CMSC 411
Computer Systems Architecture
Lecture 19
Storage Systems, concl. & Multiprocessors

Alan Sussman
als@cs.umd.edu

Administrivia

- Homework #5 posted, due April 28
- Exam #2 next Thursday, April 23
  - practice exam posted later today
- Cache simulator project questions?

Designing an I/O System

Designing an I/O System

- Price, performance, and capacity issues
- Need to choose
  - which I/O devices to connect
  - how to connect them
- Example: The CPU is seldom the limiting factor for I/O performance
- Suppose the CPU can handle 10,000 I/O operations per second (IOPS)
- And suppose the average I/O size is 16 KB

I/O Systems

- **The other links in the I/O chain are:**
  - the *I/O controller* - suppose it adds 1 ms overhead per I/O operation
  - the *I/O bus* - suppose it is a bus that can transfer 20 MB/sec = 20 KB/ms
  - the *disk* - suppose it rotates at 7200 RPM, with 8 ms average seek time and 6 MB/sec transfer rate

I/O System Performance

- **Consider the disk time first:**
  - 7200 RPM = 7200/(60*10^3) = .12 revolutions per ms
  - 6 MB/sec = 6 KB/ms
  - So the average disk time is seek + rotational latency + transfer = 8 ms + .5 / .12 ms + 16 / 6 = 14.9 ms
- **So the average time per transfer is**
  - I/O controller time + bus time + disk time = 1 ms + 16 / 20 ms + 14.9 ms = 16.7 ms
- **So with one controller, one bus, and one disk, can do at most**
  - 1/(16.7*10^{-3}) = 60 IOPS
- **If this is not good enough, should analyze to see whether it is better to add more controllers, more buses, or more disks**
- **Another, more complex, performance analysis in Section 6.7, for the Internet Archive Cluster**

Storage Example: Internet Archive

- **Goal of making a historical record of the Internet**
  - Internet Archive began in 1996
  - Wayback Machine interface performs time travel to see what the website at a URL looked like in the past
- **It contains over a petabyte (10^{15} bytes), and is growing by 20 terabytes (10^{12} bytes) of new data per month**
- **In addition to storing the historical record, the same hardware is used to crawl the Web every few months to get snapshots of the Internet**

Internet Archive Cluster

- **1U storage node PetaBox GB2000 from Capricorn Technologies**
  - Contains 4 500 GB Parallel ATA (PATA) disk drives, 512 MB of DDR266 DRAM, one 10/100/1000 Ethernet interface, and a 1 GHz C3 Processor from VIA (80x86).
  - Node dissipates ≈ 80 watts
- **40 GB2000s in a standard VME rack, ⇒ 80 TB of raw storage capacity**
- **40 nodes are connected with a 48-port 10/100 or 10/100/1000 Ethernet switch**
- **1 PetaByte = 12 racks**

Estimated Cost

- VIA processor, 512 MB of DDR266 DRAM, ATA disk controller, power supply, fans, and enclosure = $500
- 7200 RPM Parallel ATA drive holds 500 GB = $375.
- 48-port 10/100/1000 Ethernet switch and all cables for a rack = $3000.
- Cost $84,500 for a 80-TB rack.
- 160 Disks are ≈ 60% of the cost

Estimated Performance

- 7200 RPM Parallel ATA drive holds 500 GB, has an average time seek of 8.5 ms, transfers at 50 MB/second from the disk. The PATA link speed is 133 MB/second.
  - performance of the VIA processor is 1000 MIPS.
  - operating system uses 50,000 CPU instructions for a disk I/O.
  - network protocol stack uses 100,000 CPU instructions to transmit a data block between the cluster and the external world
- ATA controller overhead is 0.1 ms to perform a disk I/O.
- Average I/O size is 16 KB for accesses to the historical record via the Wayback interface, and 50 KB when collecting a new snapshot
- Disks are limit: ≈ 75 I/Os/s per disk, 300/s per node, 12000/s per rack, or about 200 to 600 Mbytes / sec Bandwidth per rack
- Switch needs to support 1.6 to 3.8 Gbits/second over 40 Gbit/sec links

