Hard Lessons Learned From Customer Health Check Studies

John Campbell
IBM DB2 for z/OS Development

Session Code: B10
10th November 2010 at 10:00am  |  Platform: DB2 for z/OS
Objectives

- Introduce and discuss the most common issues
- Share the experience from customer health check studies
- Share the experience from live customer production incidents
- Provide proven recommended best practice
- Encourage proactive behaviour
Agenda

• WLM Policy Setup
• Balanced Design across CICS and DB2
• Disaster Recovery
• Continuous Availability vs. Fast Failover
• Preventative Software Maintenance
• Multiple Data Sharing Groups
• Running Multiple DB2 Subsystems on same LPAR
• Normal DB2 Warm Restart vs. Restart Light
• Storage Tuning
• Summary
WLM Policy Setup

• Common Problems
  • WLM dispatcher can starve a thread when WLM services classes are poorly managed
    • Batch and other application workloads running as discretionary or with very low importance
    • DB2 application workloads running at or above the priority of DB2 system ASIDs
    • No clear distinction between online transaction and batch workloads
    • DIST ASID (DDF) running behind application enclaves
    • CICS and IMS/TM transaction response goals not practically achievable
    • IRLM address space not running in service class SYSSTC
  • If a thread is starved while holding a DB2 latch or other vital resource then the subsystem or data sharing group grinds to a halt
WLM Policy Setup ...

- **Recommendations**
  - The subsystems/address spaces VTAM, IRLM, and RRS should be mapped to the service class SYSSTC
  - Prior to V9, the DB2 system address spaces (DBM1, MSTR, DIST, SPAS) and the WLM-managed stored procedure AS should be mapped to a user-defined service class which has an Importance 1 and a very high velocity goal (e.g. 85)
    - Do not recommend putting DBM1 and MSTR AS into SYSSTC because of risk of DB2 misclassifying incoming work
  - However with V9, move MSTR into service class SYSSTC
    - DB2 Service Monitor is built into and operates within MSTR
    - DB2 Service Monitor must be running higher than the AS it monitors
    - DB2 Service Monitor will be impacted if MSTR is pre-empted
  - If running at very high CPU utilisation with possibility of CPU starvation
    - Move all the DB2 system address spaces (DBM1, MSTR, DIST, SPAS) to service class SYSSTC
WLM Policy Setup ...

• Recommendations ...
  • No DB2 application workload should run higher than the DB2 system ASIDs
  • No DB2 application workload should run as discretionary or with very low importance
  • For CICS and IMS, use 80th or 90th percentile response time goals as transactions are typically non-uniform
    • The transaction response time goals must be practically achievable
    • Use RMF Workload Activity Report to validate
-DIS THREAD(*) SERVICE(WAIT)

- Identifies allied agents and distributed DBATs that have been suspended for more than 60 seconds or twice the IRLM resource timeout interval which ever is the greater
- If the thread is suspended due to IRLM resource contention or DB2 latch contention, additional information is displayed to help identify the problem
- Will also attempt to dynamically boost priority (via WLM services) for any latch holder that appears to be stuck
- Can run with SCOPE(GROUP) with output piped back to the originating member
- Recommendations
  - V8 – Strongly recommend to run at one-minute interval through automation
  - V9 – Driven at one-minute interval by the internal DB2 System Monitor
WLM Blocked Workload Support

- Introduced on z/OS V1.9 (rolled back to 1.8 and 1.7 via APAR OA17735)
- Objective: Allow small amounts of CPU to be allocated to workloads that are CPU starved when system is running at 100% utilisation
- Controlled by two new parameters in IEAOPTxx parmlib member
  - BLWLINEHD – Specifies the threshold time interval for which a blocked address space or enclave must wait before being considered for promotion
    - Default value (after OA22443) is 20 seconds
  - BLWLTRPCT – Specifies how much of the CPU capacity is to be used to promote blocked workloads
    - Default value (after OA22443) is 5 (i.e. 0.5%)
  - For more information, see APARs OA17735, OA22443, and tech doc
- Recommendation: All customers should run with this function on
Balanced Design across CICS and DB2

- Common Problems
  - Imbalance across configuration of CICS and DB2, lack of fixed topology across CICS and DB2, and CICS transaction affinities
    - Sporadic unpredictable response times
    - Complex debug performance issues
    - Single points of failure
    - Increased risk of DB2 crashes
  - Use of Dynamic Plan Selection driving up CPU resource consumption
  - Over-use of protected ENTRY threads with RELEASE(DEALLOCATE)
  - Misuse of TCB Priority on DB2ENTRY
  - Missing performance improvement opportunity because of limited use of THREADSAFE
  - Lack of strategy for the resolution of in-doubt threads
  - Excessive number of collections on PKLIST of PLANs
Balanced Design across CICS and DB2...

