Computer Science 146
Computer Architecture

Fall 2019

Harvard University

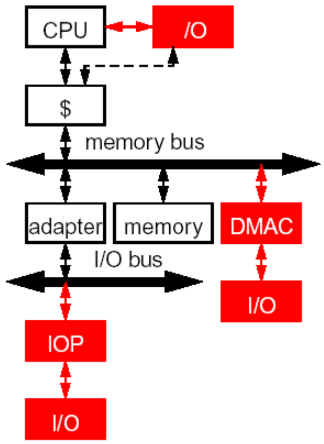
Instructor: Prof. David Brooks
dbrooks@eecs.harvard.edu

Lecture 22: More I/O

Lecture Outline

- HW5 and Project Questions?
- Storage and I/O
  - I/O Busses (7.3)
  - RAID (H&P 7.4-7.5)

I/O System Architecture

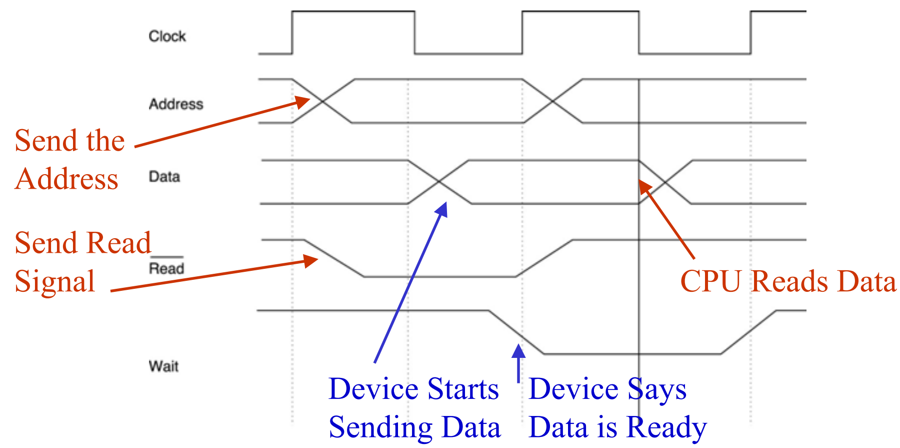


- Buses
  - Memory bus
  - I/O Bus
- I/O processing
  - Program controlled
  - DMA
  - I/O processors (IOPs)

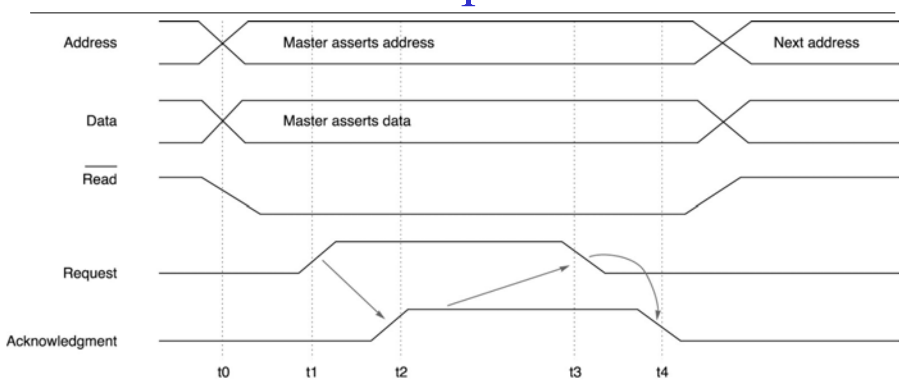
Bus Issues

- *Clocking*: is bus clocked?
  - Synchronous: clocked, short bus or slow clock => fast
  - Asynchronous: no clock, use "handshaking" instead => slow
  - Isochronous: high-bandwidth, packet-based system (uniform in time)
- *Switching*: When control of bus is acquired and released
  - Atomic: bus held until request complete => slow
  - Split-transaction: bus free between request and reply => fast
- *Arbitration*: deciding who gets the bus next
  - Overlap arbitration for next master with current transfer
  - Daisy Chain: closer devices have priority => slow
  - Distributed: wired-OR, low-priority back-off => medium
- Other issues
  - Split data/address lines, width, burst transfer

Synchronous Data Transfer: Read Operation



Asynchronous Data Transfer: Write Operation



t0: Master asserts lines
t1: Master waits and asserts Req
t2: Device asserts Ack (data recvd)
t3: Master releases Req
t4: Device releases Ack
(Handshake)

When to use?

