



Demystifying High-End Server Behavior when Scaling Database Applications

Glenn.Fawcett@Sun.com

Strategic Applications Engineering

Sun Microsystems, Inc.

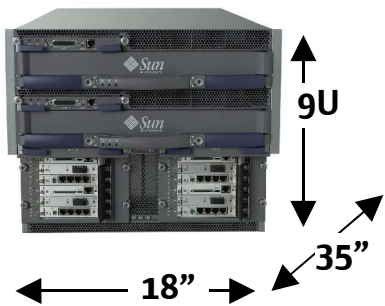


Overview of Presentation

- Review Architecture
- Discuss “Real” Differences
- Clarify Misunderstood Statistics
- Basics of Mastering Growth
 - HW, OS, DB, and Application
- Final Thoughts

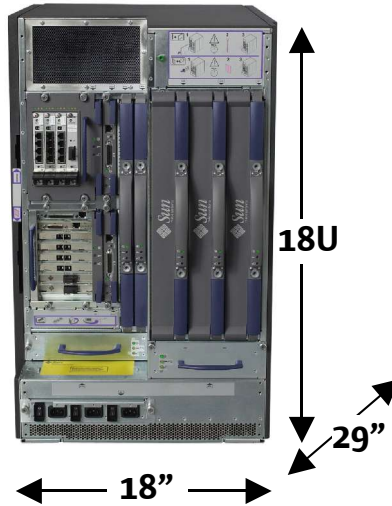
Sun Fire Architectural Review

Sun Fire 3800



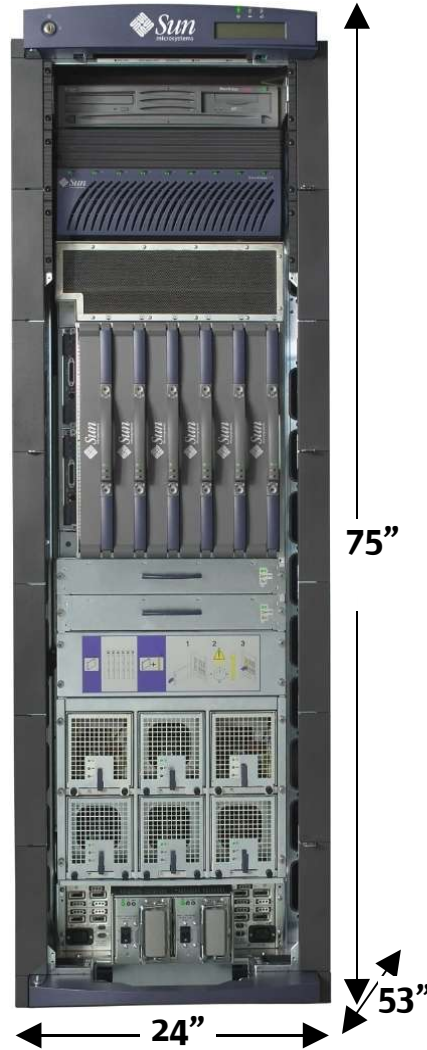
8 CPUs +
8 mem units + 4
I/O ctrls

Sun Fire 4800/4900



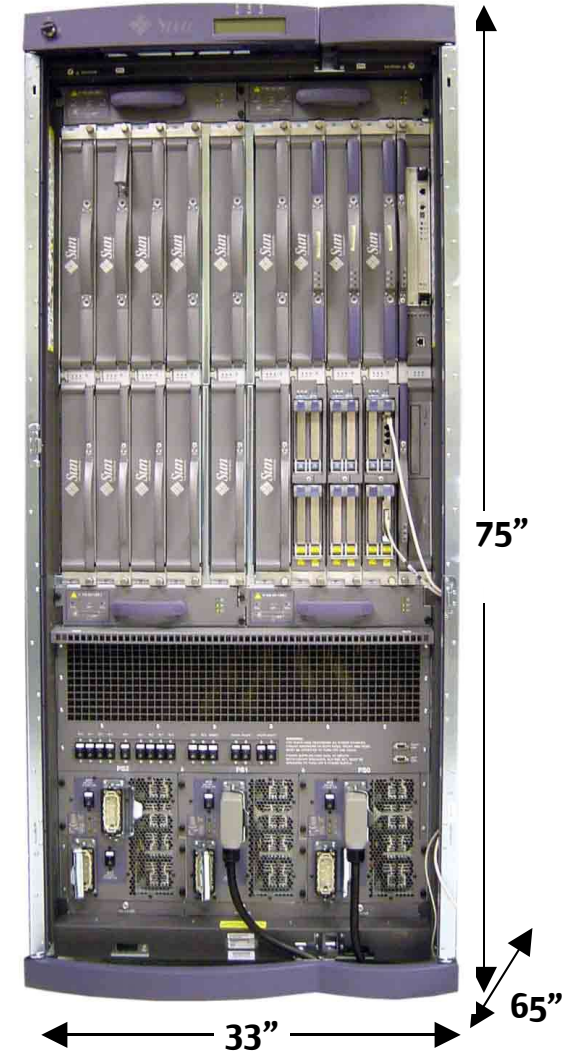
12 CPUs + 12 mem units
+ 4 I/O ctrls

Sun Fire 6800/6900



24 CPUs + 24 mem units +
8 I/O ctrls

Sun Fire 12k-15k / 20K-25K



72 CPUs + 72 mem units
+ 36 I/O controllers

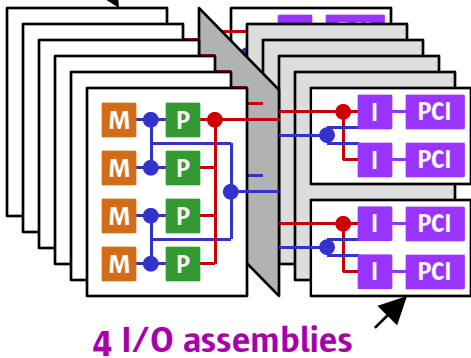
Architectural Review

