B74

The Top 10 IMS DB Performance Questions (and Answers?)

Dave Viguers

dviguers@us.ibm.com

IMS

Technical Conference

Sept. 27-30, 2004

Orlando, FL

RETURN TO INDEX

Session Description

- This session will explore the top 10 performance issues with the IMS Database Manager which have been observed over the past year or so and discuss how they show up, were diagnosed, and finally resolved. If a code defect or enhancement was involved that change will be discussed otherwise ways to monitor and tune to avoid these problems will be presented.

Topics

- IRLM
- OSAM Caching
- OSAM vs VSAM
- HIDAM inserts
- VSAM buffers
- Lock contention
- Batch buffers
- Compression
- Lock data
- Last but not least

Does PC=YES or NO make a difference?

- PC=NO tells IRLM to place the lock table and other control blocks in (E)CSA
  - Lock code executed directly from dep. region
  - Less CPU overhead
- PC=YES tells IRLM to use (E)private
  - IRLM uses Program Call instruction to switch address spaces
  - More CPU but no concern for ECSA
- IRLM 2.1 allows either setting
  - May continue to use with IMS V9
- IRLM 2.2 no longer supports PC=NO
  - Shipped with IMS V9, supported with V7,8,9

So which should I use?

- Is ECSA usage a bottleneck?
  - If not then IRLM 2.1 with PC=NO
    - Still the least CPU consumption
  - If yes then IRLM 2.1 with PC=YES
- Need more than 2GB of lock storage?
  - Are you sure your not thinking of DB2?
  - Go to IRLM 2.2
    - Additional overhead
      - PC=YES & 64 bit support
    - But probably a small percentage increase
    - IRLM trace is costly
      - Use when necessary
      - Turn off later and take credit for tuning

Is OSAM caching a good idea?

- Of course, why else would it be available
- Actually, probably not for general use
  - Additional CPU cost to read and write to CF
  - DASD access times continue to improve
  - Updates still written to DASD at sync point
  - One structure for all OSAM cached subpools
- But specific uses could benefit
  - Small (or small area of reference), high activity databases
  - Reduction of DASD response could impact transaction times
  - DASD activity to data set stretching limits

Is OSAM caching a good idea?

- Caching is by subpool
  - Assign specific data bases to subpool
  - Cache 'changed' if
    - many re-reads due to buffer invalidate
  - Cache 'all' if
    - DASD read reduction is goal
      - Not just XI re-reads

Is OSAM really better than VSAM?

- OSAM is generally more efficient (CPU) than VSAM
  - Not as generalized as VSAM
  - Limited function - no index support
- Sequential Buffering
  - Only available with OSAM
  - Can significantly reduce elapsed times for batch type jobs
  - If processing is really sequential
    - Secondary Index? Could use Sort
- Buffer purge (sync point)
  - Parallel writes & Chained buffer writes

So should I convert?

- What is the problem you are trying to solve?
  - CPU
    - Converting a few DB's - probably not
    - Converting a number of heavily used ones could help
  - Batch processing elapsed time
    - Sequential buffering can have major impact
    - Try reloading an image copy to OSAM and run some jobs
- Long sync point times
  - Buffer chaining and concurrent writes
    - Reduce tran elapsed & region occupancy
    - OSAM writes are also concurrent with VSAM

Why are DB inserts causing a problem?

- HIDAM Root insert
  - Pointer Options
    - PTR=T and TB
    - Multiple segments/blocks may be locked
    - Especially bad with data sharing
    - Impact depends on concurrent threads
- Index and/or Secondary Index
  - CI and CA splits
- Where to look
  - IMS Monitor
  - Lock trace
  - VSAM listc

What can help

- Specify NOTWIN for HIDAM roots
  - Will use index for GN
    - Probably not used much online anyway
    - Might be more CPU but fewer locks probably compensates
- Index and/or Secondary Index
  - Use Freespace
  - Don't reorg index if more ISRT's expected
    - Minimize future CI and CA splits

Why did my VSAM I/O increase?