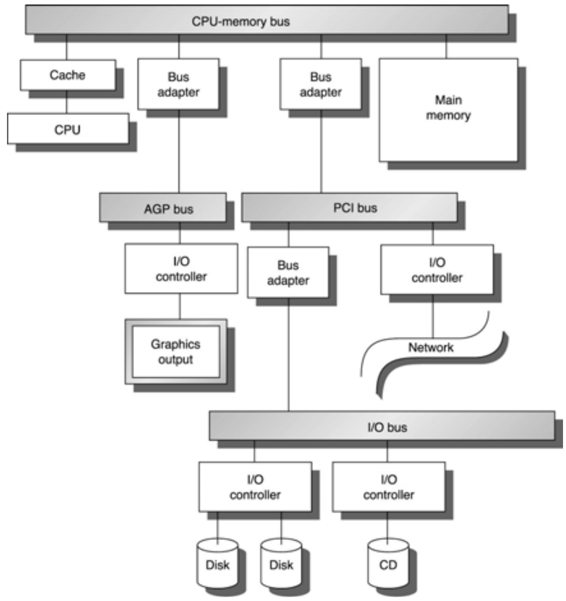
- When to use asynchronous vs. synchronous bus?
  - Mixed I/O speeds?
  - Bus length?
- Split transaction vs. atomic transaction?

I/O and Memory Buses

		Bits	MHz	Peak MB/s	Special Features
Memory Buses	Summit	128	60	960	
	Challenge	256	48	1200	
	XDBus	144	66	1056	
I/O Buses	ISA	16	8	16	Original PC Bus
	IDE/ATA	16	8-100	16-200	Disk, Tape, CD-ROM
	PCI	32 (64)	33 (66)	133-533	“Plug + Play”
	SCSI	8/16	5-160	10-320	High-level interface
	PCMCIA	8/16	8	16	Modems, “hot-swap”
	USB	Serial	A/Isoch.	1.5/60	Power line, packetized
	FireWire	serial	A/Isoch.	50/100	Fast USB

- Memory buses: speed (usually custom)
- I/O buses: compatibility (usually industry standard) + cost

Typical PC System Architecture



Who Does I/O?

- *Main CPU*
  - Explicitly executes all I/O operations
    - Memory Mapped I/O
    - Special ISA I/O Operations (x86, IBM 370)
  - Interrupt Driven, Polling Based, or Hybrid (realtime)
  - – High overhead, potential cache pollution
  - \+ But no coherence problems

Assist the Main CPU

- *I/O Processor (IOP or channel processor)*
  - (special or general) processor dedicated to I/O operations
  - \+ Fast
  - – May be overkill, cache coherency problems
    - I/O sees stale data on output (memory not up to date)
    - CPU sees stale data in cache on input (I/O system only updates memory)
- *DMAC (direct memory access controller)*
  - Can transfer data to/from memory given start address (but that's all)
  - \+ Fast, usually simple
  - – Still may be coherence problems, must be on memory bus

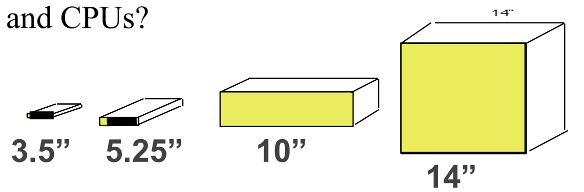
Communicating with I/O Processors

- Not issue if main CPU performs I/O by itself
- *I/O Control*: how to initialize DMAC/IOP?
  - Memory mapped: ld/st to preset, VM-protected address
  - Privileged I/O instructions
- *I/O Completion*: how does CPU know DMAC/IOP finished?
  - Polling: periodically check status bit => slow
  - Interrupt: I/O completion interrupts CPU => fast
- Q: *do DMAC/IOP use physical or virtual addresses?*
  - Physical: simpler, but can only transfer 1 page at a time
    - Pages in buffer may not be sequential pages in physical memory
  - Virtual: More powerful, but DMAC/IOP needs address translation info

Use Arrays of Small Disks?

