CS 61C: Great Ideas in Computer Architecture (Machine Structures)

Lecture 37: I/O: Disks

Guest Lecturer: Rohan Chitnis

http://inst.eecs.berkeley.edu/~cs61c/

Review

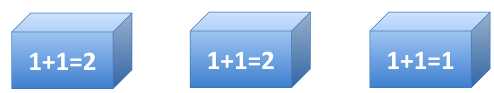
- I/O Devices: how humans + computers interact
  – Need to connect to, transfer with variety of devices
- Polling vs. Interrupts
  – Polling: processor continually asks device for updates
  – Interrupts: device interrupts processor with updates
- Exceptions are “unexpected” events
  – In MIPS, exceptions managed by System Control Coprocessor

Review - 6 Great Ideas in Computer Architecture

1. Layers of Representation/Interpretation
2. Moore’s Law
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement & Improvement
6. **Dependability via Redundancy**

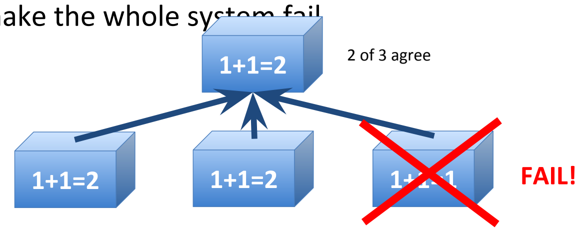
Review - Great Idea #6: Dependability via Redundancy

- Redundancy so that a failing piece doesn’t make the whole system fail



Review - Great Idea #6: Dependability via Redundancy

- Redundancy so that a failing piece doesn’t make the whole system fail



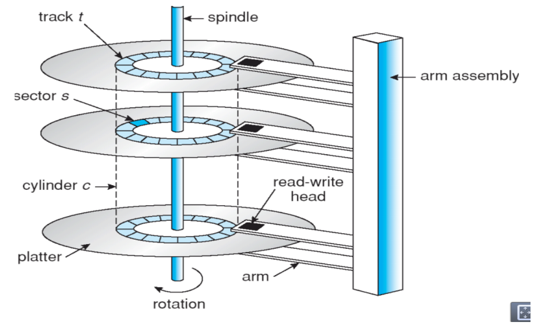
Review - Great Idea #6: Dependability via Redundancy

- Applies to everything from datacenters to memory
  – Redundant datacenters so that can lose 1 datacenter but Internet service stays online
  – Redundant disks so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID)
  – Redundant memory bits of so that can lose 1 bit but no data (Error Correcting Code/ECC Memory)

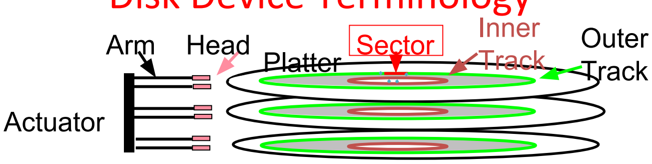
Magnetic Disk – common I/O device

- A kind of computer storage
  - Information stored by magnetizing ferromagnetic material on surface of rotating disk
    - Similar to tape recorder except digital rather than analog data
- Non-volatile storage
  - Data is retained even in the absence of power
- Purpose in computer systems (Hard Drive):
  - Long-term, inexpensive, large storage space for files
  - "Backup" for main memory (what if power goes out?)

Magnetic Disk Internals



Disk Device Terminology



- Several platters, with information recorded magnetically on both surfaces (usually)
- **Actuator moves head (end of arm) over track ("seek"), wait for sector to rotate under head, then read or write**
- **Head doesn't touch platter**

What about Flash Memory/SSDs?

- What is Flash Memory?
  - **Electronic**, non-volatile (no power ok) storage
  - Grid of transistors
    - Precise voltages applied to block the current from flowing through some transistors, generating pattern of 1s/0s
  - Benefits: durable (e.g., less sensitive to drops) & lower power
  - Limitations: **finite number of write cycles** (resistance builds in transistors, eventually can't be flipped → read-only)

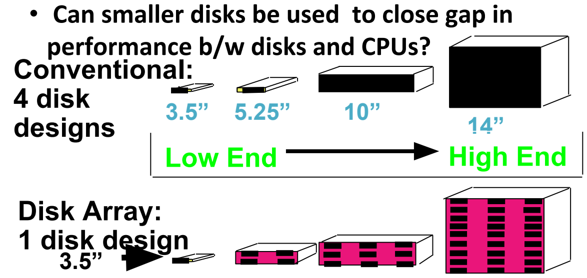
en.wikipedia.org/wiki/Flash_memory

What about Flash Memory/SSDs?