- High-End Servers support
 - Up to 72 processors / 144 cores.
 - Over ½ TeraByte of MEMORY!
 - > 18GB/sec disk bandwidth over 96 channels.
- Interconnect Hierarchy required to scale to these levels.
- Design goals must be considered to understand performance.

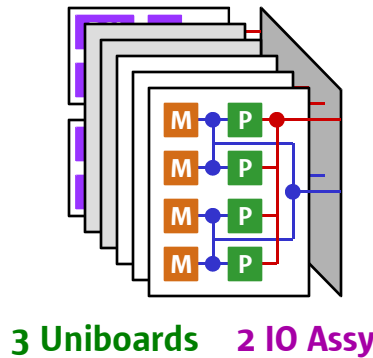
Sun Fire Family Interconnect

SF 6900

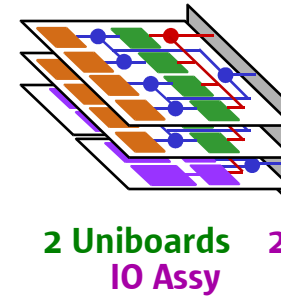
6 Uniboards



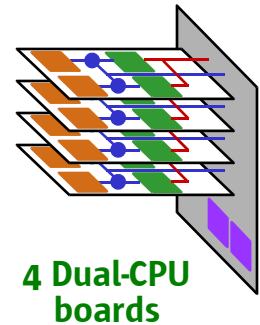
SF 4900



SF 3800

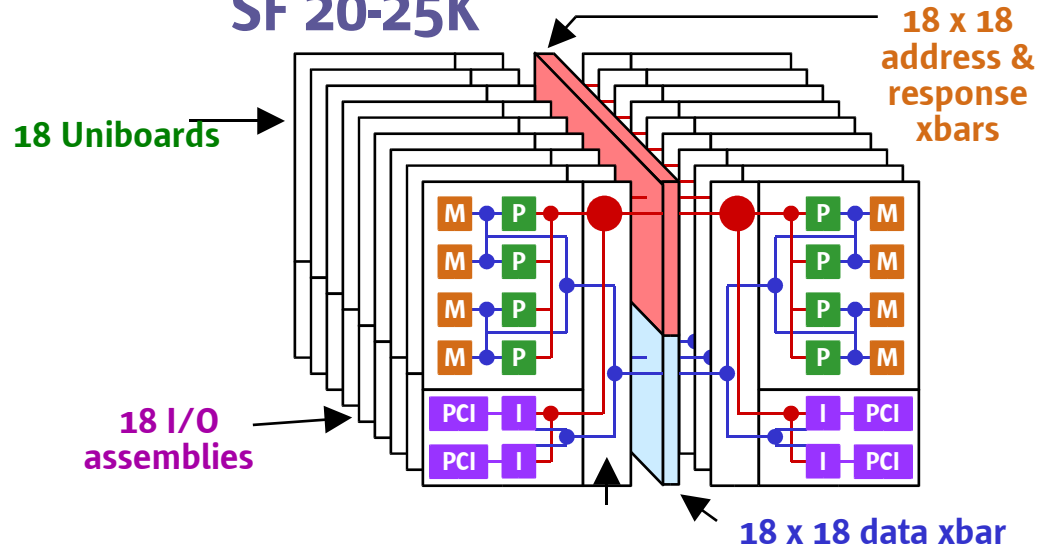


SF V880



Address path	—
Data path	—
Address repeater (+SSM)	●
Data switch	+
Processor	P
Memory	M
PCI bridge	I
PCI or cPCI card(s)	PCI

SF 20-25K



Architectural Differences

- Porsche vs. Semi-Truck
 - Fuel (Premium Gas vs Diesel)
 - Cargo Space (Overnight bag vs Pallets)
 - Performance criteria (mph vs pallets/hr)
- Choose the right vehicle for the application.