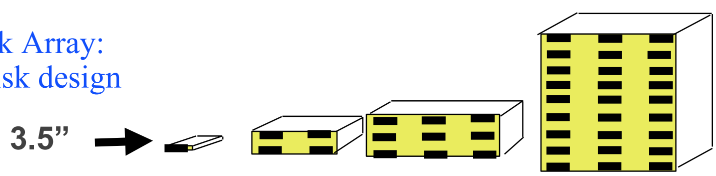
- Katz and Patterson asked in 1987:
  - Can smaller disks be used to close gap in performance between disks and CPUs?

Conventional:
4 disk designs

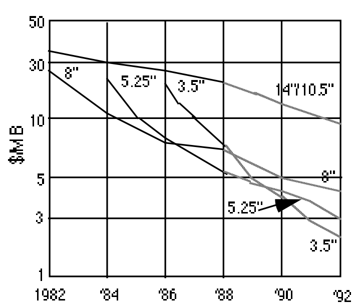


Low End → High End

Disk Array:
1 disk design

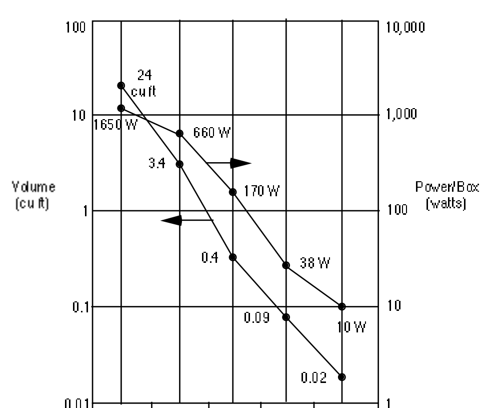


Advantages of Small Form Factor Disk Drives



Low cost/MB
High MB/volume
High MB/watt
Low cost/Actuator

Cost and Environmental Efficiencies



	14"	10.5"	8"	5.25"	3.5"
MB	7,500 F	1,320 F	1,000	765	200
MB/watt	4.54	2.0	5.88	20.13	20.0
MB/cu ft	313	388	2,500	8,500	10,000

Replace Small Number of Large Disks with Large Number of Small Disks!

	IBM 3390K	IBM 3.5" 0061	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 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

1988 Disk Drives vs. Disk Array

Disk Arrays have potential for large data and I/O rates, high MB per cu. ft., high MB per KW, but what about reliability?

Array Reliability

- Reliability of N disks = Reliability of 1 Disk ÷ N

  50,000 Hours ÷ 70 disks = 700 hours

  Disk system MTTF: Drops from 6 years to 1 month!

- Arrays (without redundancy) too unreliable to be useful!

Hot spares support reconstruction in parallel with access: very high media availability can be achieved

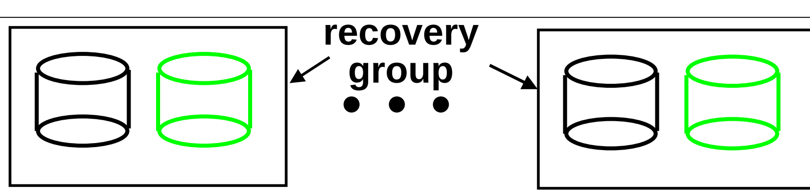
Redundant Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- 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

Redundant Arrays of Inexpensive Disks RAID 1: Disk Mirroring/Shadowing



- Each disk is fully duplicated onto its "mirror"

  Very high availability can be achieved
- Bandwidth sacrifice on write:

  Logical write = two physical writes
  - Reads may be optimized
- Most expensive solution: 100% capacity overhead
- Seek times can be optimized (choose disk with shortest seek)

(RAID 2 not interesting, so skip)

