Review: Storage System Issues

- Historical Context of Storage I/O
- Storage I/O Performance Measures
- Secondary and Tertiary Storage Devices
- A Little Queuing Theory
- Processor Interface Issues
- I/O & Memory Buses
- Show and Tell
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance
- RAID
- Tertiary Storage Possibilities

Review: I/O benchmarks

- Scaling to track technological change
- TPC: price performance as normalizing configuration feature
- Auditing to ensure no foul play
- Throughput with restricted response time is normal measure

ABCs of UNIX File Systems

- **Key Issues**
  - **File vs. Raw I/O**
  - **File Cache Size Policy**
  - **Write Policy**
  - **Local Disk vs. Server Disk**
- **File vs. Raw:**
  - **File system access is the norm: standard policies apply**
  - **Raw: alternate I/O system to avoid file system, used by data bases**
- **File Cache Size Policy**
  - **% of main memory dedicated to file cache is fixed at system generation (e.g., 10%)**
  - **% of main memory for file cache varies depending on amount of file I/O (e.g., up to 80%)**

ABCs of UNIX File Systems

- **Write Policy**
  - **File Storage should be permanent; either write immediately or flush file cache after fixed period (e.g., 30 seconds)**
  - **Write Through with Write Buffer**
  - **Write Back**
  - **Write Buffer often confused with Write Back**
    - » **Write Through with Write Buffer, all writes go to disk**
    - » **Write Through with Write Buffer, writes are asynchronous, so processor doesn't have to wait for disk write**
    - » **Write Back will combine multiple writes to same page; hence can be called Write Cancelling.**

ABCs of UNIX File Systems

- **Local vs. Server**
  - **Unix File systems have historically had different policies (and even file systems) for local client vs. remote server**
  - **NFS local disk allows 30 second delay to flush writes**
  - **NFS server disk writes through to disk on file close**
  - **Cache coherency problem if allow clients to have file caches in addition to server file cache**
    - » **NFS just writes through on file close, thereby invalidating local copies?**
    - » **Other file systems use cache coherency with write back to check state and selectively invalidate or update**

I/O Benchmarks

- **Alternative: self-scaling benchmark; automatically and dynamically increase aspects of workload to match characteristics of system measured**
  - **measures wide range of current & future**
- **Describe 3 self-scaling benchmarks**
  - **Transaction Processing: TPC-A, TPC-B, TPC-C**
  - **NFS: SPEC SFS (LADDIS)**
  - **Unix I/O: Willy**

I/O Benchmarks: TPC benchmarks

- TPC-A: version of DebitCredit
  - random vs. uniform arrival
  - smart terminals vs. dumb
  - 10 terminals per TPS vs. 100
  - 90% ≤2 seconds vs. 95% ≤1,
  - 1 branch record per TPS vs. 10
  - full disclosure, approved by TPC
  - complete TPS vs response time plots vs. single point
- TPC-B: same as TPC-A but without terminals: batch processing of requests
  - response time makes no sense: plots TPS vs. residence time (time of transaction resides in system)
- Other efforts underway on complex query processing (C) and decision support (D)

TPC results

TPC-A

Machine	*tpsA-local*	*K$/tps*	*OS/DB*	*Date*
HP 852S	43	24	HPUX 7/Infmx 4	12/90
VAX 4000	41	23	VMS 5.4/Dec 6	7/90
IBM RS6/550	32	20	Aix 3.1/infmx 4	1/91
Compaq SysPro	172	5	??	1/93
SPARCserve41	108	7	??	1/93
HP 9000 890/4	710	8	??	1/93

TPC-B

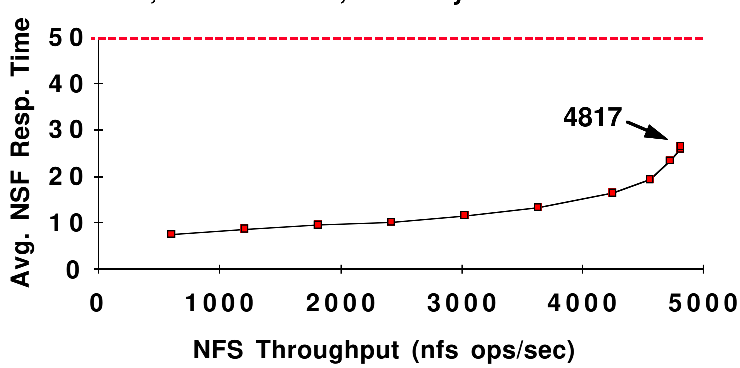
Machine	*tpsB*	*K$/tps*	*OS/DB*	*Date*
HP 852S	90	5	HPUX 7/Infmx 4	12/90
IBM RS6/550	58	5	Aix 3.1/infmx 4	1/91
Sun SS 490	57	8	Sun4.1/Sybase 4	10/90
Sun SS 2	52	4	Sun4.1/Sybase 4	10/90
Sun SC2000/10	1400	?	Solaris2/Sybase ?	9/94