“Real” Architectural Differences

- Memory Size.
 - V490: 32GB max
 - E25K: 576GB max
- Memory Bandwidth and Latency
 - V890: 3.3GB/sec @ 252ns
 - E25K: 76.1GB/sec @ 248ns -> 468ns
- IO Throughput.
 - V890: < 1GB/sec
 - E25K: > 18GB/sec

“Real” Architectural Differences

- Scheduler Dispatch Table

- E15K/E25K has larger time-slice than smaller servers.

```
dispadm -g -c TS
```

- Kernel Cage

- Cage is enable by default on E15K/25K servers.
- Support for Domains and DR.

Misunderstood System Statistics

WaitIO = IDLE!!! nothing more :)

- WaitIO available via kstat interface.
 - Used by mpstat(1M) and iostat(1M)
 - Supposed to measure %wait due to IO.
- Doesn't measure what it is supposed to...
 - AsyncIO and SMP systems drive higher IO.
 - Systems capable of more IO tend to show more waitIO even if problems don't exist.
 - Doesn't measure all IO - only measures biowait()
- Use iostat(1M) to measure IO performance.

Misunderstood System Statistics

Cross calls “xcalls”

- Used by kernel to instruct a “specific” processor to execute a low-level function
 - Virtual memory translation, Signal processing, Dispatcher preemption, “/proc” thread control
- Larger systems can produce more simply due to increased throughput.
- Application config choices effect xcalls
 - File System vs db caching... more later
 - Persistent vs Transient connections...
Avoiding the exit storm

Scaling Challenges

Expanding your business

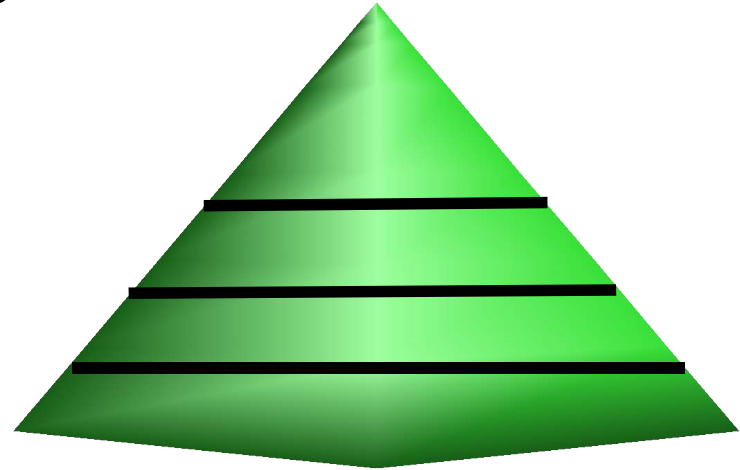
Two types of challenges

- Due to economies of scale
 - More Users
 - More Memory
 - More IO
 - More CPU
- Due to Architectural differences
 - Interconnect
 - Design choices

Building a Scalable environment

A good foundation for growth

- Focus on the core
 - Application
 - Database
 - OS
 - Hardware
- The Application IS the best place to effect performance, but often difficult.
- Best tuned applications still require a solid HW and OS combination.



Hardware choices

Building a scalable environment

- Use latest
 - FW revisions
 - AXQ
 - HPCI+
 - 66MHz slots vs 33MHz
- Build a balanced environment
 - Fully populate memory boards.. ie better to use lower density DIMMS across all boards then high density across ½ the slots.
 - Multiple HBA's across boards: avail and scale

Operating System Choice

Match OS to HW... Use at LEAST Solaris 9!!

- Solaris 8 is NOT optimal and EOL is near
 - Lacking large memory support..
No MPO, Large pages, lgroups, dtrace...
 - Less than optimal networking stack
 - Solaris 8 EOLs Oct 2005!!
Latest fixes are not, as a rule, ported.. some impossible.
- Use the LAG Solaris Possible.
 - Solaris 9 = Solaris 8.1
 - Solaris Application Guarantee...
It will just work or we will eat our hat :)

Managing memory

Choices and trade-offs can have a severe performance impact!