- **Recommendations**
  - Do not use the Group Attach to improve
    - In-doubt thread resolution since CICS–DB2 pairing is maintained
    - DBM1 31-bit virtual storage planning since the number of threads coming into DB2 subsystem are a known quantity
  - Automated recovery from LPAR failure
  - Load balancing due to better CICS-DB2 thread reuse
Balanced Design across CICS and DB2 ...

- Recommendations ...
  - Implement fixed landscape topology
    - Configure number of CICS AORs based on LPAR size
      - In non-THREADSAFE environment configure one CICS per defined CP (processor) since CICS QR TCB is bottlenecked on a single processor
    - Configure ARM policy so that CICS+DB2 are allowed to restart as a unit
      - Allow for in-doubt URs to be automatically processed and resolved
      - Data integrity maintained
    - Configure DB2 CTHREAD value based on static CICS environment to allow for better balancing of DB2 threads and better DBM1 31-bit virtual storage usage
Balanced Design across CICS and DB2 ...

- Recommendations ...
  - Thread reuse
    - Configure protected ENTRY threads based on arrival rate and not on the basis of a binary decision
    - Number of protected ENTRY threads should be balanced against the affordable CTHREAD budget of the DB2 subsystem
      - Floating CICS AORs across LPARs can cause unpredictable queuing of threads due to protected ENTRY threads not having timed out
      - Can cause unpredictable response times and wildly varying response times which can cause severe z/OS Sysplex wide issues
    - Reuse of unprotected ENTRY (and POOL) threads using natural arrival rate can provide a better alternative
      - Removes queuing for threads due to reaching CTHREAD
      - Achieve more even transaction response times
      - Lower CPU cost per transaction at higher CPU utilization
      - Achieve better use of DBM1 ASID 31-bit virtual storage due to threads going through de-allocation on a regular basis
Balanced Design across CICS and DB2 ...

- Recommendations ...
  - TCB Priority (PRIORITY on DB2ENTRY)
    - HIGH should always be the preferred configuration default
      - In a non-THREADSAFE environment the main CICS QR TCB will go to sleep if the CICS QR TCB is defined higher than the DB2 thread TCBs
      - Sleep/wake cycle can increase CPU cost by up to 30%
  - THREADSAFE
    - Can deliver much better CPU performance – up to 22%
    - Eliminates switching between CICS QR TCB and DB2 thread TCBs
- In-doubt Units of Recovery (UR)
  - In-doubts URs should be resolved automatically by resync with CICS, otherwise data integrity issues will result
  - Simplify the environment by using direct DB2 member (SSID) attach as the preferred mechanism to connect from CICS to DB2
Balanced Design across CICS and DB2 ...

• Recommendations ...
  • Number of collections on PKLIST for DB2 plans
    • Reduce the number of collections on PKLIST – even just one
    • Ruthlessly prioritize the order of the collections on PKLIST to reflect frequency of access
  • Use of single plan for CICS would allow for maximum thread reuse in the POOL
Disaster Recovery

• Problems
  • Lack of IT understanding of the real business requirements and expectations
  • Lack of transparency about what IT can actually deliver
  • Impact of extended distance when using disk subsystem based synchronous replication
  • Consistency of the remote copy with disk subsystem based synchronous replication (Metro Mirror/PPRC) is not guaranteed
  • DB2 warm restart off an inconsistent mirror will guarantee inconsistent data which will be very painful to fix up later
  • Forgetting to delete all CF structures owned by the data sharing group will guarantee logical data corruption
  • FREEZE and GO policy in GDPS/PPRC will lead to some degree of data loss (RPO>0)
  • Lack of recognition that disk subsystem based asynchronous replication (Global Mirror/XRC) will lead to some data loss
  • When using Global Copy the secondary copy is a fuzzy copy that is not consistent
  • Use of FlashCopy without ensuring consistency will lead to logical data corruption
Disaster Recovery ...

