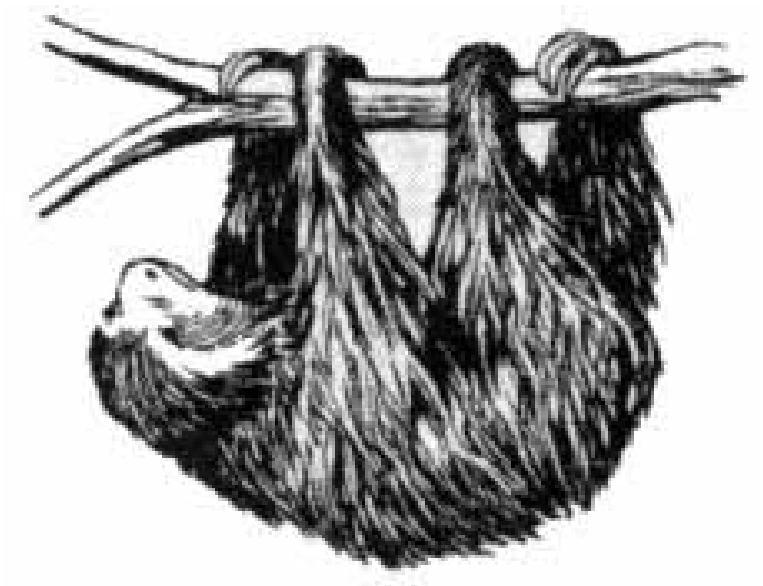
DB-5: OpenEdge® Performance Made Easy

Adam Backman
VP of Technology
White Star Software

PROGRESS SOFTWARE Exchange 2006

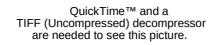
Agenda



- Why Tune?
- Procedural approach
- When to tune
- Hardware setup
- Configuring OpenEdge

Why Is Performance So Important?

- People hate to wait
- People like predictability
- Internet makes this even more important


QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Approach For Performance Tuning

- Determine where you are now
  - Establish a baseline
- Change one thing at a time so you can determine the effect of the change
- Work from the slowest resource to the fastest

How Fast is Fast?

Remember: Performance is relative so it is important to have a baseline.

Hardware Resources (Slowest to Fastest)

- Network
- Disk
- Memory
- CPU

When to Tune Performance?

- Installation or upgrade of hardware
- Installation or upgrade of software
- Change in workload
  - Number of users
  - Volume of data

Application Performance

The single largest component of performance tuning is application performance.

A well written application will run well given the proper resources while a poorly written application will rarely run well despite resource allocation.

UPDATE STATISTICS

- If you are a SQL user you should use the UPDATE STATISTICS function within OpenEdge to ensure your queries are as efficient as possible
- How Often? Weekly would be nice but I think monthly is more practical for most
- Caution: This can take some time to run

Network contention

The network is the slowest resource for client/server applications so you want to eliminate contention for this resource before moving on to the other resources

Networking tips

- Keep things local
  - No temp files on network drives
  - Move the application “close” to the user
- Use -cache to speed initial connection
- Use -pls if you are using program libraries over the network
- Application issues are magnified over a network (field-lists, no-lock, indexes, …)

How About SAN and NAS Traffic On The Network

- SAN - Storage area network
  - This will have it's own network fabric so it is generally not an issue
  - Multiple machines attached to a single SAN device can experience SAN traffic collisions and this can effect performance
- NAS - Network attached storage

  This can share the same network as the client processes so care needs to be taken to avoid contention

Odd Network Issues

- Bandwidth
- Speed - Use the fastest and avoid auto
- Always use full duplex
- Sharing network with multiple protocols (generally and old issue)
- Latency (WAN Issue)
- Mixing hardware brands

Disk contention

In most environments disks are the largest area for improvement. All of the data flows from the disks to the other resources so this effects both local and networked users.

The goal is to use all of the disks on the system evenly. This is referred to as eliminating variance

Disk tips

- No RAID-5
- Use storage areas (Type II where necessary)
- Use 8k block size
- Use the correct BI cluster size
- Use page writers
- Use private buffers (-Bp)
- Use -T to eliminate variance

Disk Tips (continued)

- Buy many small disks
  - Two heads are better than one
- Buy fast disks
  - Buy at least 10,000 RMP
- Buy fast controllers
    - Fibre channel is better than SCSI

No RAID 5

- Poor man's mirroring - This is the kiss of death for OLTP performance
- User information is striped
- Parity information is striped WITH user information
- OK for 100% read only applications
- Poor performance for writes

If no RAID 5 what do I do?

