Design Considerations for Large-Scale Storage Solutions

September 24, 2008

Evans C. Harrigan

Principal Field Technologist

Sun Microsystems

Federal Division

Agenda

- Abstract
- Key Design Drivers
- Historical Perspectives
- Current System Technologies
- Achieving High Sustained I/O Performance
- The High Performance Data Transfer Architecture POC
- Acknowledgements
- Questions and Answers

Abstract

The design drivers for large-scale storage solutions have remained remarkably consistent over the years. Practitioners responsible for delivering data acquisition and storage solutions for systems with extreme bandwidth requirements must understand these drivers and their impact upon the design space. This talk will discuss the design drivers and how these can guide teams to practical solutions for extreme ingest, capture, and archive challenges.

Key Design Drivers - Hardware

- Balanced, high performance, parallel, scalable **computer system** architectures:
  - **CPU**: Very fast clock rate, robust floating point
  - **Memory**: Very high bandwidth for supporting fast linear & random access – includes caches
  - **I/O Infrastructure**: Very high DMA interface to memory; support for high performance industry-standard external interfaces

- Highly efficient industry-standard external peripheral **interfaces, networks and storage devices** that support high data streaming and high IOP rates:
  - **Channel adapters** (HBAs, HCAs, Network Interfaces, etc.)
  - High speed **mass storage** (rotational disks and solid state devices) and **linear sequential devices (tape)**

Key Design Drivers - Software

- Highly efficient, highly adaptable, open standards based **operating system** designs that support balanced, parallel, highly scalable computer system architectures:
  - Explicit support in kernel for high performance **asynchronous**, **direct** and **priority I/O**
  - **Fair share** and **fixed priority** (**real time?**) **scheduling**
  - Efficient **device drivers** for supporting physical and logical I/O devices
- Extensive, highly efficient support for all industry standard **I/O** and **networking protocols**, to ensure a broad family of I/O peripherals would be available for inclusion in the computer system architecture that support ultra high data streaming rates
  - Kernel and/or library support

Key Design Drivers - Software

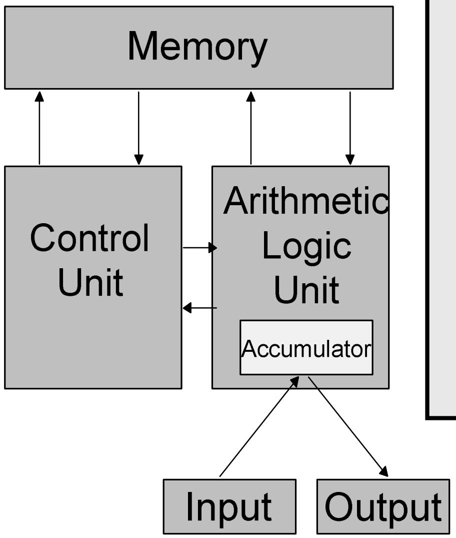
- Highly efficient, flexible **file system managers**, preferably open standards compliant, that support ultra high data streaming rates while providing **QoS attributes** such as **data integrity, high availability, recoverability** and **security**

- Highly efficient, flexible file **archive system managers**, preferably open standards compliant, that support high data streaming rates while providing **QoS attributes** such as **data integrity, high availability, recoverability** and **security**

- Highly efficient, **I/O performance-aware user application** designs

Historical Design Perspectives

- Von Neumann Architecture established formal design methodology for stored program computer (*Graphic from Wikipedia*)
  - I/O support implied



Von Neumann Architecture Enhancements

- Parallel access to Memory via banking and memory controllers
- Memory buffering, including caches (data, instructions and both)
- Virtual memory for "expanding" capacity

Historical Design Perspectives

- Seymour R. Cray understood the importance of I/O in end to end system performance
  - Significant enhancements to original Von Neumann Architecture:
    - Multiple pipelined instructions
    - Multiple banked memory for parallel access and higher bandwidth
    - Fast context switching (CPU instructions and program state registers)
    - Multi-channels for concurrent I/O
    - Extended buffer (non-executable) memory for higher I/O performance
    - Subsystems/special processors for off-loading I/O from CPU
    - Hardware I/O multiplexing for very low latency channel assignment scheduling
- Was 1996 Cray T932 the ultimate, balanced supercomputer?
    - **~1.75 GHz CPU Clock**
    - **900 GB/s memory B/W (peak)**
    - **~56 Gflops (peak)**
    - **32 GB/s I/O B/W (peak)**