SPEC SFS/LADDIS

- **1993 Attempt by NFS companies to agree on standard benchmark: Legato, Auspex, Data General, DEC, Interpahse, Sun. Like NFSstones but**
  - **run on multiple clients & networks (to prevent bottlenecks)**
  - **same caching policy in all clients**
  - **Reads: 85% full block & 15% partial blocks**
  - **Writes: 50% full block & 50% partial blocks**
  - **Average response time: 50 ms**
  - **Scaling: for every 100 NFS ops/sec, increase capacity 1GB**
  - **Results: plot of server load (throughput) vs. response time & number of users**
    - » **Assumes: 1 user => 10 NFS ops/sec**

Example SPEC SFS Result: DEC Alpha

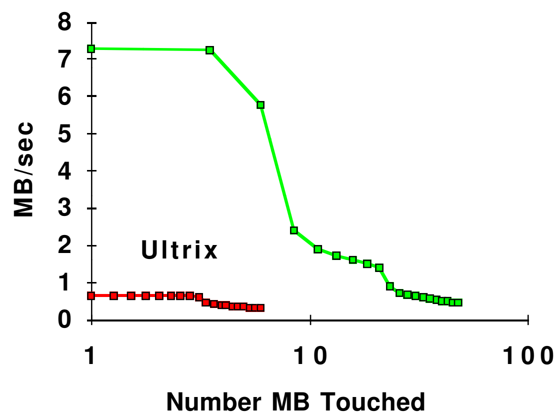
- **200 MHz 21064: 8KI + 8KD + 2MB L2; 512 MB; 1 Gigaswitch**
- **DEC OSF/1 v2.0**
- **4 FDDI networks; 32 NFS Daemons, 24 GB file size**
- **88 Disks, 16 controllers, 84 file systems**



Willy

- UNIX File System Benchmark that gives insight into I/O system behavior
- By Chen and Patterson, 1993
- Self scaling to automatically explore system size
- Examines 5 parameters
  - Unique bytes touched: ≈ data size; locality via LRU
    - » Gives file cache size
  - Percentage of reads: %writes = 1 – % reads; typically 50%
    - » 100% reads gives peak throughput
  - Average I/O Request Size: Bernoulli, C=1
  - Percentage sequential requests: typically 50%
  - Number of processes: concurrency of workload (number processes issuing I/O requests)
- Fix 4 parameters while vary 1 parameter
- Searches space to find high throughput

Willy: DS 5000 Number bytes touched



- Log Structured File System: effective write cache of LFS much smaller(5-8 MB) than read cache (20 MB) => reads cached while writes not => 3 plateaus

Sprite's Log Structured File System

Large file caches effective in reducing disk reads

Disk traffic likely to be dominated by writes

Write-Optimized File System

- **Only representation on disk is log**
- **Stream out files, directories, maps without seeks**

Advantages:

Speed

Stripes easily across several disks

Fast recovery

Temporal locality

Versioning

Problems:

