Lecture 9: EITF20 Computer Architecture

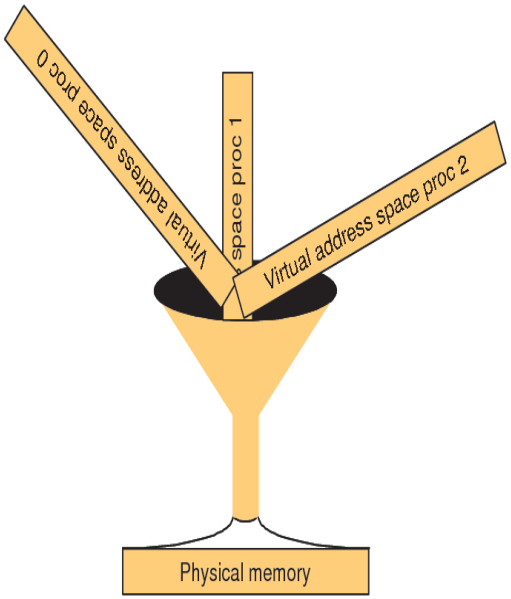
Anders Ardö

EIT – Electrical and Information Technology, Lund University

December 4, 2014

Outline

Virtual memory – why?

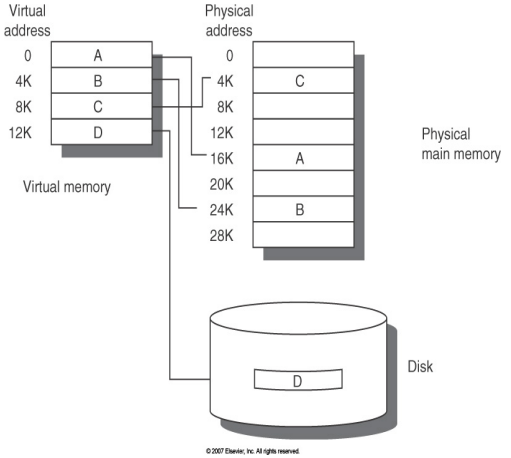


Reasons to use VM:

- Replaces overlays
- Large address space
- Several processes sharing the same physical memory
- Protection of memory
- Relocation

Virtual memory – concepts

Part of the memory hierarchy:

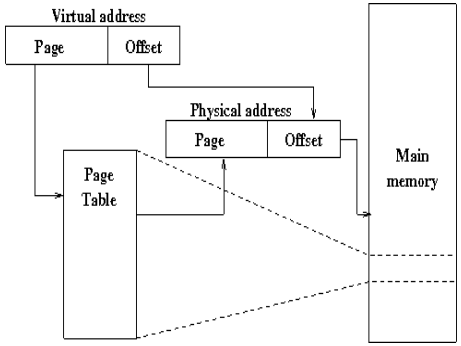


- The virtual address space is divided into ***pages***
- The physical address space is divided into ***page frames***
- A miss is called a ***page fault***
- Pages not in main memory are stored on ***disk***

- The CPU uses **virtual addresses**
- We need an **address translation** (memory mapping) mechanism

VM: Page identification

Use a **page table** stored in main memory:



- Suppose 4 KB pages, 32 bit virtual address, 4 bytes per entry
- Page table takes $\frac{2^{32}}{2^{12}} * 4 = 2^{22} = 4$ *Mbyte*
- 64 bit virtual address, 16 KB pages $\rightarrow$ $\frac{2^{64}}{2^{14}} * 4 = 2^{52} = 2^{12}$ *TB*
- Per process

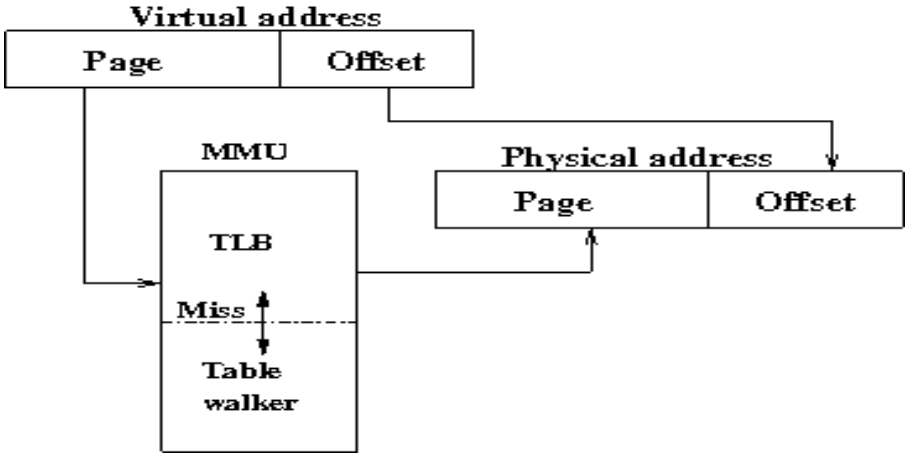
Solutions

- Multi–level page table
- (Inverted page table)

Fast address translation

How do we avoid two (or more) memory references for each original memory reference?