Current System Technologies

- Hardware
  - Significant advances in **superscalar (RISC and CISC) microprocessors** (commodity & proprietary)
    - High CPU clock rates (> 3 GHz)
    - Multiple cores (2 – 8)
    - Moderate to large mult-level on- and off-chip caches
    - Extensive pipelining for instruction parallelism
    - Fast robust, multi-function FPUs
      - 4 floats/clock
      - 64-bits
    - Virtual memory

Current System Technologies

- Hardware
  - Significant improvements in **capacity of memory devices** (DRAM), but decrease in performance relative to faster increasing CPU clock rates
    - ***Canonical barrier to sustained end to end system performance: computations and I/O on large data applications***
  - Significant advances in peripheral devices: **storage and interfaces, including networking**
    - SCSI, Fibre Channel, SSD, SAS disk devices and controllers
    - Ethernet (1 Gb &10 Gb), Fibre Channel & IB Switches

Current System Technologies

- Software
  - Significant advances in:
    - **Unix based operating systems** design with strong adherence to open standards, support for parallelism, scalability, security and high availability:
    - **File system and Archiving Managers** that are open system compliant or aware:
      - GPFS
      - HPSS
      - Lustre
      - pNFS
      - SAM-QFS
      - StorNext
      - ZFS

Current System Technologies

- Software Advances, continued
  - Industry and open standards **I/O and Networking protocols**:
    - SCSI
    - Fibre Channel (2 Gb, 4 Gb and 8 Gb)
    - iSCSI
    - SAS
    - FCoE
    - Ethernet (1 Gb, 10 Gb)
  - Industry standards based **programming languages** and comprehensive **I/O libraries**
    - C
    - C++
    - Java (including real time)
    - Fortran 2008 (with Co-Arrays)

Achieving High Sustained I/O Performance

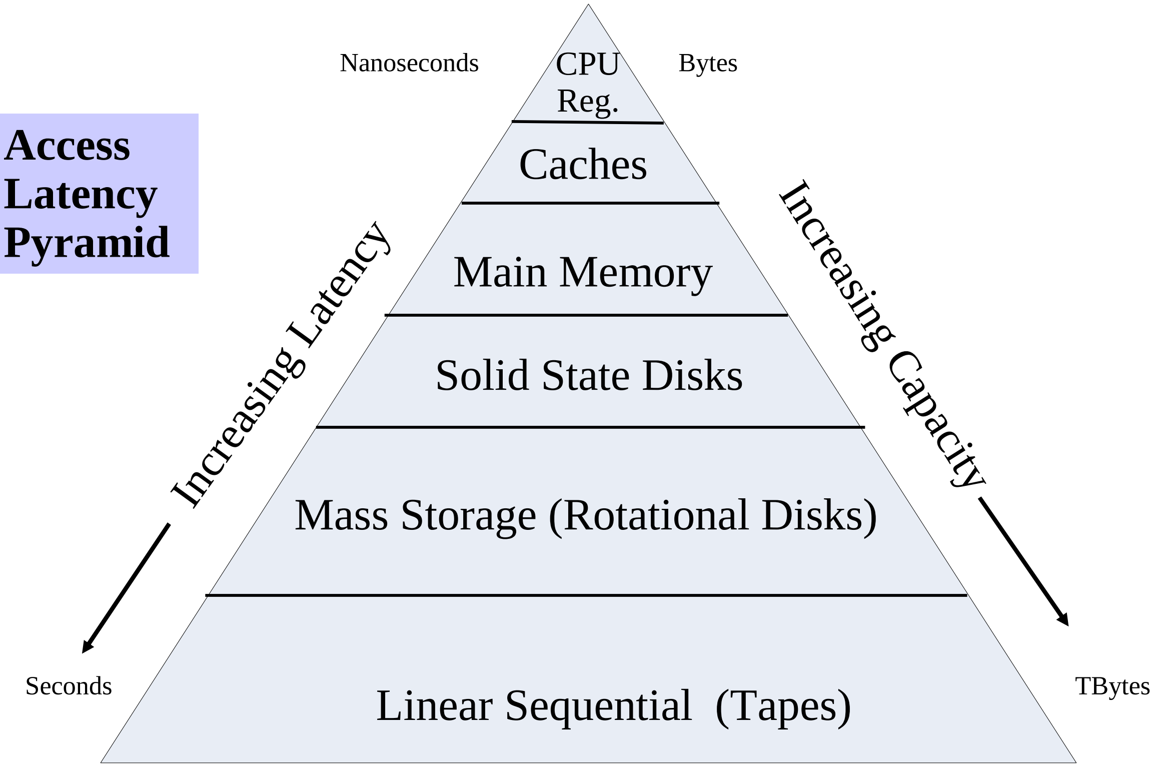
- Effectively address the key design drivers
  - Ideally develop a dynamical model of end to end system