Random access retrieval

Log wrap

Disk space utilization

UNIX File System Performance

- **8 Machines & OS**

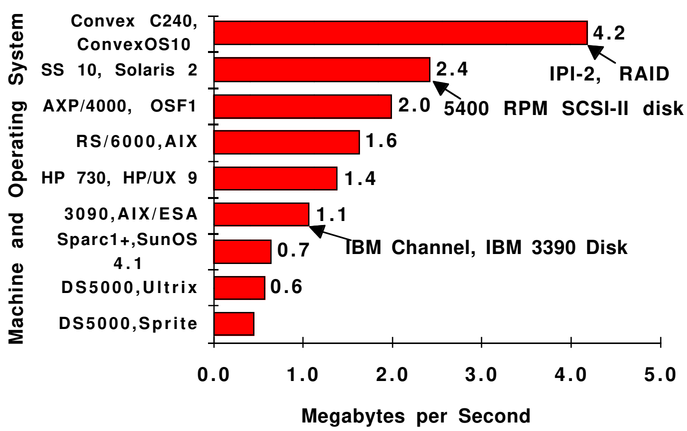
Machine	*OS*	*Year*	*Price*	*Memory*
Alpha AXP 3000/400	OSF/1	1993	$30,000	64 MB
DECstation 5000/200	Sprite LFS	1990	$20,000	32 MB
DECstation 5000/200	Ultrix 4.2	1990	$20,000	32 MB
HP 730	HP/UX 8 & 9	1991	$35,000	64 MB
IBM RS/6000/550	AIX 3.1.5	1991	$30,000	64 MB
SparcStation 1+	SunOS 4.1	1989	$30,000	28 MB
SparcStation 10/30	Solaris 2.1	1992	$20,000	128 MB
Convex C2/240	Convex OS	1988	$750,000	1024 MB
IBM 3090/600J VF	AIX/ESA	1990	$1,000,000	128 MB

Disk Performance

- I/O limited by weakest link in chain from processor to disk
- What is a fair comparison: disks, disk controller, I/O bus, CPU/Memory bus, CPU, OS?
- Common across machines?

Machine	*OS*	*I/O bus*	*Disk*
Alpha AXP 3000/400	OSF/1	TurboChannel	SCSI RZ26
DECstation 5000/200	Sprite LFS	SCSI-I	3 CDC Wren
DECstation 5000/200	Ultrix 4.2	SCSI-I	DEC RZ56
HP 730	HP/UX 8 & 9	Fast SCSI-II	HP 1350SX
IBM RS/6000/550	AIX 3.1.5	SCSI-I	IBM 2355
SparcStation 1+	SunOS 4.1	SCSI-I	CDC Wren IV
SparcStation 10/30	Solaris 2.1	SCSI-I	Seagate Elite
Convex C2/240	Convex OS	IPI-2	4 DKD-502
IBM 3090/600J VF	AIX/ESA	Channel	IBM 3390

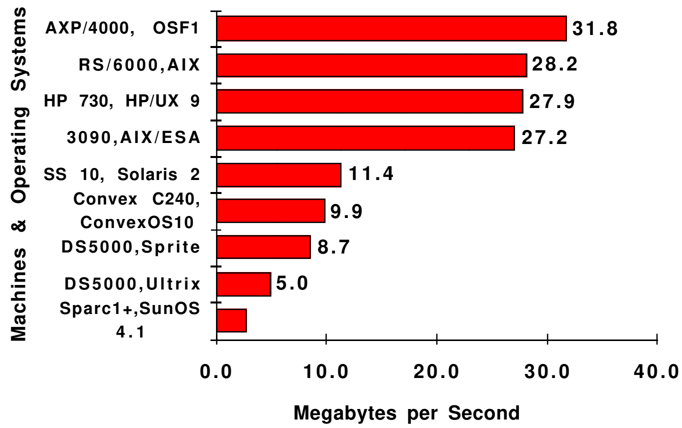
Disk Performance



- 32 KB reads
- SS 10 disk spins 5400 RPM; 4 IPI disks on Convex

File Cache Performance

- UNIX File System Performance: not how fast disk, but ***whether*** disk is used (32 KB reads; 7X
- 4X speedup between generations; DEC & Sparc



OS Policies and Performance

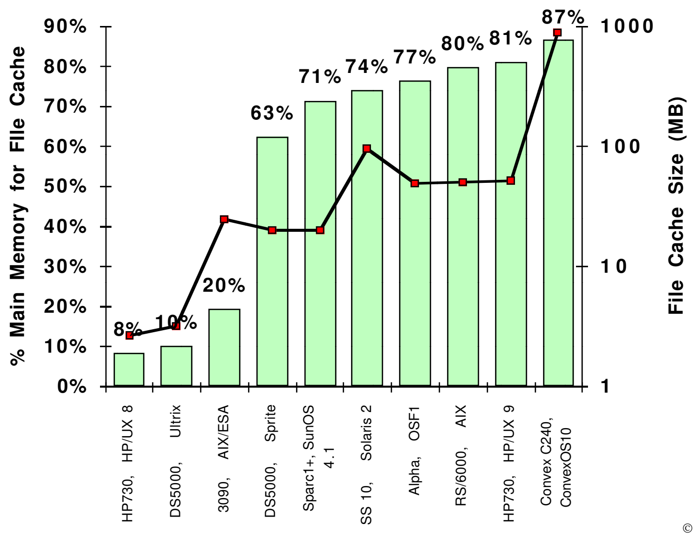
- Thus far determined by HW: DIsk, bus, memory system
- Policies on machines/OS aimed at same market

1) How much main memory allocated for file cache?

