics applicable to the queries
  - Ability of DBAs to resolve any performance issues

- Skill vs scalatility
  - Even with highly skilled SQL tuners, ability to choose targeted indexes & statistics for a workload is near impossible
    - Requires analysis of 100s or 1000s of SQLs

- OQWT exploitation of DB2 11 features
  - Selectivity overrides (Filter Factor hints)
  - Optimizer externalization of missing stats
  - New index advisor options (from DSN_VIRTUAL_INDEXES)

Terry Purcell

IBM Silicon Valley Lab

tpurcel@us.ibm.com

Please fill out your session evaluation before leaving!

Session: Z9

Title: Real Time Analytics Query Performance with DB2 for z/OS