- RAID 0+1 for the database
- RAID 1 for the AI and BI logs
- Hardware RAID is almost always better than software RAID

Storage Areas

- Progress-specific way to distribute I/O
- Only way to eliminate I/O indirection in a Progress environment
- Only way to pre-allocate database blocks
- Every database has at least 1 storage area

Type II Storage Areas

- Also known as clustered storage areas
- User definable clusters (8, 64, 512 blocks)
- Data is clustered together (all customer records are in their own blocks)
- The clusters are chained together. This makes bulk operations (index rebuild) much faster as less blocks need to be examined

Type II Storage Areas (continued)

- How to choose a cluster size
  - Consider your data
    - Number of records
    - Size of your records
  - Consider your layout
    - Which tables are in which area
  - Consider how you use the data

Type II Storage Areas (A Few More Notes)

- Why not choose 512 all of the time
  - Open debate (very little real data)
  - Waste space with small tables
  - Possible very expensive insert (adding multiple clusters at the same time)
- Generally I choose 64 or 512 for large tables and 8 or type 1 for small tables

Storage Areas

- Pros:
  - Reduce utility time (especially true of Type II)
  - Increase sequential access performance
- Cons:
  - Adds complexity
  - More potential room for failure

8k block size

- Most systems will benefit from using 8k block size (NT should use 4k)
- You will retrieve more information per physical I/O especially on index reads
- I/O is done how the operating likes it to be done.
- Remember to modify records per block if necessary

BI Cluster size

- Somewhere between 1MB and 4MB works for most people
- If you are checkpointing every 2 minutes or more often during peak periods increase the cluster size
- If you a “workgroup” version of Progress leave your cluster size alone (512kb)

OpenEdge® page writers

- Every database that does updates should have a before image writer (BIW)
- Every database that does updates should have at least 1 asynchronous page writer (APW)
- Every database that is using after imaging should have a after image writer (AIW)

Tuning APWs

- Start with 1 APW
- Monitor buffers flushed at checkpoint on the activity screen (option 5) in promon
- If buffers flushed increases during the “important” hours of the day add 1 APW

Private Buffers (-Bp)

- Not really private
- Use different LRU chain
- Good for read only clients (reports)
- _MyConnection._MyConn-NumSeqBuffers to allocate or deallocate from the application level

Use -T to level disk I/O

Local (host based) users and batch jobs should use the -T parameter to place their temporary file (.srt, .pge, .lbi, …) I/O on a drive that is not working as hard as the other drives on the system

Note: -T should never point to a network drive

Memory contention

Memory should be used to reduce disk I/O.

Broker (server) side parameters should be tuned first and then user parameters can be modified. In a memory lean situation, memory should be taken away from individual users before reducing broker parameters.

Memory hints

- Swapping and/or excessive paging is bad, buy more memory or reduce parameters to avoid it
- Increase -B in 10% increments until the point of diminishing returns or swapping, whichever comes first
- Use V9 private buffers (-Bp) for reporting
- Do not use private buffers (-I) prior to V9

Memory Tools

- Promon (activity screen) or the _ActSummary VST can be used to view buffer efficiency
- At the OS level, sar, vmstat or a system specific tool like glance can be used to view memory statistics
- Fathom management provides a way to view this information in realtime as well as to see trends over time

Memory hints (continued)

- Use memory for the users closest to the customer first (developers increase last)
- Use -Bt for large temp tables
- Set -bibufs between 50 and 120. Look at the activity screen in promon (BI buffer waits) to see if additional tuning is necessary.
- Use -directio to bypass the OS buffer pool

CPU contention

High CPU activity is not bad in and of itself but high system CPU activity is bad and should be corrected.

A host-based system will have a higher percentage of User time that client/server or n-tier

Components of CPU activity

- USER - This is what you paid for
- SYSTEM - This is overhead
- WAIT - This is waste
- IDLE - This is nothing ;-)

CPU activity goals

The goal is to have as much USER time as possible with as little SYSTEM and WAIT.

A practical split is USER: 70%

SYSTEM: 25%

WAIT: 0%

IDLE: 5%

This represents a good goal for a host-based system

Eliminating high SYSTEM CPU activity

- Always use -spin
  - Use a setting of 1 for single CPU systems
  - Use a higher setting for multiple CPU systems
- Testing has shown that the optimal setting will vary significantly based on CPU clock speed
- Start at 2000 and work up (perhaps way up) from there

Eliminating high WAIT CPU activity

- WAIT = Waiting on I/O
- If you still have IDLE time it generally is not a big problem
- Look at paging/swapping first
- Next look at your disk I/O

Points to remember

- Document where you are now before changing anything
- Change 1 thing at a time so you know what made a difference
- Move from the slowest to the fastest resource

Conclusion

- Application
  - Take best advantage of OpenEdge indexes
  - Use UPDATE STATISTICS for SQL
- Network
  - Keep things local
  - Keep number of users per server low
    (increase -Mn)

Conclusion

- Disk
  - Eliminate variance
  - Use as many disks as possible
  - DB block size and BI cluster size are key
- Memory
  - Use 95% as the goal for buffer hit percentage
  - Use -Bp to reduce impact of reporting
  - Tune broker parameters first

Conclusion

- CPU
  - Look for the right ratio of User to System
  - Use -spin (if available)

Questions?

Adam Backman

adam@wss.com

Thank you for your time!