2) Can boundary change dynamically?

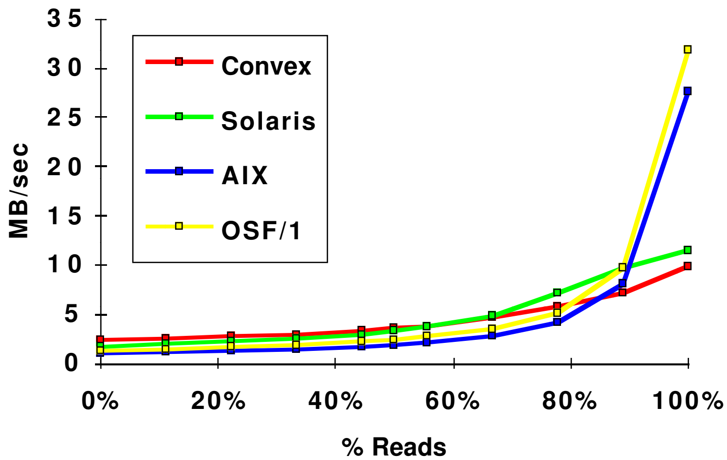
File Cache Size

- HP v8 (8%) vs. v9 (81%); DS 5000 Ultrix (10%) vs. Sprite (63%)



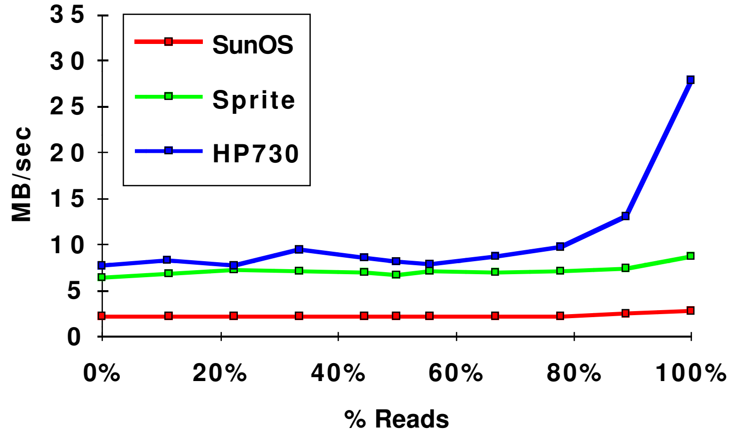
File System Write Policies

- Write Through with Write Buffer (Asynchronous): AIX, Convex, OSF/1 wt, Solaris, Ultrix



File System Write Policies

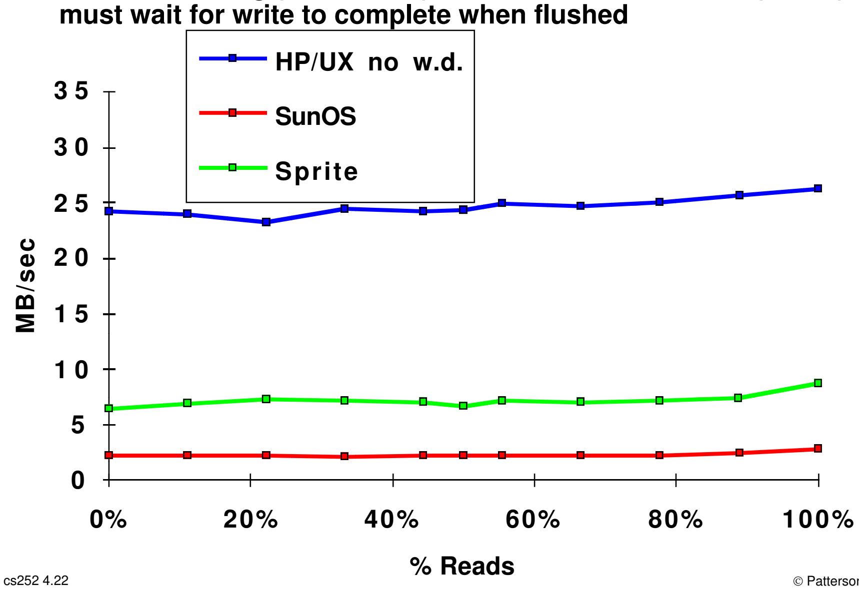
- Write Cancelling (Write Back): HP/UX, Sprite, OSF/1 w.c., Sun OS



- Why HP/UX goes up with reads?

