Enabling Starcat Performance Scaling

Bob Sneed
Sun Microsystems, Inc.
Performance and Availability Engineering

SUPerG / Spring 2005
Washington, DC
April 20, 2005
Rev 0.4 – June 14 2005
@OAUG

Sun microsystems

Agenda

- Overview
- Hardware factors
- Solaris factors
- Practices and “big knob” tuning
- Recent Solaris enhancements
- Application tuning – no, that's another session!
- Q&A

- NOTE: Bias here is towards large Oracle apps

Disclaimers

Opinions and views expressed herein are those of the author, Bob Sneed, and do not represent any official opinion of Sun Microsystems, Inc.

There is no warranty, expressed or implied, in the quality of the information herein.

Batteries not included.

Your mileage may vary.

Overview

What challenges scaling?

- Strategic
  - "Throwing iron at the problem"
  - “Bad Practices” - “The way we've always done it”
  - Managing expectations & sizing methodologies
  - Non-optimal solution architectures
- Technical
  - Ever-larger domains
  - Fewer/faster processors
  - Locks and synchronization objects
  - Sharing anything (CPU, I/O, memory ...)
  - Hardware and software revision levels
  - Design limits of algorithms in kernel and libraries

Some Low-Level Scalability Factors

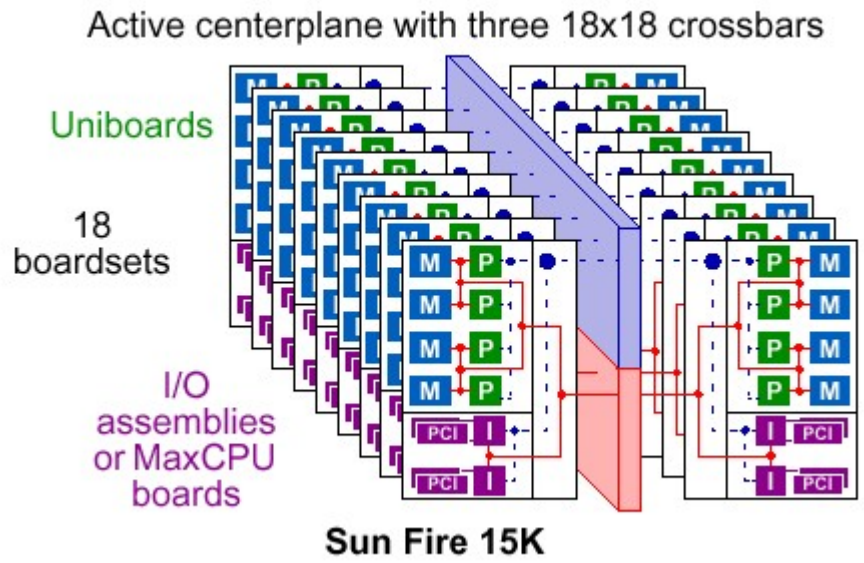
- The kernel cage and Dynamic Reconfiguration (DR)
  - kernel memory focused on a single board
  - sometimes limiting for large domains, but often issues are only provoked due to configuration practices
- Solaris segmap mechanism
  - Filesystem buffering involves global lock activity
- Global cache coherency
  - Global locks require global cache coherency
  - StarCat systems have a 3rd-level coherency mechanism on the expander boards, which contain an “AXQ” chip
- Cache-to-cache copies
  - When the CPU ownership of the most recent change occurs, data must be copied
- Local versus remote memory

Common Customer Perspectives

- Recent experiences
  - Cannot reproduce production load in Q-A/Test/Stress environment; too costly, complex, and time-consuming
  - High-end upgrades sometimes lead to alarming statistics or disappointing capacity forecast
  - Typical “performance cases” involve multiple root causes; best to fix proactively versus reactively
- Default expectations
  - Bigger is better
  - Applications will scale near-linearly
  - Existing capacity planning techniques are OK
  - Existing practices “like we've always done it” are OK