• Recommendations
  • Data consistency with Metro Mirror (PPRC) is of paramount importance
    • Use of Consistency Group (CGROUP) function combined with external automation like GDPS is mandatory to guarantee data consistency at the remote site for both planned outage and failure scenarios
    • Any sign of inconsistency found in your testing should be driven to root cause
      • Broken pages, data vs. index mismatches, etc.
    • DB2 warm restart re-establishes data consistency through restart recovery mechanisms
    • DB2 cold start, or any form of conditional restart, will lead to data corruption and loss of data
  • To eliminate DASD as single point of failure need to complement Metro Mirror (PPRC) with a non-disruptive failover HyperSwap capability,
    • GDPS/PPRC HyperSwap Manager
    • Basic HyperSwap in TotalStorage Productivity Center for Replication (TPC-R)
Disaster Recovery ...

• Recommendations ...
  
  • Use of FREEZE and STOP policy in GDPS/PPRC will avoid data loss (RPO=0)
    
      • But you have to accept a possible impact on continuous availability of running production system at the primary site
        
          • Systems could now be stopped for a reason other than a real disaster (e.g., broken remote copy link rather than a fire in the computer room)

  • Combine Global Copy with use of DB2 –SET LOG SUSPEND command
    
      • Pauses all updates to the primaries and allow the updates to drain to the secondaries so that a consistent point-in-time copy is created
      • Should only be used for creating a point-in-time copy or migrating data away
      • For Disaster Recovery, use Global Mirror solution which combines:
        
          • Global Copy
          • FlashCopy Consistency Groups
Disaster Recovery ...

- **Recommendations** ...
  - When using FlashCopy use DB2 -SET LOG SUSPEND command to temporarily freeze all DB2 update activity
    - Ensures the PIT copy is a valid base for recovery
    - Externalises any unwritten log buffers to active log datasets
    - Need to go through DB2 warm restart recovery to re-establish data consistency
    - For Data Sharing, you must issue the command on each data sharing member and receive DSNJ372I before you begin the FlashCopy
  - Test in anger and do not simulate
    - Continually validate recovery procedures to maintain readiness
    - Verify that RPO/RTO objectives are being met
  - Do not throw away your standard DB2 log-based recovery procedures
    - Even though it should be a very rare occurrence, it is not wise to start planning for mass recovery when the failure actually occurs
Continuous Availability vs. Fast Failover

- Design point for DB2 data sharing was continuous availability with a 'pay as go system' for performance as inter-system read-write interest comes and goes
- Two key ingredients of continuous availability
  - Active-active read-write sharing
  - Fine grained, dynamic transaction routing (DTR)
- Cost of data sharing can be aggravated by
  - System configuration
  - Extended Distance
  - Application locking ‘choke’ points
- There is a trade off between cost of data sharing vs. continuous availability
Continuous Availability vs. Fast Failover ...

- If affinity routing of application workload is successful in reducing inter system read write interest and the associated global locking and GBP dependent protocols, then it aggravates
  - Availability
    - As retained x mode pageset p-locks are blocking and cannot be negotiated away until end of the forward log recovery during DB2 warm restart of the failed DB2 member
  - Potential scalability issues
    - If an application workload is bound to a single DB2 member (virtual storage, latch contention)
Continuous Availability vs. Fast Failover ...

- Frequent customer issues and combination thereof
  - Lack of IT understanding of the real business requirements and expectations
  - Lack of transparency about what IT can actually deliver
  - 2-way active-passive failover model
  - System and transaction affinities
  - Critical business applications not sysplex enabled
  - Static workload routing and/or turning off workload distribution
  - Multi-site data sharing over distance
  - Lack of ‘white space’ for workload failover and N-1 availability
  - Occasional active-active read-write data sharing and not prepared for it
  - Lack of transparency about the difference between fast failover solution for high availability vs. true continuous availability
  - Single solution for continuous availability and disaster recovery
  - Parallel Sysplex and DB2 data sharing in a single CPC (box)
  - No failure isolated CF for LOCK1 and SCA structures
  - Not understanding the performance impact of system managed duplexing of LOCK1 and SCA structures
Continuous Availability vs. Fast Failover ...

