

Cost Control Methods for DB2 for z/OS Workloads

Mary Beth Jeffords

South Carolina Budget & Control Board, Division of State Information Technology

Session Code: A14

May 6th, 2011 9:15 AM – 10:15 AM | Platform: DB2 for z/OS





OBJECTIVES

- Understand the role DBAs now need to play in understanding and controlling costs in their organizations.
- Learn about how various software pricing options, hardware options, and LPAR configuration can have a huge impact on the cost of DB2 for z/OS applications and how to gain control.
- Learn about options that can minimize the cost for distributed workloads and things to watch out for when developing and implementing distributed applications using DB2 for z/OS as the database server.



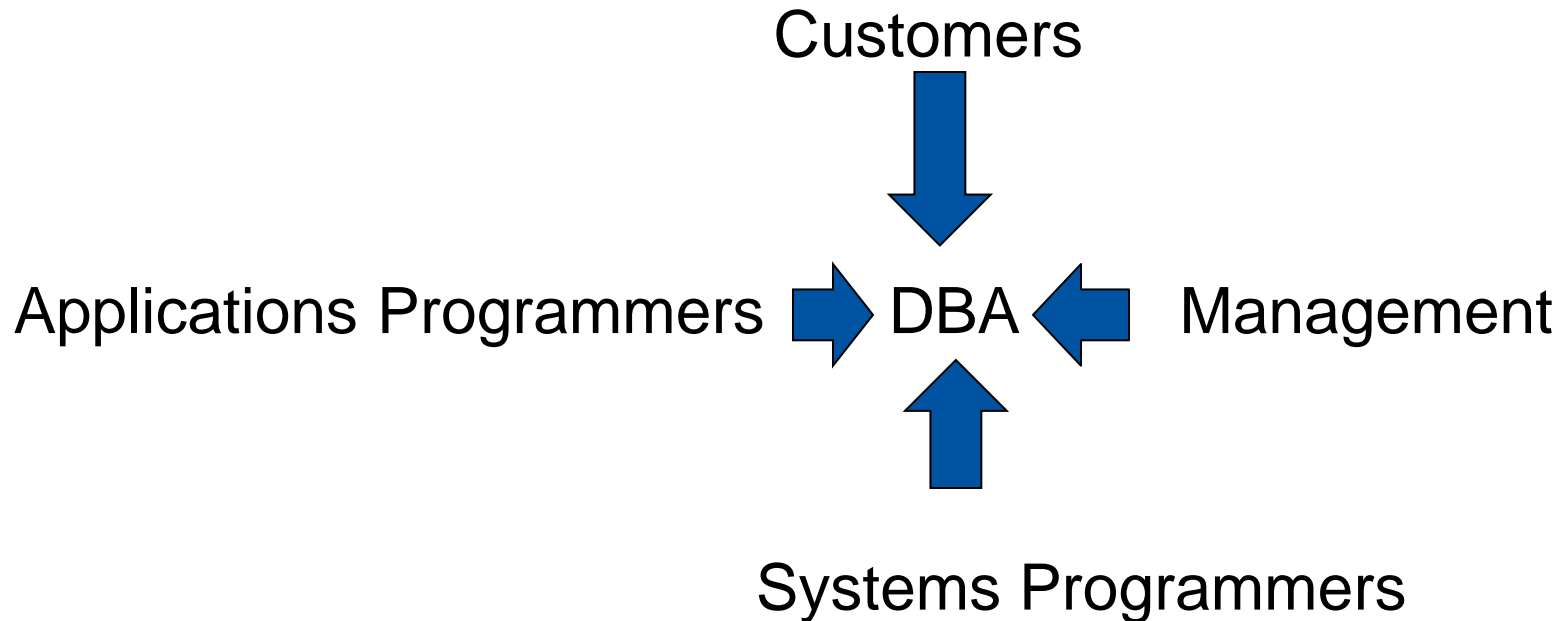
OBJECTIVES continued

- Learn about tools and techniques that can maximize performance and minimize resource utilization.
- Learn about user experiences with application cost estimation, platform migrations, DB2 subsystem cloning, IBM and third-party tools, and more.