- Cache address translations – ***Translation Lookaside Buffer* (TLB)**



Summary memory hierarchy

Hide CPU - memory performance gap
Memory hierarchy with several levels
Principle of locality

Cache memories:	**Virtual memory:**
• Fast, small - Close to CPU	• Slow, big - Close to disk
• Hardware	• Software
• TLB	• TLB
• CPU performance equation	• Page-table
• Average memory access time	• Very high miss penalty $\Longrightarrow$ miss rate must be low
• Optimizations	• Also facilitates: relocation; memory protection; and multiprogramming

Same 4 design questions - Different answers

Questions!

QUESTIONS?

COMMENTS?

Take a step back

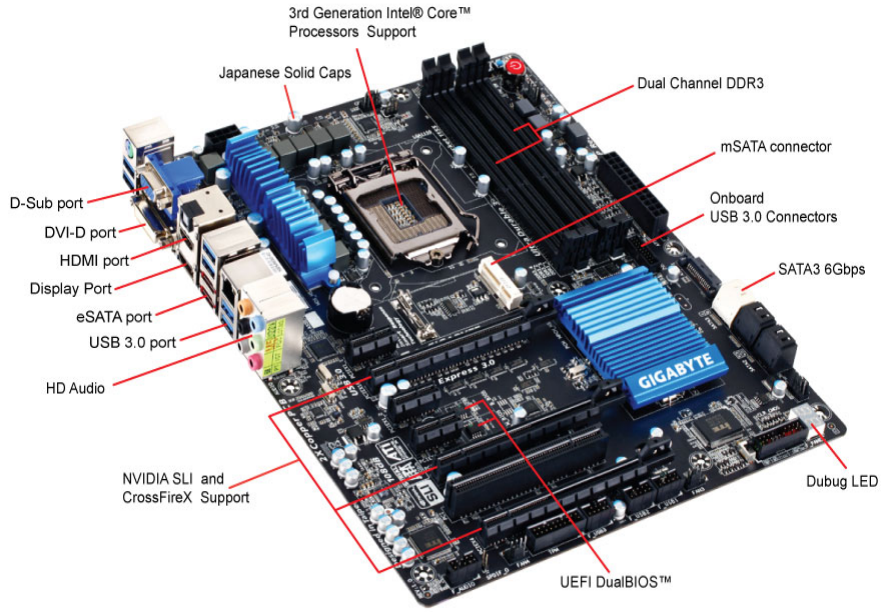
So far

- Performance, Quantitative principles
- Instruction set architectures, ISA
- Pipelining, ILP
- Memory systems, cache, virtual memory

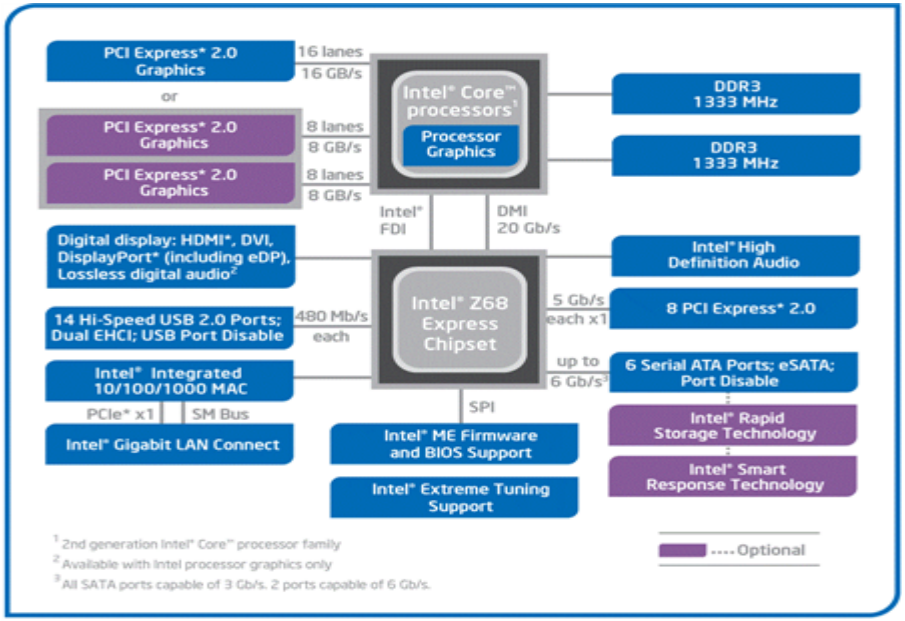
Coming

- Storage Systems, I/O
- (Case studies, special purpose, embedded)