Redundant Array of Inexpensive Disks RAID 3: Parity Disk

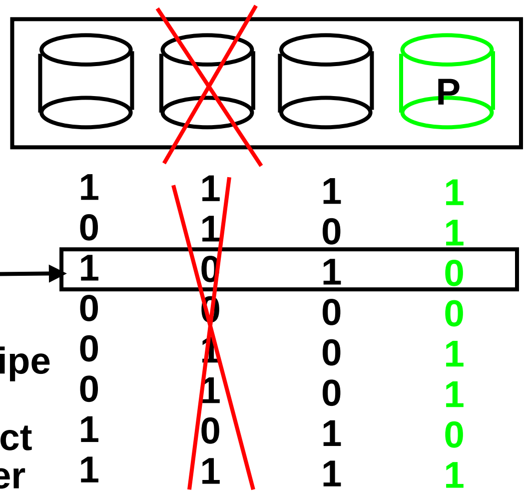
10010011
11001101
10010011
. . .

logical record

Striped physical records

P contains sum of other disks per stripe mod 2 ("parity")

If disk fails, subtract P from sum of other disks to find missing information



RAID 3

- Sum computed across recovery group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk: good for large transfers
- Wider arrays reduce capacity costs, but decreases availability
- 33% capacity cost for parity in this configuration

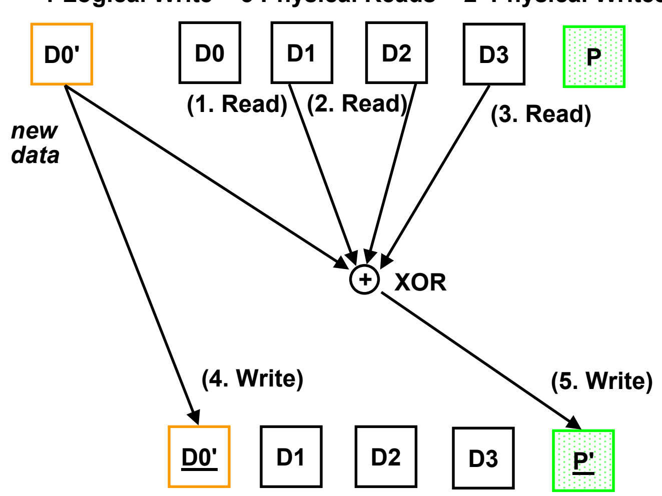
Inspiration for RAID 4

- RAID 3 relies on parity disk to discover errors on Read
  - Every access goes to all disks
  - Some apps want to do smaller accesses, allowing independent accesses to occur in parallel
  - Independent small *reads* are ok because disk can detect errors
    - Every sector has an error detection field
  - Independent small *writes* are trickier – don't we have to update parity?

Problems of Disk Arrays: Small Writes on RAID3

RAID-3: Small Write Algorithm

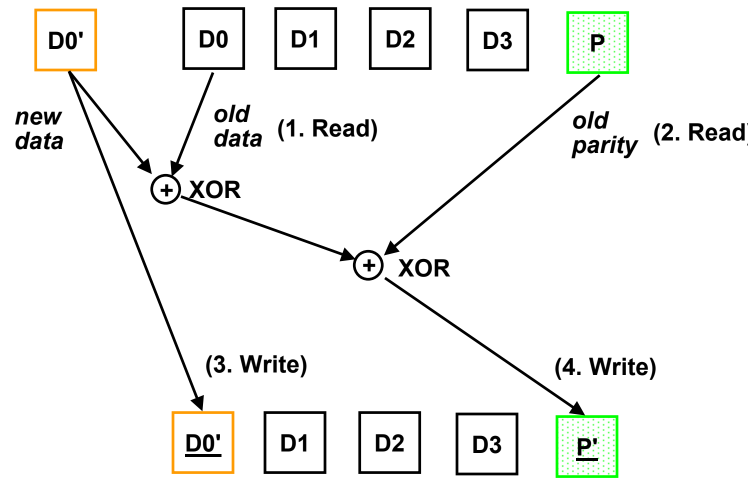
1 Logical Write = 3 Physical Reads + 2 Physical Writes