DBA's Role in Cost Control

- The DBA is the main person that can control DB2 costs





Basic Cost Control Methodology for a DBA

1. Identify all costs associated with DB2 for z/OS workloads
– TCO (total cost of ownership)
2. Identify the ones you can control
3. Prioritize cost reduction projects
4. Ensure desired results are achieved



Basic Cost Control Methodology for a DBA

5. Continue monitoring, training, and research efforts
6. Carefully evaluate new hardware and/or software products before purchasing
 - Free trial evaluation period
 - In some cases, talking with technicians at other companies that are successfully using the technology is very helpful
 - This will usually also allow you to evaluate how responsive their technical support organization is, which is also a very important factor to consider



Chargeback Considerations

- If your shop provides IT services to outside customers and charges them, then you also need to know how they are charged
 - Charged by CPU utilization?
 - Charged by I/Os?
 - Charges for batch jobs, online transactions?
 - DASD space charges?
 - Billing package used?
 - SMF records and fields used: SMF 30, SMF 101, etc.
 - Rates charged (CPU rate, DB2 rate, etc.)
 - Method of rate determination



DB2 Related Costs

- Hardware
 - CPU, memory, I/O devices, DASD, tape drives, printers
 - Some hardware devices (such as tape drives and DASD) can actually be shared or used with both mainframe processors and servers.
- Software
 - DB2, z/OS, CICS, COBOL, utilities, third-party tools, scheduling software, monitoring software, etc.
- System Configuration
 - LPARs, shared DASD, SYSPLEX
- CPU Utilization ★



DB2 Related Costs Continued

- Backups / Utility Jobs (REORGs, RUNSTATS, etc.)
 - REORG TABLESPACE A0%.% RTS CONCURRENT
- Disaster Recovery
 - Immensely easier for mainframes vs. a whole bunch of different sorts of servers



DB2 Related Costs Continued

- Building / Floor Space
- Physical Security
 - Should be shared by Linux, Unix, Windows servers present in the building as well as the mainframes
- Electricity / UPS System
 - zSeries mainframes are considered “green technology”
- Administrative
 - DBAs probably don’t have much control over this, but it is a factor that must be included



DB2 Related Costs Continued / Human Resources

- Human Resources
 - Automate wherever possible, keep training current
 - Advantageous and cost-effective to maintain and retain deep technical skills
 - Need close interaction between IT customers, management, and those with current deep technical skills



Hardware

- Ensure all hardware is current and supported
- z9 and above – determine cost effectiveness of purchasing one or more zIIP and/or zAAP engines
 - zIIP – offloads distributed DB2 processing
 - Java, WebSphere, .NET/C#, DELPHI, etc. applications running on distributed platforms (Linux, Unix, Windows) utilizing ODBC and/or JDBC via DB2 Connect and/or direct JDBC type 4 connectivity
 - Current version of SAP now uses DB2 Connect and will utilize zIIP
 - V9 native SQL stored procedures (if called by DDF applications)
 - DB2 utilities (IBM and third-party)
 - Other DBMS software (CA-IDMS Release 17 and above)
 - Beyond z10



Hardware / zIIP

- If have significant amount of processing which can be offloaded to zIIP, send SMF data to vendor to determine cost effectiveness of purchasing one or more zIIPs
- Lowers cost of hardware as the amount of MSUs needed on central processor will be lower
 - DSIT went from IBM z9 2096-S03 / 617 MIPs / 85 MSUs to IBM z10 2098-P03 with 572 MIPs / 71 MSUs
- No software cost associated with zIIP
- Decreased MSUs/MIPs results in lower software costs as most mainframe software pricing is based on either MSUs or MIPs



Hardware / zIIP continued

- zIIP is a faster, cheaper processor
- Be aware that there is some overhead on the main CP for zIIP offload (DB2 uses enclave SRB for this)
- Be aware of chargeback implications of zIIP usage (SMF 30 for batch jobs, SMF 101 accounting trace records for DB2 applications, SMF records cut by other DBMSs)
 - The normalized zIIP time plus the main CPU time will generally be higher than running without zIIP because of the overhead
 - zIIP is cheaper so CPU on zIIP should not be charged at the same rate, else charges can actually increase because of the overhead on the main CP
 - Check with billing package vendors and verify formulas used to compute total CPU time used for chargeback and make appropriate adjustments