- Recommendations
  - Understand the true business requirement and determine what is required: fast failover high availability within x time vs. true continuous availability
    - How aggressive is the availability requirement?
    - What is good enough?
  - Minimum of 4-way active-active single site data sharing across two CPCs (boxes) is the right direction in most cases for
    - True continuous availability
    - Stay away from single image constraint
  - All critical business applications should be sysplex enabled and deployed for redundancy
  - Dynamic fine-grained workload balancing to route around failure and balance the workload
    - CICSPlex System Manager
    - Distributed connections: DVIPA, System Distributor, workload balancing and automatic client reroute
  - Understand the trade-offs between system-managed CF structure duplexing vs. use of failure isolated CF and very fast structure rebuild into an alternate CF
Preventative Software Maintenance

• Problems
  • Too many customers are very back level on preventative service
  • High profile production incidents could have been avoided by missing HIPER
  • The risk of ‘no change’ is not balanced against the risk of ‘change’
  • ‘Fix on failure’ culture introduces the probability of long prerequisite chain when having to apply emergency corrective service
  • ‘One size fits all’ approach across multiple different application environment leads to escalating maintenance costs
  • Not exploiting DB2 Data Sharing technology to avoid planned outages and remove dependency on change windows
  • Lack of rigour and discipline in the methodology
  • No HIPERs or PE fixes applied since the last major preventative service upgrade
Preventative Software Maintenance ...

- **Solutions**
  - Apply preventative maintenance every 3 months
    - Two “major” and two “minor” releases
    - Refresh of the base every 6 months (“major”)
    - Each base should be based on latest quarterly RSU
    - In addition, two mini packages covering HIPERs and PEs in between (“minor”)
    - Exploit Enhanced HOLDDATA to identify and pull all HIPERs and PE fixes
  - Early adopters of new releases and/or new functions should be more aggressive about applying preventative service
  - Develop processes/procedures and technical changes to implement ‘rolling’ maintenance outside of heavily constrained change windows
    - Separate SDSNLOAD per DB2 subsystem
    - Separate ICF User Catalog Alias per DB2 subsystem
Preventative Software Maintenance ...

- Solutions ...
  - Aim for company wide ‘certification’ of new release/maintenance
    - Shared application test environment with collaborative use by systems programmer, DBA, and all the application teams
    - Additional ‘insurance policy’ before starting to roll out new DB2 maintenance package
  - Separated DB2 maintenance environments
    - Have single “master” SMP/E environment as base
    - Have one additional SMP/E environment to match each DB2 application environment
    - Certification testing based on “master” before starting to promote new maintenance to Test and Production environments
    - Can be more aggressive on applying maintenance to specific DB2 environments
    - Whilst protecting the stability of other DB2 application environments
    - Very flexible proven approach that meets application development requirements to use new functions
    - Supports the migration of new DB2 releases perfectly as DB2 application environments can be treated independently
Multiple Data Sharing Groups

- Introduced for technical and political reasons, claiming
  - Improved availability
  - Simplified operations
  - Application isolation
  - Perceived architectural limits

- Problems
  - Data Sharing Groups are not completely independent
  - Synchronous remote SQL calls over DRDA inter-connected groups
  - Resulting in more complex environment
  - Major cause of slow-downs and sympathy sickness
  - ‘Nightmare’ for problem determination and problem source identification
Multiple Data Sharing Groups ...

- **Strong recommendations**
  - Each data sharing group should be loosely coupled from the others
  - Eliminate synchronous remote SQL calls
  - Implement low latency SQL replication to provide the loose coupling and independence
  - Connect directly to the target group and avoid ‘hopping’

- **Other points**
  - Near low latency SQL replication is a huge price to pay for independence of data sharing groups which is still to be achieved
  - If the main data sharing group was managed better there would be no need to break the main production data sharing group into multiple independent data sharing groups
  - With the advent of DB2 V10 there will be significant DBM1 31-bit virtual storage constraint relief which should provide for much greater vertical scalability of individual DB2 subsystems (5x plus)
    - Opens up possibility to consolidate DB2 members and/or
    - Enhanced scaling of the main data sharing group
Running Multiple DB2 Subsystems on same LPAR