File System Write Policies

- Write Cancelling (Write Back): HP/UX no write daemon (v. 30s); must wait for write to complete when flushed

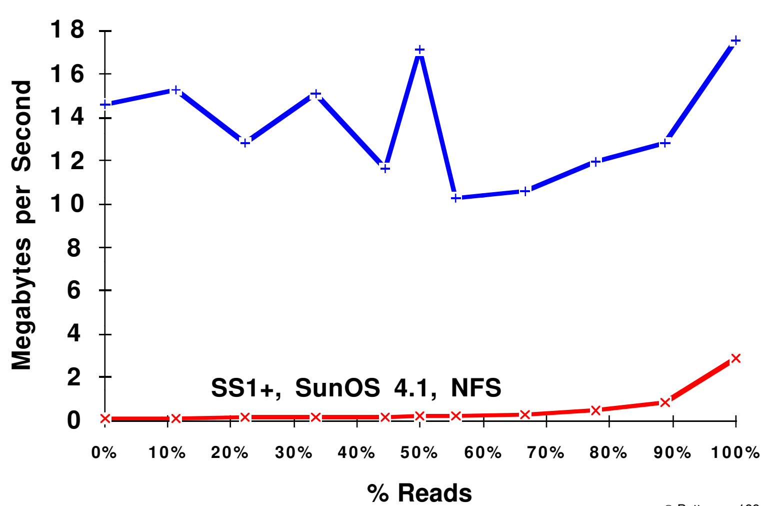


Will File Cache Blocks be Rewritten?

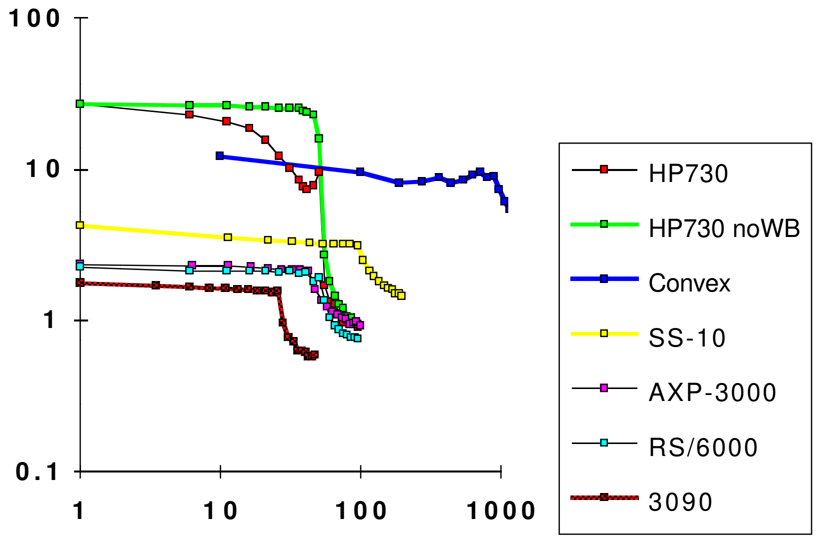
- 36% to 63% of all bytes do not survive 30 second window; 60% to 95% do not survive 1000 second window
- Short file lifetimes => blocks will be rewritten

Client Server

- NFS: write through when close (no buffers)
- HP DUX: client-lelve caching of writes; 25X faster @ 80% reads



Overall: Bandwidth vs. Bytes Touched



cs252 4.25

Summary: UNIX I/O

- HW determines potential performance, OS policies determine how much potential delivered
- File Cache performance improving rapidly: 4X in 3 years
- File cache performance on mainframes & minisupercomputers ≤ workstations
- Write cancelation (write buffer) improves file cache performance
- File cache policy (for machines aimed at same market) determine performance; 1st place to start

cs252 4.26