Hardware / zAAP

- zAAP offloads Java application processing for Java applications running on z/OS Unix System Services
- If you run a significant amount of Java on z/OS, then also evaluate potential cost effectiveness of zAAP
 - Note that CA-MSM uses Java on z/OS USS
- Can now do zAAP processing with just a zIIP engine
 - Automatic with z/OS 1.11
 - PTFs needed for versions of z/OS prior to 1.11



Hardware / Memory, I/O, DASD

- Memory
 - Paging
 - Buffer Pool Tuning
- I/O
 - Buffer Pool Tuning (access pattern determines pool assignment, ensure application DBAs note current assignment before DROP/CREATE)
 - I/Os per second – minimize
 - Page Sizes (GETPAGE is CPU intensive, use larger sizes for sequential)
- DASD
 - VARCHAR
 - Compression
 - Archiving Strategy
 - Clean up



Software

- Stay current and supported for all software products!
- Evaluate all pricing options (MIPs, MSUs, measured usage, etc.)
- Initial licensing cost, maintenance cost, upgrade fees?
- Be involved with contract renewals
- Get rid of obsolete software
 - If unsure about current usage and you are unable to get a definitive answer from the user community and no response from notices of discontinuance, rename a library and see if anyone screams. Run this way for at least a year before actually removing the software.
 - DSIT was able to get rid of several products that were no longer needed, including QMF.



Software continued

- If significant upgrade fees, either replace the product with a less expensive equivalent or isolate the software on a separate box (sometimes two mainframes are cheaper than one)
- Work with your vendors, some may cut you a break



Software / Measured Usage Licensing

- IBM products / Measured Usage Licensing
 - DB2
 - z/OS
 - CICS
 - COBOL
- For DB2 measured usage, check zPARM SMF89



Software / Measured Usage Licensing Continued

- Measured according to average peak MSU utilization per LPAR (entire LPAR utilization, not just utilization of the particular product).
- 3 MSU minimum charge



Software / zNALC Pricing

- Originally known as z/OS.e
- IBM offering that offers reduced software licensing costs
- A separate LPAR must be set up to run zNALC-qualified applications
- These applications must not utilize any COBOL or CICS
- DB2 subsystems on this LPAR act only as data servers for distributed workloads
- These applications must be qualified by IBM for the zNALC pricing



Software / zNALC Qualifications

- Some types of applications automatically qualify –
 - Packaged Vendor Applications (SAP, etc.) that run on a distributed platform (LUW)
 - Java / WebSphere Applications
 - Other distributed applications that formerly qualified on z/OS.e
- Others must be verified with IBM –
 - Any distributed application that involves business intelligence should qualify
- Offers more of an incentive for your customers to replace the CICS/COBOL with more modern solutions



LPAR Configuration - zNALC

 If you have significant zNALC qualified applications, set up a separate LPAR for zNALC pricing

- Limited to 8 TSO sessions, no COBOL, no CICS
- Run only DB2 for z/OS as a database server and related supporting software (monitoring tools, utilities, backups, etc.)
- Can use subsystem cloning procedure to split DB2 subsystems shared by both distributed and traditional applications
 - SAP R/3 Homogeneous System Copy Release 4.6C SR2
 - IBM SAP Homogeneous System Copy manual
 - DB2 for z/OS Administration Guide Volume 1, Chapter “Altering Your Database Design”, Section “Changing the High Level Qualifier for DB2 Data Sets”



LPAR Configuration – zNALC continued

- DB2 subsystems already running only as data servers for zNALC-qualified distributed applications (for us, SAP) can simply be moved to the zNALC LPAR



LPAR Configuration / Measured Usage Licensing

- ★ If using measured usage licensing and you have significant products other than DB2 running on the same LPAR(s), then evaluate setting up a separate LPAR for DB2 and connecting CICS regions, plus all the resources needed by the applications using those regions.
- This can result in a huge cost savings (10K per month for DSIT).
- Considerations for multiple LPARs are tape drive balancing, shared DASD, global resource serialization.