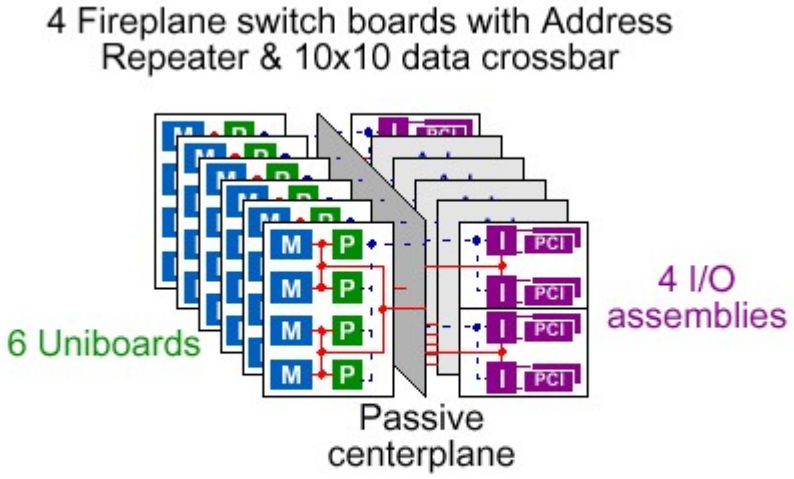
Hardware Factors

StarCat Architecture



Sun Fire 15K

Excerpted from: "Solaris Memory Placement Optimization and SunFire Servers", by Alan Charlesworth, March 2003 (Easily found on www.sun.com using Google)



Sun Fire 6800

Hardware Factors

- Performance-impacting components upgraded since first StarCat release
  - Expander Boards
    - AXQ >= 6.2 needed to support LPA+MPO to deliver full S9 benefit
  - hsPCI+
    - Some older cartridges did not do 66 Mhz correctly
    - I/O multiplexor chip (PCI bridge) improved in newer gear (older chip had issue requiring pcisch driver to use an extra DMA sync operation, which can diminish the odds of data streaming)
    - New bridge chip has deeper data buffer that improves data streaming
  - CPUs
    - US-III - “Cheetah”
    - US-IV - “Jaguar” – dual-core US-III+
    - Soon: US-IV+ - “Panther” - dual-(improved)-core, enhanced cache architecture

Hardware Factors

- New components shipped as standard in new systems - of course!
- Be aware of what you have
- Consult Sun for upgrade pricing

Solaris Factors

Solaris for StarCat Systems

- Solaris 8 - Design point: E4500
  - Now entering the EOL process
  - IP scalability permanently limited
  - Reduced price/performance; no MPO, no MPSS
- Solaris 9 - Design point: F12K
  - Prevailing “Best Practice” on all systems
  - Improved CPU, IP, and memory scaling
  - Improved instrumentation and controls
- Solaris 10 - Design point: CMP/CMT
  - Best technically - but ...
    - Data centers are not usually “early adopters”
    - Some key ISV certifications still in process
    - Training required for Fault Management Architecture (FMA) - at least

Solaris 9 - The Recommended "Minimum" OS Release

- Solaris 9 - high-order bits ...
  - A "Best Practice" on high-end gear
  - Delivers generally improved performance
  - Required minimum for best StarCat performance
  - Better support scenario than Solaris 8 (less to fix)
  - Extreme and seamless Solaris 8 compatibility
  - No re-training required
- Countdown the "Top 8" reasons ...

Solaris 9 - Reason #8

- In general, improved performance
  - “M-Value” uplift (Sun's “horsepower” metric)
  - Numerous smaller fixes
  - Enhanced UFS and NFS performance
  - Some things, for example fsync() performance, are abysmal in Solaris 8 – an will never be improved there
  - Other performance improvements more likely to flow down to Solaris 9 than Solaris 8 (many have Solaris 9 foundation prerequisites)

Solaris 9 - Reason #7

- IP scaling enhancements
  - Multi-threaded (MT) STREAMS scheduling
  - IP syncq fixes
  - Overall more robust “out of the box” networking
  - A major category of where high-end users “hit the wall” with Solaris 8 design limits