Motherboard

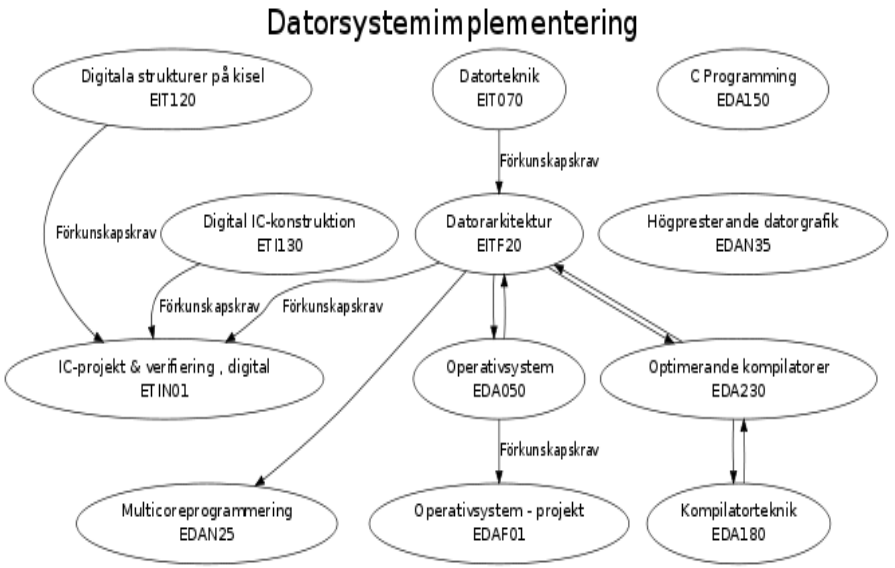


Chip-set architecture



Take a step forward

Computer Architecture course environment



Lecture 9 agenda

Chapters 6.1 - 6.4, 6.6 - 6.7 in "Computer Architecture"

Outline

Outside the processor and memory

Storage Systems

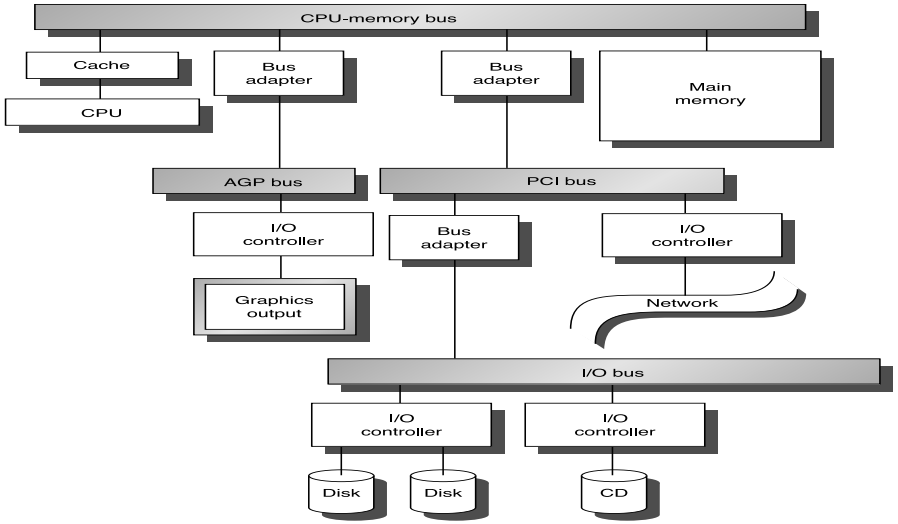
I/O

Who cares about I/O?