1. Build and demonstrate a functional prototype
2. Update model with results from functional prototype execution
3. Repeat 1 – 3 until performance targets of model are achieved
4. Build scaled version of operational system and test as required

- Adhere strictly to access latency pyramid on Slide #14

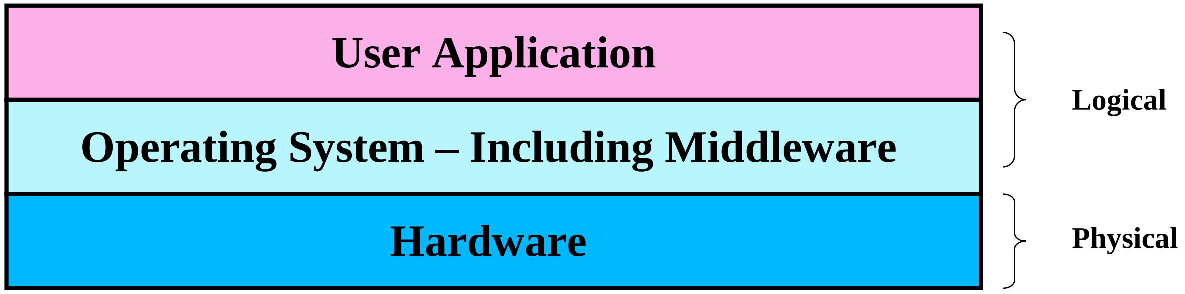
Achieving High Sustained I/O Performance

Access Latency Pyramid



Achieving High Sustained I/O Performance

- Decompose system architecture into Layers to reduce complexity



User Application

Operating System – Including Middleware

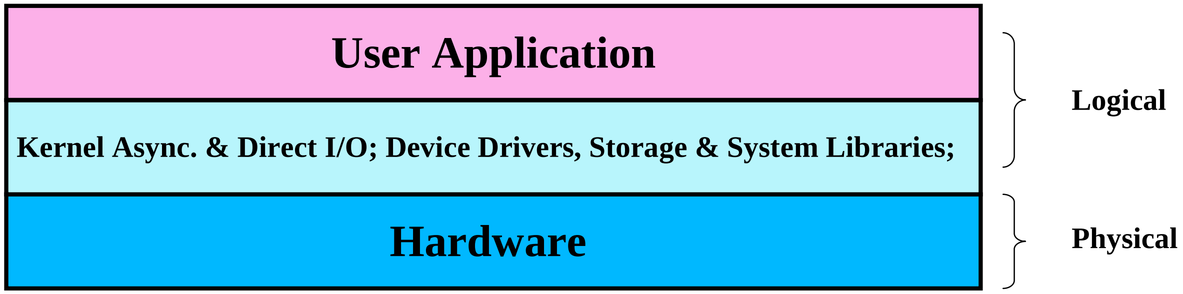
Hardware

Logical

Physical

Achieving High Sustained I/O Performance

- Must consider how layers integrate and how each layer impacts performance



User Application

Kernel Async. & Direct I/O; Device Drivers, Storage & System Libraries;

Hardware

Logical

Physical

- Concurrency & multithreading opportunities can exist at multiple layers too!