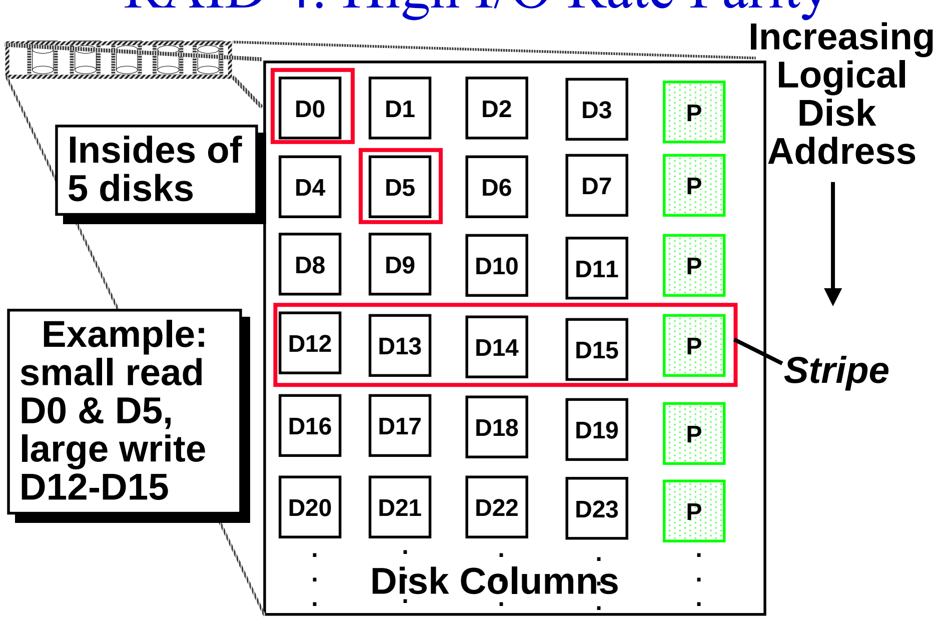
Problems of Disk Arrays: Small Writes on RAID4/5

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes

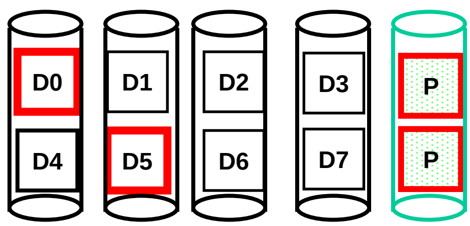


RAID 4: High I/O Rate Parity

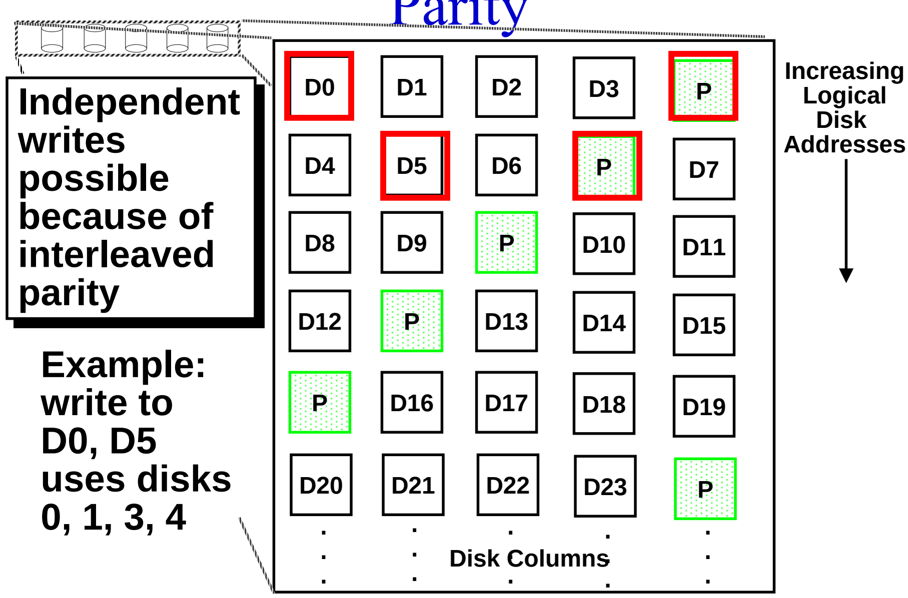


Inspiration for RAID 5

- RAID 4 works well for small reads
- Small writes (write to one disk):
  - RAID 3: read other data disks, create new sum and write to Parity Disk
  - RAID 4/5: since P has old sum, compare old data to new data, add the difference to P
- Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk (Parity Disk Bottleneck)



RAID 5: High I/O Rate Interleaved Parity



Berkeley History: RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software
- Today RAID is $19 billion dollar industry, 80% of non-PC disks sold in RAIDs