Solaris 9 - Reason #6

- Memory Placement Optimization (MPO)
  - Required to get best price/performance from StarCat
  - 8-12% OLTP uplift often cited
  - Some compute-bound tasks benefit as much as 300%
  - Benefits are immediate and automatic (on properly-configured StarCat of recent vintage)
  - Quick check (as root)
    - echo “nlgrps/D” | mdb -k
    - multi-board domains should show greater than 1
  - NOTE: Requires modern AXQ hardware

Solaris 9 - Reason #5

- Multiple Page Size Support (MPSS)
  - Decreases overhead of managing Very Large Memory (VLM)
  - Offers tuning opportunities for applications with large memory footprint
  - Directly exploited by Oracle 9i

Solaris 9 - Reason #4

- Improved instrumentation
  - 'trapstat' - facilitates MPSS benefit analysis
  - %sys more accurately counted
  - improved 'ipcs' and 'pmap' commands

Solaris 9 - Reason #3

- Improved Resource Management facilities
  - FSS - Fair Share Scheduler (previously Solaris Resource Manager (SRM)) - now bundled for free
  - FX - Fixed priority scheduling class added - useful for improving efficiency in complex workloads
  - Persistent processor sets (psrsets) - processor sets can be made to survive reboots
- These can help a lot when pushing the system scaling performance envelope

Solaris 9 - Reason #2

- Maturity - Stable and market-proven
  - Solaris 8 - “old” - about to EOL, known limitations
  - Solaris 9 - “mature” - reduces life cycle maintenance costs, driving down Total Cost of Ownership (TCO)
  - Solaris 10 - “wonderful” - but requires some training and integration investments

Solaris 9 - Reason #1

- "None of the above"!
  - Once again, many S9 improvements will never be in S8
  - Having S9 as a baseline avoids re-evaluating S8 issues and limitations if there are operational issues
  - Diagnosing new issues in S10 is much easier with DTrace ... but S9 can give huge serviceability gains without a big adoption delay

OS Version – Not a panacea!

- Sometimes, tangible difference will be zero
  - "Nominal" operations may be imperceptibly different
  - The shape of the scaling curve changes at the high end
  - Perceived difference may be partly due to improved instrumentation in newer releases
- Many issues are not addressed by the OS version
  - ISV certification gaps
  - Bad application configuration and tuning
  - 3rd-party factors (Storage, system software)
  - Interrupt management pathologies
  - Pathological locking / non-scalable code
  - … but there is continuous progress in all of these areas

Solaris 9 – Apparent Barriers

- ISV Certifications
  - Generally, a “paper tiger”
- Q-A/Test and promote-to-production costs
  - Another “paper tiger” - patching S8 is higher-risk
- Customer staff training
  - None required, really
- “Skip 9, go straight to 10”
  - OK, if it will happen quickly
- “If it isn't broken, don't fix it!”
  - It's all a matter of TCO
- Unforeseen compatibility issues
  - We just haven't seen any!
  - Occasionally, a script will gratuitously check OS version

Practices & “Big Knob” Tuning

Recurring Issues

- "Worst-of-all worlds" I/O tuning
  - disk_asynch_io=false
  - db_writer_processes=10
  - self-imposed filesystem constraints (single-writer lock and/or memory management underlying filesystem caching)
- Fear of large SGA sizes
  - 64-bit Oracle is all about using large memory efficiently
- Incorrect evaluation of filesystem options
  - When the OS page cache is removed, the Oracle SGA typically needs to be increased to exploit memory
  - When AIO is used (the default) the motivation for multiple DB writers is radically reduced

Buffered Filesystems Limit Scaling

- Application scaling can be limited by multiple "invisible" factors that are not in sizing models
  - segmap scalability
  - AXQ bandwidth
  - kernel cage bandwidth
  - cache-to-cache copies
  - cost of database block checksum re-validation
- Not all "unbuffered" solutions perform the same; VxFS "direct I/O" enforces a single-writer lock where UFS "direct I/O" does not
- The "extra copy" of I/O buffering can begin to hurt at high I/O rates, but the first things to bite are locking and serialization effects