Achieving High Sustained I/O Performance

- Minimize Data Movement - number of times data is copied
  - System Buffered I/O (e.g., Unix I/O streams, memory-mapped I/O)
  - Object operators (e.g., C++)
  - Misaligned buffers (reads from, writes to)
- DMA directly to/from user data buffers or shared memory segments

Achieving High Sustained I/O Performance

- Implement Well-Formed I/O
  - Choose I/O sizes that are (small) integral multiples of the device stripe width
    - Stripe width typically determined by factors such as block size, sector size, RAID layout
    - Stripe width may vary by file system or device
- Perform I/O to and from page-aligned buffers
  - Otherwise libraries may perform extra data copy to aligned buffers
  - Pre-requisite for DMA, asynchronous I/O

Achieving High Sustained I/O Performance

- Understand and manage Client interactions
  - Multiple clients that simultaneously access the same device also impact performance
  - Consider mix of reads/writes, I/O sizes, rate of requests
  - Assess impact of misbehaved clients
    - Example: how does a client that performs small writes to a RAID-5 system impact all other clients?

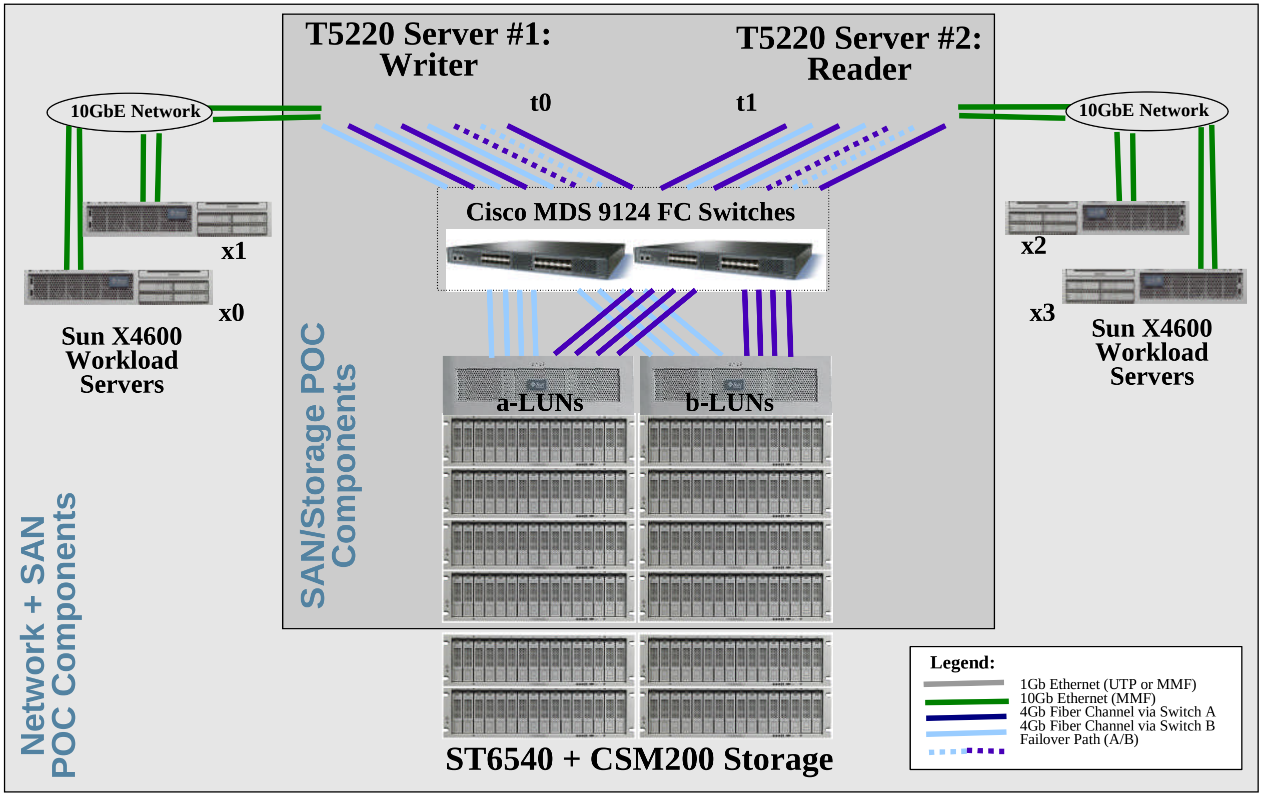
Achieving High Sustained I/O Performance