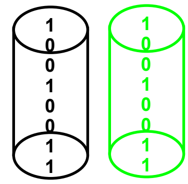
RAID in Industry

RAID Level	Name	Minimum Number of Disk Faults Survived	Example Data Disks	Corresponding Check Disks	Industry Use
0	Nonredundant Striped	0	8	0	Widely Used
1	Mirrored	1	8	8	EMC, IBM Compaq
2	Memory-style ECC	1	8	4	
3	Bit-interleaved Parity	1	8	1	Storage Concepts
4	Block-interleaved Parity	1	8	1	Network Appliance
5	Block-interleaved distributed parity	1	8	1	Widely Used
6	P+Q redundancy	2	8	2	

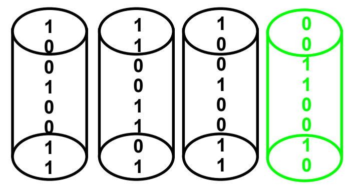
Summary: RAID Techniques

Goal: Performance, popularity due to reliability of storage

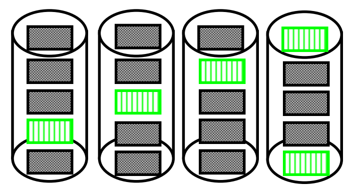
- ***Disk Mirroring, Shadowing (RAID 1)***
  - **Each disk is fully duplicated onto its "shadow"**
  - **Logical write = two physical writes**
  - **100% capacity overhead**



- ***Parity Data Bandwidth Array (RAID 3)***
  - **Parity computed horizontally**
  - **Logically a single high data bw disk**



- ***High I/O Rate Parity Array (RAID 5)***
  - **Interleaved parity blocks**
  - **Independent reads and writes**
  - **Logical write = 2 reads + 2 writes**



I/O System Example

- Given
  - 500 MIPS CPU
  - 16B wide, 100 ns memory system
  - 10000 instrs per I/O
  - 16KB per I/O
  - 200 MB/s I/O bus, with room for 20 SCSI-2 controllers
  - SCSI-2 strings (buses) – 20MB/s with 15 disks per bus
  - SCSI-2 1ms overhead per I/O
  - 7200 RPM (120 RPS), 8ms avg seek, 6MB/s transfer disks
  - 200 GB total storage
- Q: Choose 2GB or 8GB disks for maximum IOPS?
  - How to arrange disks and controllers?

I/O System Example (cont'd)

- Step 1: Calculate CPU, memory, I/O bus peak IOPS
  - CPU: 500 MIPS/(10000 instructions/IO) = *50000 IOPS*
  - Memory: (16-bytes / 100ns) / 16KB = *10000 IOPS*
  - I/O bus: (200 MB/s) 16KB = *12500 IOPS*
  - Memory bus (10000 IOPS) is the bottleneck
- Step 2: Calculate Disk IOPS
  - T_{disk} = 8ms + 0.5/120 RPS + 16KB/(6MB/s) = 15 ms
  - Disk: 1 / 15ms = *67 IOPS*
  - 8GB Disks => need 25 => 25 * 67 IOPS = *1675 IOPS*
  - 2GB Disks => need 100 => 100 * 67 IOPS = *6700 IOPS*
  - 100 2GB disks (6700 IOPS) are new bottleneck
- Answer: 100 2GB disks!

I/O System Example (cont'd)

- Step 3: Calculate SCSI-2 controller peak IOPS
  - T_{SCSI-2} = 1ms + 16KB/(20MB/s) = 1.8ms
  - SCSI-2: 1/1.8ms = *556 IOPS*
- Step 4: How many disks per controller?
  - 556 IOPS/ 67 IOPS = *8 disks per controller*
- Step 5: How many controllers?
  - 100 disks/ 8 disks/controller = 13 controllers
- Answer: 13 controllers, 8-disks each

Next Lecture

- Wednesday:
  - Google Cluster

  Reading:

    L. Barroso, J. Dean, and U. Holzle, "Web search for a planet: The Google Cluster Architecture," IEEE Micro, 23, 2, March-April 2003, pp. 22-28
  - Course Summary and Wrapup
  - Schedule a time for the Final Review