LPAR Configuration - Example

- One box – IBM z/10 2098-P03 w/ 4 LPARs
 - BCBTEST -Test LPAR
 - PRODDB2 - Production DB2 LPAR with CICS/COBOL
 - DB2, CICS, COBOL, Batch application programs, Batch utility jobs
 - Performance Monitoring Tools
 - Print-related software
 - ZOSEDDB2 - Production DB2 LPAR for zNALC-qualified applications
 - Data-serving only DB2 subsystems (includes SAP subsystems)
 - Batch utility jobs
 - Performance monitoring tools
 - BCBPROD - Production LPAR for non-DB2 workloads (original LPAR)
 - Non-IBM DBMS systems (CA-IDMS central versions with IDMS/DC)
 - Old CICS regions, VSAM/flat file application systems, print, supporting software, CA-DOCVIEW, CA7 (can route jobs to other LPARs), etc.



LPAR Configuration – Other Cost-Saving Possibilities

Software that does soft capping of the peak CPU utilization on an LPAR, reducing measured usage licensing costs



CPU Utilization

- SQL
- Application Design
- Database Design
- DB2 Utilities
- DB2 Subsystem Configuration / Tuning
- z/OS Configuration / Tuning



CPU Utilization / SQL

- SQL coding and tuning is a largely controllable factor affecting cost.
- Ensure application programmers/DBAs are trained in how to code SQL for high performance. (On-site training classes may be the best value for a large number of students).
- Ensure application programmers/DBAs know how to EXPLAIN and tune SQL via the IBM-provided free tools (OSC/Data Studio) and/or other tools you may have available at your site.



CPU Utilization / SQL

- Utilize RLF (Resource Limit Facility) to control run-away queries.
- Utilize plan stability feature available in V9 for static SQL.
- Explain and save good access paths for dynamic SQL.



CPU Utilization / SQL

- REORG, RUNSTATS
 - Automated set-up previously discussed that processes all objects, skipping objects for which no significant change has occurred since last run (according to real-time stats)
 - Various third-party utilities may also be using alternative processing methods that further reduce CPU utilization
- Also run all SQL statements through the OSC/Data Studio Statistics Advisor, collecting recommended additional statistics and adding those statements to your RUNSTATS jobs
- Regularly rebind packages/plans for static SQL (no fear with plan stability)



CPU Utilization / SQL

- ★ It is absolutely essential to be able to immediately begin tracing all SQL (both dynamic and static) for a particular workload, consolidate like SQL statements, and be able to quickly pinpoint the high-cost SQL statements in terms of total CPU usage and frequency of execution.
- Other relevant statistics such as average CPU, total and average GETPAGES, etc. should also be provided.
- The tool/method used should also do an EXPLAIN and save the access path.
- The high-cost statements found are the ones for which it is worthwhile to spend further tuning efforts on.



CPU Utilization / SQL Emergency Tuning

- It is possible for a seemingly innocuous SQL statement that starts performing only a little badly (still runs too fast to catch it on real-time monitor screens) to eat up your CPU if hundreds of simultaneous executions of the statement are being processed.
- After finding the SQL statement causing the performance degradation (using the aforementioned absolutely essential tool) see if you have the old access path available and return to that if possible. This is usually not possible with dynamic SQL unless you contact IBM.



CPU Utilization / SQL Emergency Tuning continued

- If neither additional statistics collection nor a coding improvement fail to remedy the situation and it is not possible to return to the former access path, open a PMR (severity 1 if degradation is severe enough) with IBM and send the relevant info (SQL statement , DDL, and statistics) via the OSC/Data Studio “Service SQL”/”Capture Query Environment” facility. So far in my experience, IBM has been able to provide a quick solution in this situation.



CPU Utilization / SQL Emergency Tuning continued

- If it is possible to return to the good access path, consider this a temporary relief measure and try to find a permanent solution (coding change, index change, improved statistics, IBM APAR, ZPARM change, etc.) that will allow the DB2 optimizer to choose an efficient access path again on its own. Use optimization hints and the like only as a last resort.



