E85

IMS TM Performance Considerations

Dave Viguers

IMS

technical conference

Las Vegas, NV   **September 15 - September 18, 2003**

Session Description

■ The IMS Transaction Manager supports high volumes and large networks very efficiently. However, there are many tuning variables which can have an affect on the overall performance of your IMS environment. This presentation looks at when, where, and how the various storage pools and other transactions manager parameters can influence the throughput and performance of a message as it is processed through the IMS Transaction Manager.

Topics

- Overview of IMS Transaction Manager
- Input Processing
- Editing / Formatting
- Message Routing
- Queuing
- Scheduling
- Output Processing

- ETO
- Fast Path
- Storage Management
- Tracing
- Logging
- Misc.
- Summary

Transaction Manager Overview

- Handles network interface
  - ► Session management
  - ► Edit and format messages
- Route messages to destination
  - ► This IMS? Other IMS? CICS? MQSeries? APPC? OTMA?
- Message Queue Management
  - ► Private Queues / Shared Queues
- Transaction Logging
- Application Scheduling

Some things have changed

- Good news
  - ► Faster CPU's
  - ► Faster DASD
  - ► More storage
- Bad news
  - ► Faster CPU's
  - ► Faster DASD
  - ► More storage

What is tuning?

- Making the most of the available resources
- Physical Resources
  - ► CPU
  - ► Storage
  - ► I/O
- Logical Resources
  - ► TCB'S, SRB's
  - ► Storage Pools
  - ► Latches / Locks

Input Message Processing

- RECA
  - ► If none available VTAM queues in dataspace
  - ► Small number is normally enough
  - ► Stats will almost always show max was used
    - – Usually this condition occurs at checkpoint
  - ► If always high then probably need to look elsewhere
- SAP
  - ► Saveset must be assigned before ITASK can be started
  - ► Selective dispatching may occur if not enough
  - ► IMS will dynamically expand if necessary

Editing / Formatting

- Message Format Pool
  - ► Size depends on number and size of blocks
  - ► Increase size to decrease I/O
    - − Every I/O is a suspend / resume of task
    - − increases time that ITASK is active which
    - − increases need for SAPS and
    - − increases demand for RECA which is still held on input
- Fetch Request Elements
  - ► Must be one for every format in pool
  - ► Startup number will be expanded dynamically so rarely will 'blocks washed for FRE' occur

Tran / Terminal Definitions

- OPTACK (Terminal or TYPE macro)
  - ► eliminates VTAM pos ack on input
    - – Only for FINANCE, 3601, SLUP
- TRANSACT INQ=(YES,NORECOV)
  - ► minimize overhead (logging, CPU)
  - ► only would make a real difference if lots of trans could fit this definition
- Conversational
  - ► Causes additional queuing and logging
  - ► use when necessary for function, not as default
- TRANSACT MSGTYPE=RESPONSE
  - ► reduce DC turnarounds, CHKW's, WTWT's

Full Function Queuing

- QBUF=nnnn (DFSPBxxx)
  - ► Goal is to reduce I/O
  - ► May not always be possible if some destinations normally queue
- EXVR=Y or POOLS=QBUF
  - ► long term page fix the queue buffers
  - ► if large numbers of QBUFS then watch out for possible paging consequences
  - ► If not doing I/O then doesn't really help
  - ► If doing I/O then reduces CPU cost for I/O processing

Full Function Queue Datasets

- QBLKS
  - ► Often overlooked as I/O problem
- SMSG (up to 10)
- LGMSG (up to 10)
- Monitor I/O response time with RMF
  - ► Especially critical if using MSC
- Look for writes occurring at checkpoint
  - ► Buffer purge
    - − Means buffers contain 'altered' data for a relatively long time
    - − Is this normal for your installation?

Fast Path Input

- EMH processing may substantially reduce message processing overhead
  - ► Must live within restrictions
- EMHL= (DFSPBxxx)
  - ► Sets default EMHB size if specified
- EMHB= (DFSPBxxx)
  - ► Larger of EMHL or TRANSACT
  - ► Gotten on first FP input
  - ► Size will increase but not decrease
  - ► Kept until session termination

Fast Path BALG

- ■ Messages are assigned to BALG by RTCDE
  - ► Default RTCDE is TRANSACT name
- ■ One or more dependent regions
  - ► IFP regions
  - ► Scheduled only once at region initialization
  - ► One PSB per region
- ■ Some queuing is not necessarily bad
  - ► May reduce MVS Suspend/Resume
  - ► Reduces CPU overhead
  - ► Don't tune to this but.......

Scheduling (1)

- The best schedule is the one that doesn't happen.
- Pools
  - ► DMB
    - – make big enough to hold ALL DMB's or else
    - – database close / open - very disruptive
    - – ACBLIB I/O
  - ► DBWP (actually gotten as needed on call)
    - – database work pool
    - – doesn't hurt to make too large but generally is very small anyway

Scheduling (2)

- Pools (continued)
  - PSBW
    - Must handle concurrently scheduled PSB's
    - Too big is no problem
    - Too small may keep available region from scheduling
  - PSB (CSAPSB / DLIPSB)
    - Balance pool sizes - increasing one may not help
    - Reduce I/O and castout process
    - Consider effects of parallel scheduling
    - Have seen some very large pools

Scheduling (3)

- Pools (continued)
  - ► EPCB
    - ► Fast Path databases only
    - ► Like PSBW - Too big doesn't hurt but too small does
- PST=
  - ► Balance with pools or else adding regions could make things worse
  - ► Will expand dynamically up to MAXPST
    - − Cost to create / tear down if frequent
  - ► Don't forget BMP's
  - ► This number is also default VSAM strings