- z/OS 1.3 changed index CI size calculation
  - See Flash1080 at ibm.com/support/techdocs
    - Details
    - Migration / analysis tool
  - Will not notice until a Delete/Define (Reorg, Load, etc.)
    - If unable to open message at least you know
    - Else just assigned to different buffer size
      - Same or different pool
      - May be drastic difference in buffer allocations
  - Could also impact RECON's if changes made while online is up

Can I do anything about lock contention?

- First you need to know which PSB's and DB's are involved
  - IMSMON, LOCK TRACE, ONLINE MONs
- Check PSB procopt
  - Is it 'A' - does it need to be?
  - Change to G if no updates
  - How about GOx? Appl handle GG/BA?
- How many records in a CI?
  - CI size reduction may avoid conflicts
  - Especially in data sharing environment
- In some cases less concurrency might yield more thruput

Anything else?

- Check pointer options previously mentioned
- Reduce time holding locks
  - I know this sounds silly but
    - Buffer tuning may reduce overall appl time
    - Therefore time holding locks
    - OSAM may get thru syncpoint sooner
    - WADS's can also impact syncpoint time
  - Look at I/O's per call
    - Long twin chains could be an issue
    - Maybe a secondary index is in order
- Updates to 'control' segment
  - Maybe order of update be changed

Will additional buffers help my batch?

- First let's talk about read jobs
  - If access is mostly sequential
    - Very little, if any, re-access of data
    - Increasing buffers probably won't help
    - Use faster DASD and/or OSAM SB
  - If random access
    - Frequent re-reference of data
      - Large buffer pools can help
    - If little or no re-reference
      - Increasing buffers probably won't help
  - If multiple DB's involved
    - Separate (sub)pools if different access patterns

What about batch update?

- Same considerations as read plus
- Buffers written when
  - Application Checkpoint
  - All buffers modified
    - If VSAM use BGWRT active and high %
    - OSAM will chain write the buffers
- Logging
  - IEFRDER
    - Buffers and Blocksize should be specified
    - Can be DASD or Tape (maybe virtual tape)
    - IMS forces TWI for integrity and recovery
      - Real tape can be the bottleneck
      - Some Virtual tape hardware not so good either

Any issues with segment compression?

- IMS has HALDB and 2048 AREAS
  - IBM sells DASD so no need - right?
- Compression is generally good
  - More data per block and fewer I/O's
- But watch out for
  - Heavily referenced segments
  - Latch contention
    - DMBB
  - Replace activity
    - Might split segment
      - Multiple blocks
      - Increased locks and contention

How can I find long lock holders?

- Only really a problem if someone waiting
  - Except for maybe lock storage
- Can't run IMS Monitor or lock trace all day
- Nothing on the log to identify them
- Online monitor may help identify
- IRLM Long Lock detection as trigger
  - F irlmproc,SET,TIMEOUT=sss,imsid
    - Can use RMF or SMF 79.15 records
    - This high level info might be enough
      - Depends on Appl to PSB relationship, etc.
  - Could SLIP on DXR162I message - get dump
  - Even better, run with lock trace 'OUT' to DFSTRAxx, turn off when DXR162I is issued

Can anything outperform IMS?

- Is DB2 really as efficient and fast as IMS?
- First some facts
  - DB2 has improved vastly over the years
    - CPU's are faster and bigger
    - Sysplex means even more power
    - DASD is really fast
    - Lot's of real storage available
  - IMS has not exactly been standing still
    - IMS runs on the same CPU's
    - IMS takes full advantage of sysplex
    - IMS data can exist on same DASD
    - But IMS can't yet use quite as much storage
      - 64 bit exploitation will come more gradual

So what IS the answer?

- As always - It depends
  - What are you measuring?
    - Transaction elapsed time, raw thruput, cost?
  - Certainly some applications will perform better with DB2
  - High volume transactional processing with IMS is hard to beat
  - There probably isn't ONE right answer
  - Decision should be based on business needs
    - Not based on unfounded statements
    - Appl. needs and environments all different

Summary

- IMS DB performance and tuning can be complex
  - Larger databases
  - More sharing systems
- Lots of options and tuning parameters
- Only hit some of the questions
  - Many others would take hours by themselves
    - HALDB
    - BMP vs Batch
    - and on and on and on

RETURN TO INDEX