CPU Utilization / Performance Monitoring and Tuning

- It is also essential to use the absolutely essential tool to trace SQL workloads for new/changed applications to ensure that high cost statements are found and tuned.
 - This lessens the risk of a problem with the new application/change from the start and ensures SQL statements and access paths have been captured.
 - The same tuning tools and methods as is used for emergencies can also be used in non-emergencies.
 - There is also a productivity benefit in that SQL tuning effort is only expended on the true high cost statements that need it.
 - Also do this before and after DB2 version upgrades for existing workloads.



CPU Utilization / Performance Monitoring and Tuning

- It is also absolutely essential to have a real-time monitoring tool that can also produce statistics reports.
- You should save at least the daily summary data from the SMF 101 records cut by turning on the DB2 Class 1 accounting trace.
- Most performance monitors provide a batch component for loading this SMF data into a set of DB2 tables for historical tracking and inquiry purposes.
- We produce spreadsheets and charts using this data so that we can track CPU usage for customer applications.
- This also helps us ensure our Billing package is in line.



CPU Utilization / Application and Database Design

- Ensure application programmers/analysts and DBAs are trained appropriately in good design techniques.
- IDUG and other conferences are a great source of cost-effective education on innovative designs to meet all sorts of needs, including high performance.



CPU Utilization / Application and Database Design / System-Level Tuning

- Application/Database Design
 - Historical and/or older less frequently accessed data
 - Reduce the number of those CPU-intensive GETPAGEs
- DB2 Subsystem Tuning
 - Must have real-time performance monitor
 - Training, documentation, testing, etc.
- z/OS and Network Tuning
 - Utilize OS-level performance monitor
 - Training, documentation, testing, etc.
- All groups should be working together to ensure optimal configuration for your workload



Tips for Distributed Applications

- Distributed applications are extremely advantageous as they can take the most advantage of zIIP and zNALC pricing
- Takes advantage of the latest and greatest development tools without sacrificing the core strengths of the z/OS environment
- No need to fear DDF (WLM can prioritize, dynamic SQL is controllable, can also utilize Stored Procedures)
- With all the new pricing options and technological innovations, there is no longer any reason that the data serving portion should be more expensive on z/OS than LUW
- Redbook: “Distributed Functions of DB2 for z/OS and OS/390”
 - Latest is “DB2 9 for z/OS: Distributed Functions”
 - Section 3.3 “Workload Manager Setup”
 - Section 5.9 “Remote Application Recommendations”



Tips for Distributed Applications

- WebSphere Java applications can now connect directly via type 4 JDBC driver without going through DB2 Connect server. A DB2 Connect license is still required.
- We have also had customers develop distributed applications using the Microsoft .NET development framework and the C# language, and also DELPHI.
- Younger generation programmers can utilize their favorite Windows-based tools using DB2 for z/OS as a data server and never have to use TSO.



Distributed Applications / Things to Watch Out For

- Utilize all practical forms of connection pooling available
- Beware of held cursors past commit points - effectively disables all connection pooling
- Beware of lack of COMMITs after read-only transactions (including UR isolation level)
- Note that MAXDBAT + CTHREAD is limited to 2000
 - This limit is vastly increased in DB2 10 (20000) and DB2 10 also offers “high performance DBATs”
- If large number of concurrently held cursors and no COMMITs, you may see hundreds of DBATs on your monitor screens and won't be able to easily tell which are active and which are idle
- Idle active threads also prevent use of WLM response time goals



Distributed Applications / Things to Watch Out For

- Beware of AUTOCOMMIT. It's better for commit processing to be controlled by the application at the proper time.
- Beware of excessive COMMITs as COMMITs are expensive.
- Rule of thumb: If it was bad practice in “traditional applications”, then it's probably still bad practice for any type of application.