- Fully understand application's behavior!!
  - Generally, many applications expect streaming (sequential) I/O - GOOD!
  - Some applications require random access to high-speed devices and file systems – Not ideal for achieving highest rates in massive data ingest scenarios
    - Typically violate Well-Formed I/O principle
    - Smaller I/O requests
    - Misaligned buffers

HPDT Proof of Concept Benchmark

Sustained End to End Minimum 10 Gb/s
Sustained: From 10 Gb Ethernet Network Source
Through SAN
To 10 Gb Ethernet Network

HPDT Benchmark Configuration


T5220 Server #1: Writer
T5220 Server #2: Reader
t0
t1
10GbE Network
10GbE Network
Cisco MDS 9124 FC Switches
x1
x0
x2
x3
Sun X4600 Workload Servers
Sun X4600 Workload Servers
a-LUNs
b-LUNs
SAN/Storage POC Components
Network + SAN POC Components
ST6540 + CSM200 Storage
Legend:
1Gb Ethernet (UTP or MMF)
10Gb Ethernet (MMF)
4Gb Fiber Channel via Switch A
4Gb Fiber Channel via Switch B
Failover Path (A/B)

POC Hardware: Compute & Network

- T5220 Servers (2), each configured with:
  - 8-core UltraSPARC T2 processor
  - 32GB RAM
  - 4 Dual-Port FC 4Gb HBAs, Qlogic QLE2462 (PCI-Express)
  - 2 x 10Gb Ethernet ports per server (XAUI, Neptune PCIe)
- Workload Generation Servers
  - Sun X4600 platforms (4), four dual-core AMD processors each
  - Two Transmit @ 10Gbps+ via network to T5220 #1/SAN
  - Two Receive from T5220 #2/SAN
  - 2 x 10GbE ports per server
- Foundry IP Switches
  - 10Gb Ethernet switches, jumbo-frame support

POC Hardware: Storage/SAN

- Sun StorageTek 6540 arrays (2)
  - Two 4Gb Dual-Controller Arrays, __GB cache & 8 host-ports each
  - 12 CSM200 Drive Trays total, 6 per array (min ___ required)
  - 28 TB raw capacity: 16 x 146GB/15Krpm drives per tray
  - 21 TB usable RAID5 capacity (min ____ used for benchmark)
  - 3 x RAID5 (4+1) Data LUNs per tray, 256KB & 512KB seg sizes
  - Vertical-stripes RAID5 LUNs
  - RAID1 Metadata LUN (1 disk + mirror, one array only)
  - Global hot spare drives (minimum 1 per tray)
- Qlogic Host Bus Adapters (8)
  - Qlogic QLE2642 dual-port HBAs, 4Gbps
  - Each T5220 with 4 HBAs: 6 ports active, 2 standby/failover paths
- Cisco MDS 9124 FC Switches (2)
  - 32 Ports per switch, 4 Gbps FC, 4 zones per switch

POC Software

- Solaris 10 Update 4 (SPARC)
  - Kernel jumbo patch (KJP) level 127111-09
  - No MPXIO configured
  - Latest nxge 10GbE driver updates from KJP -09
- SAM-QFS 4.6 (patch 126512-03)
  - Host 1 = QFS/metadata server
  - Host 2 = shared client
  - Disk Allocation Units (DAU) = 2MB
  - 24MB file system i/o request sizes
  - Stripe = 1 (block-based round-robin??)
  - Testing with one and multiple stripe groups, 1-24 LUNs
  - Simultaneous Read-after-write to shared file from two hosts
  - Metadata on separate device (RAID1) from file system (RAID5)

POC Software (continued)

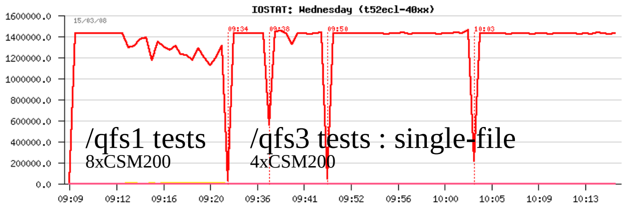
- QLogic
  - Native QLA driver and failover, version 5.03
  - Explicit HBA/LUN mappings and primary/secondary failover paths
- Test Suite
  - xdd
  - pdvt
  - ptp, ttcp, iperf for network tests
  - custom benchmark scripts & test programs
  - all applications leverage multithreading for throughput
- Management Software
  - SANsurfer CLI & Management GUI
  - Sun StorageTek Common Array Manager (CAM)
  - SANtricity 9.19 (LSI)
  - Solaris File System Manager (GUI) & QFS CLI