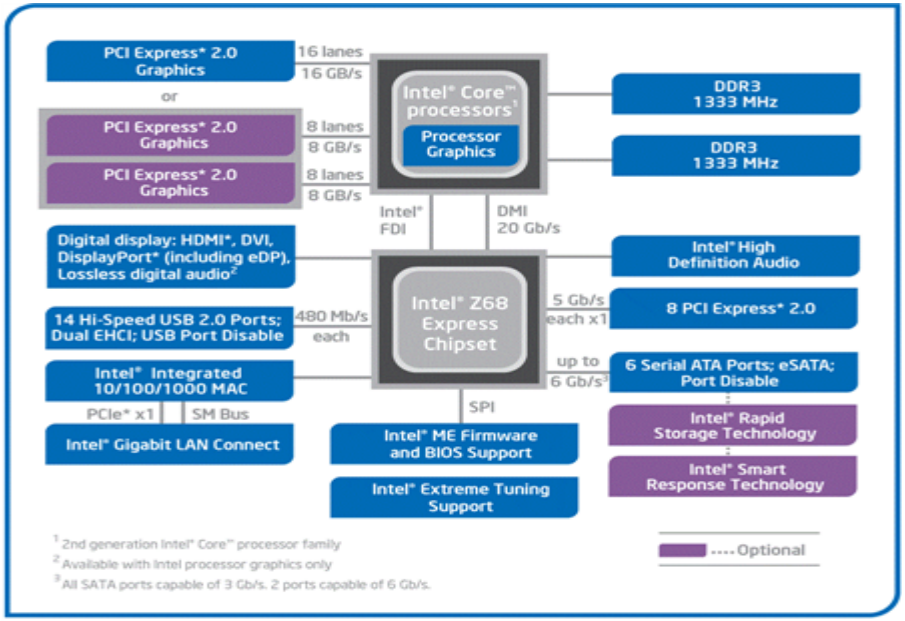
- CPU performance increases dramatically
- I/O system performance limited by mechanical delays $\Longrightarrow$ ***less than 10% performance improvement per year***
- Amdahl's law: system speedup limited by the slowest component:
  - Assume 10% I/O
  - CPU speedup = 10 $\Longrightarrow$ System speedup = 5
  - CPU speedup = 100 $\Longrightarrow$ System speedup = 10
- ***I/O will more and more become a bottleneck!***

I/O system

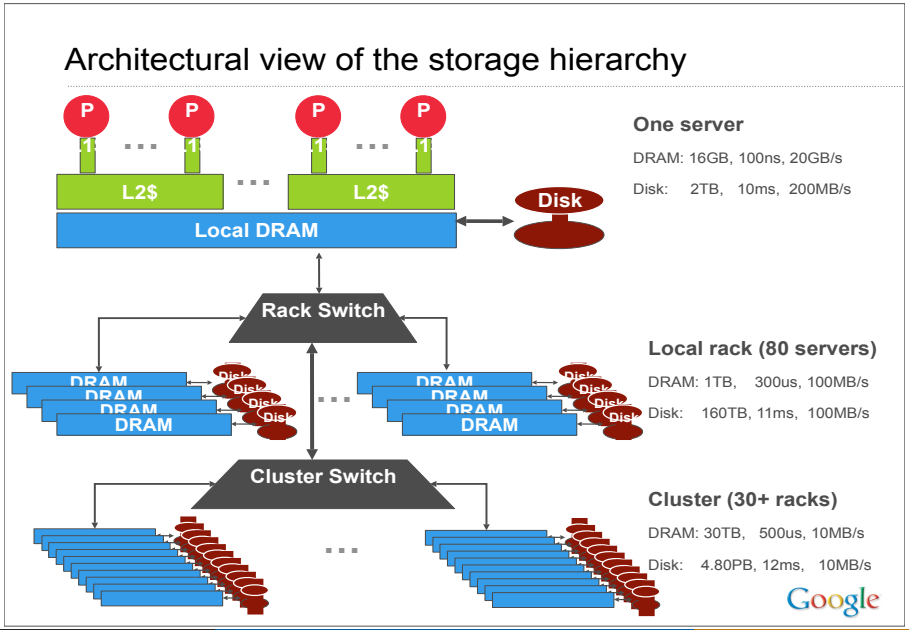
- We concentrate on disk systems
- Disks are used for virtual memory and file storage