- Real storage consideration
  - A runaway DB2 member can take out the whole LPAR and take out all DB2 members running on that LPAR
  - Recommendation to use DB2 system parameter SPRMRSMX to limit the amount of real storage used by an individual DB2 member
- CPU constraint consideration
  - Single DB2 member can stall due to waiting on a global latch
  - Normally this would be released because CPU would naturally be available due to the stall i.e., self correcting condition
  - In the case when multiple DB2 members running on the same LPAR, latent demand from workload running on the other DB2 members can soak up all available CPU and not relieve the stall condition on the subject member
  - As per previous recommendation implement automation to drive regular DB2 - DISPLAY THREAD(*) SERVICE(WAIT) command at one minute interval and exploit WLM Blocked Workload Support
Normal DB2 Warm Restart vs. Restart Light

• Common Problem
  • Over use of Restart Light
  • Common misconception is that Restart Light is "Restart Fast" or "Backout Fast".

• Recommendation
  • Design point of Restart Light is for cross system restart on a different LPAR
    • Avoid ECSA/CSA shortage on the alternative LPAR
    • Avoid real storage shortage on the alternative LPAR
  • Restart Light is likely to be slower than normal DB2 crash restart
  • If you do not specify LIGHT(NOINDOUBT) option on DB2 start
    • If there are in-doubts URs then DB2 will wait for CICS to reconnect
  • If you do specify the LIGHT(NOINDOUBT) option on DB2 stat
    • DB2 subsystem will shut down and not resolve in-doubts
    • If you abort all the in-doubts then when CICS is warm started then CICS will try to re-sync for in-doubts and there will be error messages
  • Warm start both DB2 and CICS together as a pair and then go on to clean up the GRECP/LPL conditions
Storage Tuning

- Problems
  - DBM1 ASID 31-bit virtual storage constraint
    - CTHREAD and/or MAXDBAT set too high
      - No denial of service
        - EDM 31-bit Pool full condition
        - DBM1 full system contraction
        - DBM1 storage critical
        - DB2 crashes out
      - Inflated size of storage cushion
        - DBM1 full system storage contraction
  - Aggravated by
    - Lack of balance across CICS and DB2
    - Use of CICS Group Attach
    - Workload failover
    - Abnormal slow downs
    - Workload moves
Storage Tuning ...

• Problems ...
  • Shortage of real storage
    • Under provisioning can lead to excessive paging and severe performance issues
    • Long dump times when insufficient storage to cover both working set size and MAXSPACE requirement
      • Excessive amount of storage on auxiliary storage (DASD)
    • Periods when DB2 working set is pushed out to auxiliary storage (DASD)
    • Under sizing of MAXSPACE resulting in partial dumps seriously compromising PD/PSI
    • Reluctance to use bufferpool long term page fix
Storage Tuning

• Recommendations
  • Virtual storage
    • Collect IFCID 225 data from DB2 start to DB2 shutdown
    • Use SPREADSHEETDD subcommand option of OMPE to post process SMF data
    • Use sample REXX program (MEMU2) to pull the data via IFI
    • Plan on a basic storage cushion (free) to avoid full system storage contraction plus approx 100MB to allow for some growth and margin for error
    • Project how many active threads can be supported
    • Set CTHREAD and MAXDBAT to realistic values that can be supported
    • Balance the design across CICS AORs connecting to the DB2 subsystem
  • Real storage
    • Needs to be monitored just as much as DBM1 ASID 31-bit virtual storage
    • Needs to plan the right capacity and then provisioned correctly
    • DB2 should not be paging IN from auxiliary storage (DASD)
    • Keep page in rates near zero
    • Monitor using RMF Monitor III
Summary

- WLM Policy Setup
- Balanced Design across CICS and DB2
- Disaster recovery
- Continuous Availability vs. Fast Failover
- Preventative Software Maintenance
- Multiple Data Sharing Groups
- Running Multiple DB2 Subsystems on same LPAR
- Normal DB2 Warm Restart vs. Restart Light
- Storage Tuning
Session
Hard Lessons Learned From Customer Health Check Studies