Benchmark Workloads

- Unit Test Performance
  - Single/Multiple LUNs
  - Disk trays
  - HBAs
  - 10GbE NICs
- SAN/Storage Throughput
  - Sustained writes, sustained reads (single host)
  - Read-after-Write using shared file (two hosts)
  - Single large-file
  - Multiple sequential files
  - Multiple files in parallel using concurrent processes
- Network + SAN “Edge Architecture”
  - Data transfer between 10GbE network and storage
  - Same workloads as SAN/Storage Tests

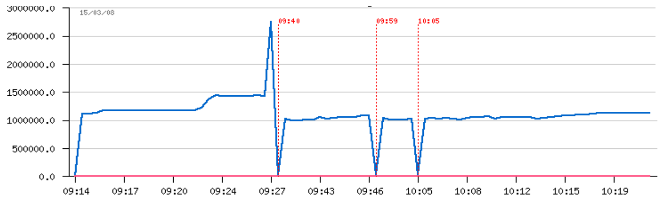
POC Results: Read-after-Write

1400 MBps
WRITE
Throttle=1400



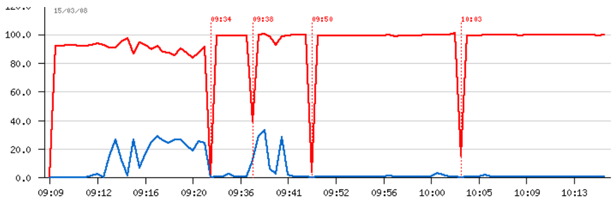
🟥 T0 host Write KB/s:
Sum of QFS volumes

