How to do your own performance review

By David Beulke
DaveBeulke@cs.com
703 798-3283

David Beulke of Pragmatic Solutions, Inc.

- IBM Gold Consultant
- IDUG President 2003-2004
- Former TDWI instructor
- Best speaker at CMG conference
- Author of the V8 and V7 z/OS DB2 DBA certification test
- Author of the Business Intelligence certification test
- Columnist for DB2 Magazine
- Author of the Data Warehouse Performance Seminar
- Former Editor of the IDUG Solutions Journal
- Author of Syspedia - the repository for data element analytics
- Extensive experience in performance and design of large database and data warehouse systems
  - Working with DB2 on z/OS since V1.2
  - Working with DB2 on LUW since OS/2 Extended Edition
- DaveBeulke@cs.com or (703) 798-3283

Agenda

- Research, Process & Impact
  - Where to look?
  - What to look at?
  - What are the best practices?
- Areas of interest
  - Platforms
  - Architecture
  - Systems
  - Database
  - Applications

 DaveBeulke@cs.com

Platforms – Where is the bottleneck?

- Mainframe used as the source of the majority of data
  - Determine the transaction drivers
- UNIX used for the data warehouse
  - How many transactions are in the warehouse? Years?
- Windows Websphere clients
  - Distributed transactions come from how many servers?
  - What are the configurations of the servers?
- So many parameters - so little time
  - DB2, DB2 connect, CICS, UNIX, Windows and Websphere

 DaveBeulke@cs.com

Architecture – How it helps performance

- Get a global view of the architecture flow
  - Enterprise architecture model of overall data flow
  - Regulatory, disaster recovery and processing window requirements
- Include all systems, interfaces and processing partners
  - Discover and explore workload characteristics
  - Study the baseline statistics or create them
- Evaluate application volatility for improvements
  - Stable applications provide neither good/bad but stable statistics
  - Workload can cause hitting the knee in the performance curve

 DaveBeulke@cs.com Page 5

Architecture – Logical flow of processing and data?

- Transactions against system, DB2 and other processes
  - Response time sensitive transactions with many store procedures
- UNIX data warehouse I/O & Sort intensive
  - Websphere, B. O., and Ad Hoc OLAP users driving big requests
- Batch programs processing additional transactions
  - Nightly routines and processes filling orders
- Nightly or weekend utilities
  - Backups, more backups and reorgs of many different size tables

 DaveBeulke@cs.com Page 6

Architecture – Could the flows be improved?

- Reduce I/O activity or improve data flow
  - Free up system resources
  - 50%-90% CPU I/O performance improvement
- Eliminate or improve number of processes
  - Complete CPU cost savings
  - Reduce deadlocks
- Change direction or focus of processing
  - Combine processes,
  - parallelism of single or multiple programs
  - summarization or data

 DaveBeulke@cs.com Page 7

Systems – Which components must perform?

- System settings can dramatically effect performance
  - Caching or reuse of resources can determine performance
    - Security checking can be a huge overhead
    - Code table checking should be switched to memory tables
  - Connections should be reused as much as possible
  - Network bandwidth should be evaluated against workload
- Database priority determines resource allocations
  - Stored procedure naming conventions help track usage
  - Work load manager velocity settings determine resource slice
- Sort Pool is used by the majority of transactions
  - Hybrid joins, Multi and matching index access use the Sort pools

 DaveBeulke@cs.com Page 8

Systems – Where are the critical systems/processes?

- Inventory all the connected systems or key processes
  - Within all the workloads which programs execute the most
  - What functions are critical to the performance equation
- Isolate the workload for each system against its statistics
  - Relate the processing against the history
- Determine whether the process is efficient
  - Do the number of I/Os reflect an efficient process?
- Does the workload reflect the business?
  - Processed 500 orders and do 5,000,000 I/Os?

 DaveBeulke@cs.com Page 9

Systems – Is it system or process performance?

- Would isolation or caching benefit the situation?
  - Errors or locking issues can be resolved
  - Faster access to available data or settings
  - Total Memory R-O-T = buffer pools + dbheap + util_heap_sz + pkgcachesz + aslheapsz + locklist + approx 10% overhead
- Realign system priorities to the workload(s)
  - Interconnect processing dominates processors
  - Time of day system priorities aligned with settings
- Change the system attributes
  - Workload, program, authorization, velocity
  - How many different settings?