Filesystem Scalability Factors

- Memory management (segmap) scalability
  - underlying mechanism involves global locking
  - huge cross-call generator
  - tunable, but tuning rarely satisfies (segmap_percent)
  - much improved in Solaris 10
- POSIX single-writer lock throttle
  - impacts reads as well as writes (!)
- Filesystem logging mechanisms
  - impact varies with workload
  - impact varies with I/O modes
  - good news: most Oracle files are static-sized
  - good news: tweaks like “noatime” mount option are sometimes worthwhile

AIO Clarifications

- Solaris supports Asynchronous I/O to all file targets
  - API contained in libaio.so
  - For RAW, QFS samaio & VxFS Quick I/O, uses kaio(2)
    - This is a Sun private, non-public API
  - For ordinary files, uses LWP-based threads
    - LWP payload is a pwrite() or pread()
- Implementation inside libaio.so has evolved ...
  - Used to see kaio() -> ENOTSUP as normal for filesystem targets
  - Now used pre-spawned LWPs and remembers target attributes

Filesystem & Feature Selection

	Cost	Logging	Write		Kernel IO	Administrative	Performance
RAW	FREE[1]	N/A	YES	NO	YES	HIGH	BASELINE
			NO	YES			
UFS direct I/O	FREE	YES [2]	YES	NO	NO	LOW	NEARLY EQUAL
			NO	YES			
QFS Q-Writes	-	N/A	YES	YES	NO	LOW	
QFS samaio	-	N/A	YES	NO	YES	LOW	NEARLY EQUAL
			NO	YES			
			NO				
VxFS Quick I/O (QIO)	$++	YES	YES	NO	YES	HIGH	NEARLY EQUAL
VxFS Cached Quick I/O (CQIO)	$++	YES	YES	YES	YES	HIGH	
VxFS Oracle Disk Manager (ODM)	$++	YES	YES	?	YES	VERY LOW	NEARLY EQUAL

Network Scaling Issues

- **“IP syncq” - system “hangs”, high %sys**
  - Better fixes in Solaris 9, still better in S10
  - Incremental improvements in Solaris 8
- Gigabit Ethernet slow relative to non-StarCat
  - PCI latency limiting
  - hsPCI+ may benefit
  - Application-specific tuning may help
- Possible CPU saturation from interrupt handling
  - Can cascade into degraded RT performance
- Possible degradation from interrupt competition
  - NICs can compete with each other or HBAs
- “localhost” performance being improved
  - Critical to messaging apps and much more

Network Scaling Remedies

- Tune application
  - Fewer, larger packets better
  - Fewer packets always a win
- Prefer Solaris 9 or 10
  - S9 IP sync patches - better than Solaris 8
  - S9 -> MT STREAMS scheduler; S10 -> “Fire Engine”
- NIC selection; Prefer Cassini (ce), generally
  - Many tunables! (Too many - see Sun Online Blueprints)
  - “Jumbo Frames” - useful in some settings

Network Scaling Remedies

- Other possibilities
  - Consider possible benefit of hsPCI+
  - Tweak sq_max_size: default of 2 is too small, “20” is a “good number”; tuning is controversial!
  - Use Ethernet flow control
  - Using more “less-loaded” NICS
  - Card placement in I/O boats; bus sharing
  - Interrupt Fencing; processor sets
  - Using trunking & IPMP across NICs

Network Scaling Remedies

- In summary ...
  - Expert guidance may be needed
  - Gigabit Ethernet should always be purposefully tuned on high-end systems
  - A simple means is needed for specifying an appropriate NIC population early in a project
  - Expect the products to become self-tuning over time

Recent Solaris Enhancements

Recent Solaris Enhancements

- Some context ...
  - Recent patches have aggressively addressed issues with high-end application scaling
  - This is not just a routine “install the latest patches” pitch!
  - Some enhancements are still trickling into Solaris 8, but it will forever have some inherent limitations