1000 MBps
READ
Throttle=1100
(delayed startup)



🟦 T1 host Read KB/s:
Sum of QFS volumes

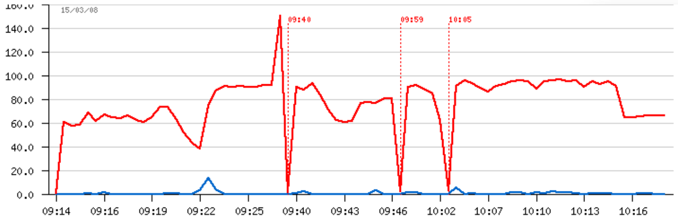
100% Busy
WRITE DISKS



🟥 T0 host
%Busy QFS disks

🟦 T0 host
%Wait QFS disks

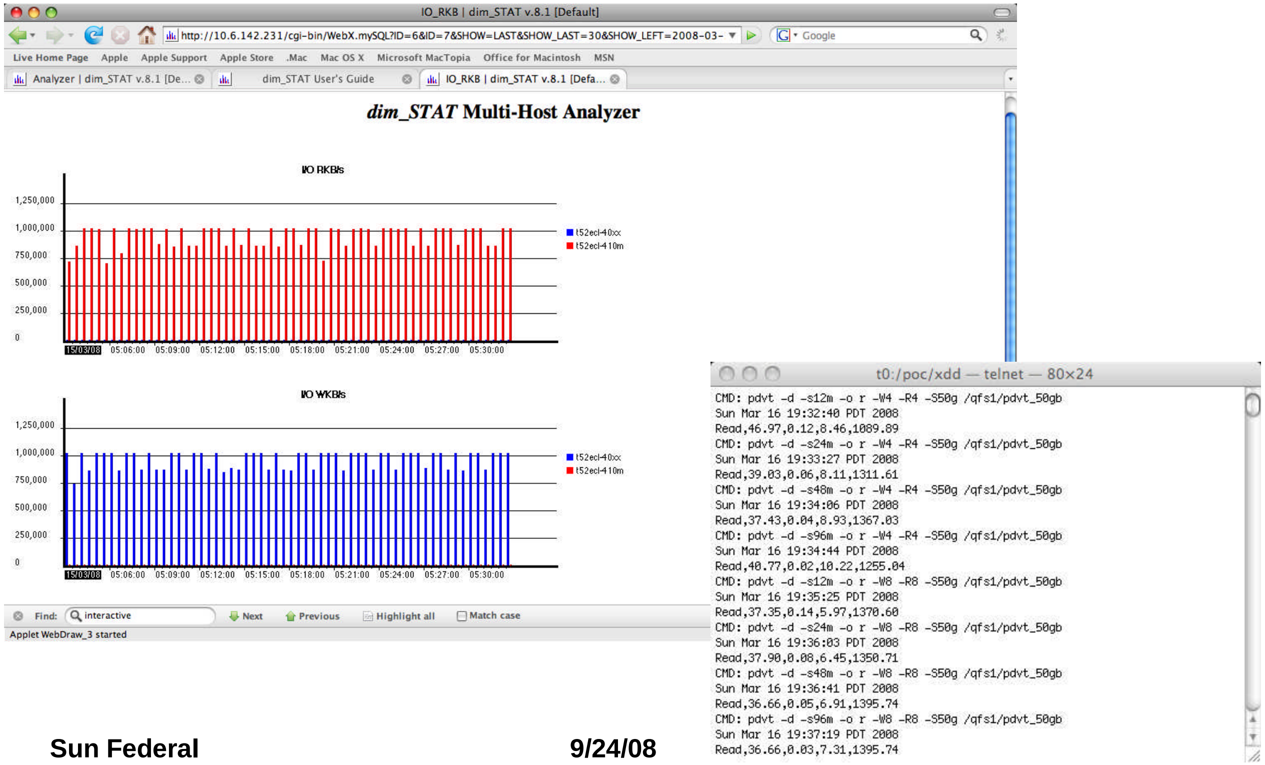
100% Busy
READ DISKS



🟥 T1 host
%Busy QFS disks

🟦 T1 host
%Wait QFS disks

POC Demo: Performance Testing


IO_RKB | dim_STAT v.8.1 [Default]
http://10.6.142.231/cgi-bin/WebX.mySQL?ID=6&ID=7&SHOW=LAST&SHOW_LAST=30&SHOW_LEFT=2008-03-
Google
Live Home Page
Apple
Apple Support
Apple Store
.Mac
Mac OS X
Microsoft MacTopia
Office for Macintosh
MSN
Analyzer | dim_STAT v.8.1 [De...
dim_STAT User's Guide
IO_RKB | dim_STAT v.8.1 [Defa...

dim_STAT Multi-Host Analyzer

```
CMD: pdvt -d -s12m -o r -W4 -R4 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:32:40 PDT 2008
Read,46.97,0.12,8.46,1089.89
CMD: pdvt -d -s24m -o r -W4 -R4 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:33:27 PDT 2008
Read,39.03,0.06,8.11,1311.61
CMD: pdvt -d -s48m -o r -W4 -R4 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:34:06 PDT 2008
Read,37.43,0.04,8.93,1367.03
CMD: pdvt -d -s96m -o r -W4 -R4 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:34:44 PDT 2008
Read,40.77,0.02,10.22,1255.04
CMD: pdvt -d -s12m -o r -W8 -R8 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:35:25 PDT 2008
Read,37.35,0.14,5.97,1370.60
CMD: pdvt -d -s24m -o r -W8 -R8 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:36:03 PDT 2008
Read,37.90,0.08,6.45,1350.71
CMD: pdvt -d -s48m -o r -W8 -R8 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:36:41 PDT 2008
Read,36.66,0.05,6.91,1395.74
CMD: pdvt -d -s96m -o r -W8 -R8 -S50g /qfs1/pdvt_50gb
Sun Mar 16 19:37:19 PDT 2008
Read,36.66,0.03,7.31,1395.74
```

Acknowledgements

- Sun Microsystems

Chuck Meyer	▪Sr. IT Architect
Sean Cochran	▪Principal Field Technologist
Larry McIntosh	▪Principal Field Technologist
Ron Emerick	▪Senior I/O Engineer
Kevin Colwell	▪Senior IT Architect
Carolyn Bumatay	▪Senior System Engineer

- The Boeing Company

Dennis Kuehn	▪Technical Fellow