- So then what are SSDs?
  - Solid-State Drives
  - Another type of disk (long-term storage) that has NO moving parts, like magnetic disks had
  - Just an implementation (special use) of flash memory
  - Benefit: speed

Use Arrays of Small Disks…

- **Katz and Patterson asked in 1987:**
  - **Can smaller disks be used to close gap in performance b/w disks and CPUs?**



Replace Small # of Large Disks with Large # of Small!

(1988 Disks)

	IBM 3390K	IBM 3.5" 061
Capacity	20 GBytes	320 MBytes
Volume	97 cu. ft.	0.1 cu. ft.
Power	3 KW	11 W
Data Rate	15 MB/s	1.5 MB/s
I/O Rate	600 I/Os/s	55 I/Os/s
MTTF	250 KHrs	50 KHrs
Cost	$250K	$2K

Replace Small # of Large Disks with Large # of Small!

(1988 Disks)

	IBM 3390K	IBM 3.5" 061	← x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 I/Os/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays potentially high performance, high MB per cu. ft., high MB per KW, but what about reliability?

Disk Array Reliability

- Reliability - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks
  = Reliability of 1 Disk ÷ N
  (assuming failures independent)
  - Example: 50,000 hours ÷ 70 disks = 700 hours
- Disk system MTTF:
  Drops from 6 years to 1 month -- unreliable!

Redundant Arrays of (Inexpensive) Disks

- Files are spread out (segmented) across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed

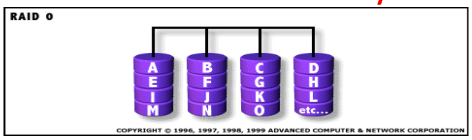
Redundant Arrays of (Inexpensive) Disks

- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store redundant info
  - Bandwidth penalty to update redundant info

RAID: Redundant Array of Inexpensive Disks

- Invented @ Berkeley (1989)
- A multi-billion dollar industry
  - 80% non-PC disks sold in RAIDs
- Can **parallelize** read/write accesses to/from separate disks
- Many different levels of RAID
  - 1 through 5 protect single disk failure

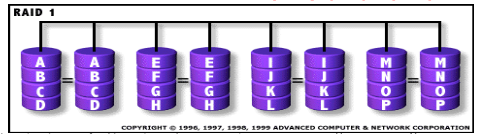
“RAID 0”: No redundancy = “AID”



- Striping: assume have 4 disks of data for this example
- Large accesses faster since transfer from several disks at once (parallelized)

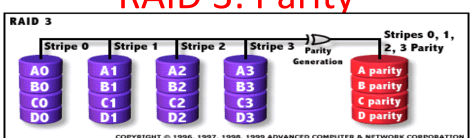
This and next 5 slides from `RAID.edu`, `http://www.acnc.com/04_01_00.html`
`http://www.raid.com/04_00.html` also has a great tutorial

RAID 1: Mirrored disks



- Each disk is fully duplicated onto its “mirror”
  - Very high availability can be achieved
- 1 logical write → 2 physical writes
- Most expensive solution: 100% capacity overhead

RAID 3: Parity

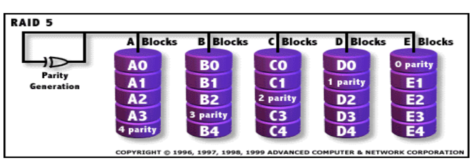


- Each sequential byte on a different drive
- Parity computed across disk group to protect against hard disk failures, stored in special “parity disk”
  - Logically, a single high-capacity, high-transfer-rate disk
- 25% capacity cost for parity in this example vs. 100% for RAID 1 (5 disks vs. 8 disks)

Drawbacks of RAID 3

- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk (access all disks)
  - Option 2: since P has old sum, compare old sum to new data, add the difference to P:
    1 logical write = 2 physical reads + 2 physical writes to 2 disks
- Parity Disk is bottleneck for small writes: Write to A0, B1 → both write to parity disk, cannot parallelize this!

RAID 5: Rotated Parity, faster small writes



- Independent writes possible because of interleaved parity
  - Example: write to A0, B1 uses disks 0, 1, 3, 4, so can proceed in parallel
  - Still 1 small write = 4 physical disk accesses

Peer Instruction

1. RAID 1 (mirror) and 5 (rotated parity) help with performance and availability
2. RAID 1 has higher cost than RAID 5
3. Small writes on RAID 5 are slower than on RAID 1

	123
A:	FFF
B:	FFT
B:	FTF
C:	FTT
C:	TFF
D:	TFT
D:	TTF
E:	TTT