- Memory page sizes
 - 8k vs 4M
- File System page cache
 - Segmap
 - Db buffering
- TLB management
- Caged kernel
- Board Density and population

Large Memory Pages

OS choice still critical

- 4MB memory pages for heap help reduce TLB miss issues.

`trapstat(1m)` can diagnose these problems.

- Multiple page size support MPSS only available in S9/S10.
- Oracle supports large heap for PGA via a hidden `init.ora` parameter

`_use_realfree_heap_pagesize_hint=4194304`

Improved query performance by 14% on recent test.

Memory Fragmentation

OS choice still important

- Overuse of the “buffered” IO.
 - UFS page cache uses 8k pages
 - ISM shared memory segments use 4M pages to reduce tlb issues.
 - If no 4M pages exist, then 8k pages are coalesced.
 - Coalescing is an expensive operation, better in S9/S10.
- Can be avoided by mounting database filesystems direct or using raw partitions.

Filesystem Segmap scaling

Necessary if buffered IO is used heavily

- segmap_percent governs the amount of mappings that are prebuilt... default 12%
- Monitoring stats
 - hits/misses via kstat() interface
 - Amount of memory in pagecache
“::memstat” via mdb()
- Increase segmap_percent if a large pagecache is observed or if hits/miss are low... < 75%

Where do you Cache?

Database cache vs. UFS page cache

- 64bit databases have matured and are capable of scaling to 100+GB.
- Quick experiment w/ Oracle
 - 46GB table was populated via 100 processes.
 - 1st test: DB cache was set to 1GB and the database was mounted on a buffered file system.
 - 2nd test: DB cache was set to 50GB and the database was mounted direct.. not buffered.

Where do you Cache?

Database cache vs. UFS page cache

- More than 2x performance running with the DB cache vs FS cache!!!

<i>Cache</i>	<i>OS</i>	<i>Xcall/sec</i>	<i>rtime</i>	<i>Uusr</i>	<i>sys</i>
FS	S9	2,600,000	63	71	28
DB	S9	3,000	28	94	5
FS	S10	500	54	77	21

Veritas configuration choices

Unbuffered best...

- DB caching is best... as previously shown.
- Good choices:
 - VxVM with raw partitions is GOOD.
 - VxFS with ODM or QIO
- Avoid buffered VxFS or Cached QIO.
- Increase worker thread count!!
 - `vxiod set <# cpu cores>`

Solaris Scheduling

Fixed Priority Scheduler (FX) -- Solaris 9 min

- Default scheduler is Time Sharing
 - Priorities get adjusted according to CPU utilization
 - Time slice varies from 200 – 20 ms
 - Can cause excessive involuntary context switches
 - Not very efficient for batch workloads.
- Fixed priority scheduler (FX) allows equal equal scheduling and configurable time-slice.

Solaris Scheduling

Fixed Priority Scheduler (FX) cont..

- Use `priocntl(2)` to adjust database and application processes

```
# priocntl -s -c FX -m 59 -p 59 -t 1000 \  
-i pid <pid of db shadow>
```

Oracle Tuning

Buffer pool

- Statspack shows waits for reads with few other areas of contention.

Event	Waits	Time (s)	Ela Time
db file sequential read	582,754	6,662	43.25
CPU time		6,260	40.64
db file parallel write	32,127	1,303	8.46
SQL*Net more data to client	986,689	740	4.80
log file sync	9,335	224	1.46

- Hit ratio was 99.88%, but there was still 700+ reads/second.

Oracle Tuning

Buffer pool cont...

- Initial SGA size was 10G, but the system had 50GB of free memory :)
- Kept the default cache at 10G but added 40GB to the keep buffer pool.
- Statspack showed which queries were doing the most reads per transaction.

Oracle Tuning

storage parameters

- Candidate tables/indexes were altered to “Cache”.
- Storage parameters forced the object to be placed in the keep buffer pool.
- Physical reads/sec dropped 700 -> 7!

```
SQL> alter table PROCESSSKU cache;
```

```
SQL> alter table PROCESSSKU storage (buffer_pool keep);
```

Final Thoughts

Understanding behavior is the key to scaling

- Scaling requires a balance of CPU, Memory, and IO.
- Operating System selection critical to success... Solaris 9 a minimum!
 - MPO
 - MPSS
- IO and file system selection critical
 - DB caching is 2x as fast as Buffered IO caching.



QUESTIONS????

SUPerG DC, April 2005

Glenn.Fawcett@sun.com

<http://blogs.sun.com/glennf>