Distributed Applications / Related ZPARMS

- CMTSTAT=INACTIVE
- MAXDBAT=750 (choose value appropriate for you)
- CONDBAT=2300 (choose value appropriate for you)



Distributed Applications / Related ZPARMS continued

- IDTHTION=2100 (choose value appropriate for apps)
 - Terminates idle active threads/DBATs
- POOLINAC=120
 - Gets rid of unused inactive pooled DBATs
- CACHEDYN=YES
- ACCUMACC=5000 (choose setting appropriate for you)
 - Cuts way down on volume of SMF 101 records
- ACCUMUID=0 (choose setting appropriate for you)



Distributed Applications / DB2 Connect

- DB2 Connect Client Configuration Keywords
 - CURSORHOLD=0
 - AUTOCOMMIT = 0
- DB2 Connect Maintenance / Upgrades
 - Note that fixes requiring additional actions are not marked (such as addition of a patch1 or patch2 keyword to implement the fix)
- No SMP/E and/or HOLDDATA equivalent for LUW platform



Application Cost Estimation

- We have been able to utilize the absolutely essential tool that traces the SQL in a workload to estimate the cost of an application's DB2 for z/OS usage
- We know our DB2 rate per hundredth CPU second
- We are able to see the average CPU of the SQL in the workload and multiply that by our rate multiplied by the expected number of SQL statements executed per day for rough estimates
- We can also easily determine accurate DB2 cost for particular statements



Platform Migration Experiences / z/OS to LUW

- We successfully assisted with the porting of a DB2 for z/OS V8 database for a large-scale distributed application (written in DELPHI, running on Windows Citrix server farm) to an equivalent DB2 for LUW V9 database running on a mainframe-sized pSeries (9133 Model 55A running AIX)
- No significant application coding changes were necessary, code can be tested on a Windows machine running its own copy of DB2 for LUW
- If a third-party is enlisted to perform such a migration, make sure they are aware of all known operating characteristics of the application (number of concurrently active DBATs, GETPAGES/I/O's per second, CPU utilization, buffer configuration, etc.) on the current platform
- Application data serving runs fine on either platform if appropriate features and cost control methods are utilized



Platform Migration Experiences / LUW to z/OS

- Tested porting a DB2 LUW V8 database for an application developed with Java and Websphere to a DB2 for z/OS V8 database
- Used DB2 Estimator for Space Calculations
- Scalar UDFs using SQL procedure language converted to equivalent stored procedures
 - Note: DB2 for z/OS V10 now supports this
- Data Studio (the free version) was used extensively. Was able to simultaneously connect to both LUW and z/OS databases to work with the DDL
- Used Microsoft Access to transfer the data directly from LUW to z/OS tables



Platform Migration Experiences / LUW to z/OS

- Used our absolutely essential SQL capturing tool to trace all the SQL during testing by the application developers
- Used Data Studio to compare response times of certain SQL statements between the two platforms. This worked great!
- Used OSC and LUW EXPLAIN tools to check poorly performing SQL with OSC Statistics Advisor and to analyze optimizer/access path differences between LUW and z/OS
- IBM does not document optimizer differences between DB2 for z/OS and DB2 for LUW
- z/OS does not currently push down predicates into views containing outer joins
- Eliminated above problem by coding all the SQL directly with manually pushed down predicates instead of using these views
- Absolutely essential SQL capturing tool was also able to reproduce some lost SQL source for the developer
- Was able to roughly estimate cost for application's data serving on z/OS using the info from the absolutely essential SQL capturing tool, the statistics from the application running on DB2 LUW, and our DB2 rate



Summary

- DBAs are uniquely positioned to more actively participate in cost-related decisions in their organizations
- Technological advances and new pricing options on zSeries servers have made it into an attractive alternative platform for data serving workloads for distributed, WEB-based, and e-business applications
- Many tools and techniques are available to assist DBAs with keeping DB2 for z/OS-related costs under control
- Don't forget the importance of continued research to stay abreast of future cost-related changes, in order to stay knowledgeable and keep your IT infrastructures as cost-effective as possible
- Questions???

Mary Beth Jeffords

South Carolina Budget & Control
Board, DSIT

turner@cio.sc.gov

Session: A14

Cost Control Methods for DB2 for z/OS
Workloads