 DaveBeulke@cs.com Page 10

Database – System level performance attributes?

- Database interaction with the overall subsystem?
  - How does the workload manage memory or sorts?
    - Do particular tables dominate buffer pools
    - Do the index definitions eliminate sorts
- Physical database attributes assist performance
  - Parallelism and partitioning are aligned
    - FREEPAGE and PCTFREE are defined properly
  - Indexes are defined with appropriate columns in order
- Timely maintenance and archiving are done
  - Maintenance schedule considers workload physical settings
  - Clear tables

 DaveBeulke@cs.com Page 11

Database – Table and index performance attributes?

- Use tablespace partitioning to spread out the I/Os
  - Parallelism dramatically cuts processing elapsed time
  - UNION ALL Views are utilized to spread data out
- Table design minimizes the number of I/Os
  - Analyze tables inserted or accessed together
- Index column type and order determine performance
  - SQL column data type and column order effect performance
  - Column comparison methods effect performance and access type
- Referential integrity can effect performance
  - Optimizer uses RI for determining access path choices

 DaveBeulke@cs.com Page 12

Database – Processing performance attributes?

- Does the database processing happen orderly?
  - What are the number of errors/abends per daily process?
  - How many Sort Heap overflows?
- Process utilize the physical database definitions
  - Partitioning, parallelism, locking and referential integrity
  - Big block I/Os are utilized appropriately
- Processing patterns are consistent
  - Access is similar day to day with consistent workloads I/O numbers
- Research the backup and reorganization schedule
  - Do these schedules reflect the workload
  - PCTFREE & FREEPAGE are different for each table

 DaveBeulke@cs.com

Application – Defined with performance in mind

- Plan package usage can effect reuse and security
  - Workloads should be able to reuse authorization and connection
  - Connection pooling and Java DataSource are implemented
- Logic flow should be consistent among all access
  - Application should follow a specific pattern to maximize caching
  - Current Statistics support optimized SQL that's Explained/reviewed
- Join validation through select and where column usage
  - Verify the join criteria matches index column definitions
- Minimal Cursors should be defined in a process
  - Retrieving many rows per SQL for processing logic

 DaveBeulke@cs.com

Application – Performance is a priority

- Error checking should be done for ALL conditions
  - Not only SQL but all external module calls
- Errors should always be evaluated
  - Errors indicate programming, logic or data issues to research
- Order of application execution should help database
  - Help space management throughout your processing
- Have the input order match the index or clustering order
  - Improves the data caching within the database

 DaveBeulke@cs.com

Application – Standard procedures are followed

- Application deployment/development follows standards
  - How are changes verified that they are improvements?
- Business changes are communicated to applications
  - Processing is reflective of current business processing
- Application utilizes normal and exception processing
  - Optimized for majority of transactions
- Processing is key driven for performance
  - Ordered keys are used for every application process

 DaveBeulke@cs.com

 DaveBeulke@cs.com

Application – Performance is part of CM & QA

- Analyze only the “impact” programs
  - Which programs are most active? 80/20 rule
  - Lock escalations, I/O cleaners
- Focus on SQL access paths executed frequently
  - Push processing into the database engine through Joins
    - How many Joins, Hybird, Nested, Inner, Outer etc...are used?
  - Additional analysis of most executed application functions
- Focus on DB2 and internal SQL functions
  - Dramatic improvement by using SQL functions
  - OLAP functions provide tremendous improvement

Summary – Performance reviews can save $millions$

- Research
  - Determine programs that are biggest consumers
    - Determine if CPU or I/O is the problem 80% CPU, 45% I/O
  - Use supplied tools for system, database or program modifications?
    - Use Configuration Assistant, DB2 Memory Tracker, Design Advisor and SQL Explain - Be realistic on what can be changed
- Process
  - Gather base statistics before changing anything
  - Involve all business programs partners
- Impact
  - Determine performance difference against baseline
  - Calculate the ROI of your efforts