Estimated Reliability

- CPU/memory/enclosure MTTF is 1,000,000 hours (x 40)
- PATA Disk MTTF is 125,000 hours (x 160)
- PATA controller MTTF is 500,000 hours (x 40)
- Ethernet Switch MTTF is 500,000 hours (x 1)
- Power supply MTTF is 200,000 hours (x 40)
- Fan MTTF is 200,000 hours (x 40)
- PATA cable MTTF is 1,000,000 hours (x 40)
- MTTF for the system is 531 hours (≈ 3 weeks)
- 70% of time failures are disks
- 20% of time failures are fans or power supplies

Conclusions - Fallacies

- Disks never fail
  - a mean time to failure (MTTF) for one disk of 1.2M hours, or 140 years, computed by running thousands of disks for a few months, then counting the number that failed
  - but a more useful measure is the % of disks that fail in a given time period (e.g., 5 years), computed as #failed disks/total #disks
    - » where # failed disks = #disks * (#hours/disk)/MTTF

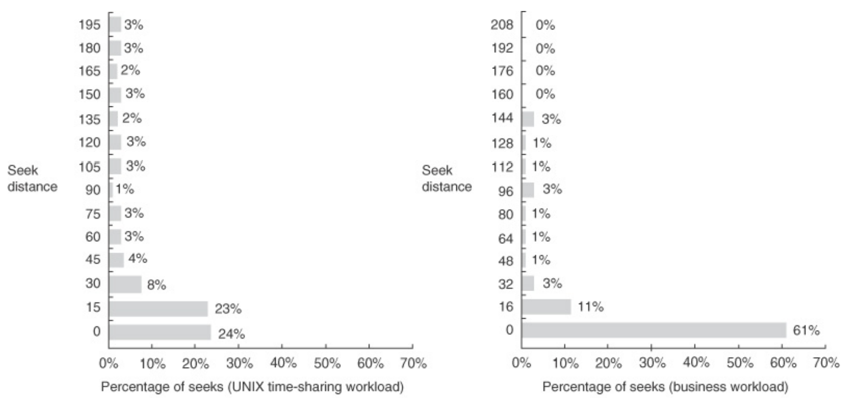
Conclusions - Fallacies

- **Computer systems can achieve 99.999% availability**
  - that's 5 minutes per year downtime, and highly unlikely in your environment
  - in 2001, well managed servers typically available 99% to 99.9% of time
- **DRAM will replace disks in desktop and server machines**
  - disk manufacturers have pushed the rate of technology improvement in disks to match or exceed that of DRAM
  - instead of DRAMs killing disks, disks are killing tapes

Conclusions - Fallacies

- **Average disk seek is for a seek of 1/3 of the cylinders**
  - just a rule of thumb for seeking from one random location to another random location on a different cylinder, assuming a large number of cylinders
  - problems with that rule are that
    - » seek time is not linear in distance (mechanical issues)
    - » there is locality to disk accesses

Conclusions - Fallacies – Fig. 6.24



© 2007 Elsevier, Inc. All rights reserved.

Multiprocessors - Snooping Caches and Directory-Based Multiprocessors

Outline

- Coherence
- Write Consistency
- Snooping
- Building Blocks
- Snooping protocols and examples
- Coherence traffic and Performance on MP
- Directory-based protocols and examples
- Conclusion

Challenges of Parallel Processing

1. Application parallelism ⇒ primarily via new algorithms that have better parallel performance
2. Long remote latency impact ⇒ both by architect and by the programmer

- For example, reduce frequency of remote accesses either by
  - Caching shared data (HW)
  - Restructuring the data layout to make more accesses local (SW)
- Start now on HW to help latency via caches

Symmetric Shared-Memory Architectures

- From multiple boards on a shared bus to multiple processors inside a single chip
- Caches both
  - Private data that is used by a single processor
  - Shared data that is used by multiple processors
- Caching shared data reduces
  - latency to shared data,
  - memory bandwidth for shared data, and
  - interconnect bandwidth

⇒ but cache coherence problem