Scheduling (4)

- **PARLIM / MAXRGN**
  - Use together to avoid flooding regions
  - Can significantly affect pool usage
- **PROCLIM**
  - Max messages processed in single schedule
  - Quick re-schedule is automatic
    - Unless PROCLIM=0
- **Resident options for APPLCTN/DATABASE**
  - Can be used to get some very large PSB's or DMB's isolated from pools
  - However PSB copies (parallel) are in pools
  - Online change will make non-resident

WFI / PWFI

- Use if possible (remember the first scheduling statement?)
- PWFI is usually more appropriate since it is just a region parm
- Use IMS Class Scheduling
  - ▶ Don't fall into the 'single' class trap
  - ▶ May work fine today but watch out as workload changes
  - ▶ Best to isolate high volume trans to their own classes where possible
  - ▶ If not practical then use PARLIM/MAXRGN to minimize 'thrashing'

New Scheduling Statistics

- Placed in '4506' log record at checkpoint
- Added to IMS V6 and above
- PSB (CSA and DLI) and PSBW pool counts and times
- Helps you understand the cost of scheduling
  - ► Can be high with todays large pools

Program load / initialization

- These options really haven't changed and therefore we won't go into here.
  - ► Program Fetch
  - ► DBLDL
  - ► Virtual Fetch
  - ► LLA / VLF
  - ► Preload
  - ► RENT, RES, NODYNAM Cobol options
  - ► LIBKEEP runtime option
  - ► LRR for LE

Output Message Processing

- OUTBUF
  - ► Best if can hold entire message
  - ► May be limited by physical device buffer
  - ► VTAM SEND for each OUTBUF
- RESPONSE MODE
  - ► eliminates unlock keyboard
- VAUT=1
  - ► Use VTAM authorized path

Fast Path Output

- **FPACK**
  - ► eliminate positive ack on FP output - LU0 only
- **OTHR**
  - ► Number of concurrent DEDB output operations
  - ► Delayed buffer writes may cause delayed response times
- **LGNR**
  - ► Too small may cause extra CPU overhead
    - – and / or
  - ► Increased log volume
- **Fix BLOCKS=EPST (or BLOCKS=FP)**

ETO Considerations

- ETO can minimize storage and checkpoint times if more terminals defined than used
- ASOT / ALOT
  - Control block creation / deletion
  - Session initiation / termination
- Hash Tables
  - CHTS, LHTS, NHTS, UHTS
- Output creation exit
  - Invalid destinations could cause queue buildup
- DLQT
  - Remember undelivered messages are interspersed on Queue data sets

IMS Fixed Pool Storage Manager

- DFSSPMxx
  - ► Pools are dynamically expanded / contracted
  - ► If large variations then could cause unnecessary overhead
  - ► POOLS
    - CIOP, HIOP, EMHB, FPWP, CESS, LUMP, LUMC, AOIP
    - Look for pools where excess storage is being used and
    - A lot of gets / frees
    - Look for 'oversize' being used
    - Override where necessary

Traces and Serviceability Options

- On is OK for most traces and in fact may be invaluable in diagnosing problems
  - ► LATC is exception because of extra overhead
- Use OUT option sparingly and only for short intervals
  - ► Goes to DFSTRAxx if allocated
  - ► Otherwise goes to LOG

Logging

- OLDS
  - Old rules still apply
    - Fast Dasd - Isolated Volumes
    - Large blocks but > 2GB must be mult of 4K
    - Buffer to reduce or eliminate buffer waits
    - Dual OLDS is only as fast as the slowest
- WADS
  - Fast Dasd
  - Especially critical if MSC is heavily used
  - Dual WADS also only as fast as the slowest

Log Volume

- Current devices can log over 20 MB/sec
- Helps especially with checkpoints
- Ways to reduce volume
  - Non-recov DB's
  - Proper LGNR for FP
  - ETO
  - Address any application abends

Checkpoints

- Lots of data written in burst
  - ► Have seen more than 30MB
  - ► Fastest DASD keeps time to minimum
- Frequency
  - ► Timed intervals (automated) or CPLOG?
  - ► Watch out for interaction of both
- Paging
  - ► All those unused Terminal controls blocks referenced
- Qbuffer Purge
  - ► Only buffers which have remained modified since before previous checkpoint

Latch Conflicts

- LOGL
  - ► Typically the highest contention - just make sure you have enough log buffers
- ACTL
  - ► Monitor latch - check bufno / blksize
- Scheduling latches (SCHD, TCTB, etc.)
  - ► Check scheduling pools
  - ► Reduce scheduling
- Others
  - ► Normally low but watch for trends

Security

- Multiple RACF TCB's help signon
  - ► Should also have multiple RACF data sets
- APPC / OTMA
  - ► Signon done first outside IMS
  - ► Depending on security level then again in IMS
  - ► Make sure ACEE is cached
  - ► Otherwise security can limit transaction thruput
- DFSBSEX0 user exit
  - ► Consider bypass of security on 'backend'
    - − might be appropriate for MSC or SMQ

Summary

- **Highly efficient by design**
  - Wrong specifications might negate
- **Many constraints have been removed**
  - However systems getting larger
- **Can generally make pools very large**
  - Avoid I/O where possible
- **Monitor regularly**
  - Know what's normal and what is not
- **Watch for trends and changes**
  - An ounce of prevention - well you know!