Recent Enhancements

- “Idle loop” issues
  - Solaris idle loop degrades throughput on largely-idle systems - and not-so-idle systems, too
  - Explains most of “my system runs better when it's busier” and “adding CPUs hurt my performance”
  - Increased %sys
  - Increased migrations
  - Poor CPU cache utilization
  - Impacts network throughput indirectly
  - Fix is generally beneficial; but critical for larger domains
- “Idle loop” answers
  - Addressed in Solaris 8, 9 (KU-17), and of course S10

Recent Enhancements

- Semaphore scaling issue
  - poor scaling with high contention on high CPU counts
  - key to scaling major applications like Oracle
- Semaphore scaling answer
  - “exponential back-off” algorithm reduces contention
  - Delivered in recent patches of S8, S9, and of course S10
  - (More refinements in-process: use of “undo” has scaling limitations … but Oracle does not use this)

Recent Enhancements

- fsflush issues
  - Multiple issues cascaded from fsflushd hogging a CPU (at SYS priority)
  - On large-memory systems, “Best Practice” has been “set autoup=<4*N_GB_RAM>”
- fsflush answers
  - Addressed in S8, S9 (KU-17, or just prior), and of course S10: CPU usage by fsflushd greatly reduced
  - “set tune_t_fsflushr=1” - new default setting; probably a Universally Good Idea (so don't override)
  - Need for autoup tuning reduced or maybe eliminated - but not yet fully characterized

Recent Enhancements

- fsync(3) and fdsync(2) issue
  - Tragically poor performance in S8
- fsync(3) and fdsync(2) answer
  - S9, U5 minimum.

- MPO issue
  - No memory placement and related scheduling improvements without it
- MPO answer
  - S9 is the required minimum. (recent patch advised)

Recent Enhancements

- Thread starvation issue (not so recent, actually)
  - Low-priority sleeping threads never got higher priority
  - A major cause of Oracle crashes from AIO timeouts on filesystem-based asynchronous I/O (AIO)
- Thread starvation answer
  - “set TS:ts_sleep_promote=1” in /etc/system as Best Practice on Solaris 8 only (it's default in Solaris 9)
  - This tunable completely removed in Solaris 10

Memory Management Issues

- Using Dynamic Intimate Shared Memory (DISM)with an inadequate OS patch level
  - Triggered by setting sga_max_size to anything
  - DISM uses small pages only on Solaris 8; not generally recommended where performance is critical
  - Solaris 9 adds large-page DISM; improves performance
  - Be certain that oradism is correctly configured
- System page fragmentation leading to slow instance start and stop operations
  - Fixes (and related tuning) in recent Solaris patches
- Overuse of filesystem cache, such as with VxFS “Cached Quick I/O (CQIO)”
  - If it hurts when you do that - don't do that

Wrapup

Rx for Scalability

- Modern system hardware components
- Modern OS release - Solaris 9 or later
  - Recent Solaris patch levels
    - especially for high CPU-count systems!
    - especially where recent enhancements are high-payoff!
- Oracle's “big knobs” settings
  - SGA size
  - I/O concurrency controls
- Purposeful filesystem and feature usage
  - avoid segmap and single-writer locks as needed
- Purposeful network configuration and tuning
- And ... don't forget ... fix that application code!

References

- "Avoiding Common Performance Issues Scaling Oracle on Sun Servers"
  - http://www.sun.com/blueprints/0303/817-1781.pdf by Glenn Fawcett
  - http://www.sun.com/blueprints/0303/817-2196.pdf (Appendix to paper)
- “Sun/Oracle Best Practices"
  - (Aging) http://www.sun.com/blueprints/0101/SunOracle.pdf by Bob Sneed
- “Performance Forensics”
  - http://www.sun.com/blueprints/1203/817-4444.pdf by Bob Sneed
- "Oracle I/O: Supply and Demand"
  - Sun User's Performance Group (SUPerG) whitepaper by Bob Sneed
- "To 100% - and Beyond!"
  - Sun User's Performance Group (SUPerG) whitepaper by Bob Sneed

? . ? ? ? ! !

Q&A

. . ? ? . ! ??