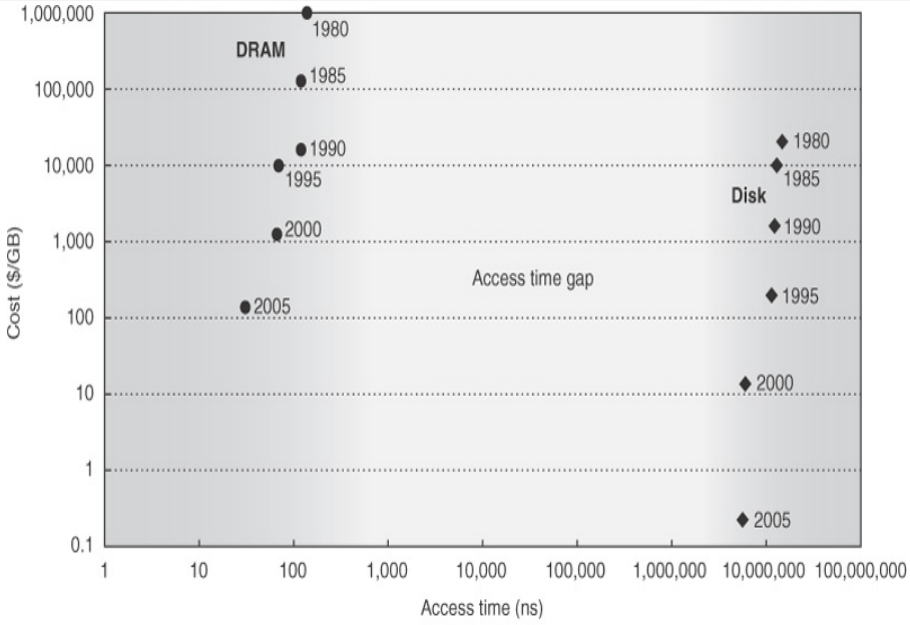
Chip-set architecture



Google view of storage hierarchy



Cost vs access time



Bus-based interconnect

Buses are the number one technology to connect the CPU with memory and I/O subsystems

- ***Advantages***: Low cost, shared medium to connect a variety of devices
- ***Disadvantages***: Inherent problem – limited bandwidth
- Bandwidth is limited by bus length and number of devices
- It is common to use dedicated memory and I/O-buses

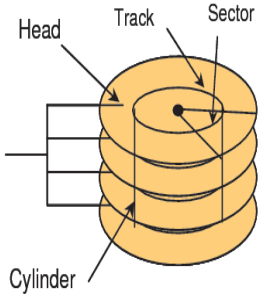
Buses vs Point-To-Point links

- Moore's Law ⇒ Point-to-Point links, switches
- Higher I/O bandwidth requirement

Standard	Width (bits)	Clock rate	MB/sec
(Parallel) ATA	8/16	133 MHz	133/266
Serial ATA	2	3 GHz	300
Serial ATA	2	6 GHz	600
USB 2.0	1		35
USB 3.0	1	4-5 Gbit/sec	400
(USB 3.1)	1	7-10 Gbit/sec	?
SCSI	16	80 MHz	320
Serial Attach SCSI	2	(DDR)	375
PCI	32/64	33/66 MHz	533
PCI Express	2	3 GHz	250
Ethernet	1	1 Gbit/s	<100

Outline

The organization of a disk

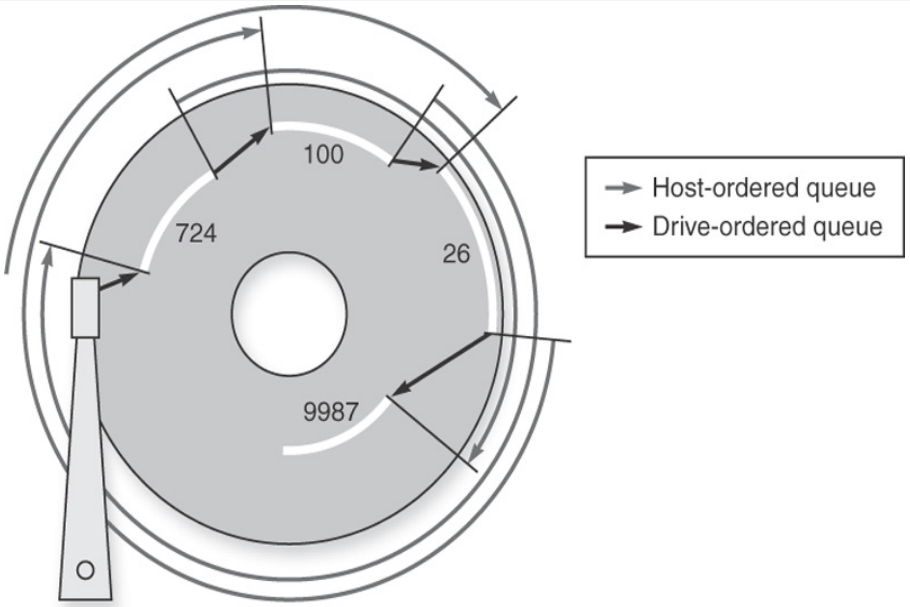


$$T_{response} = T_{queue} + T_{service}$$

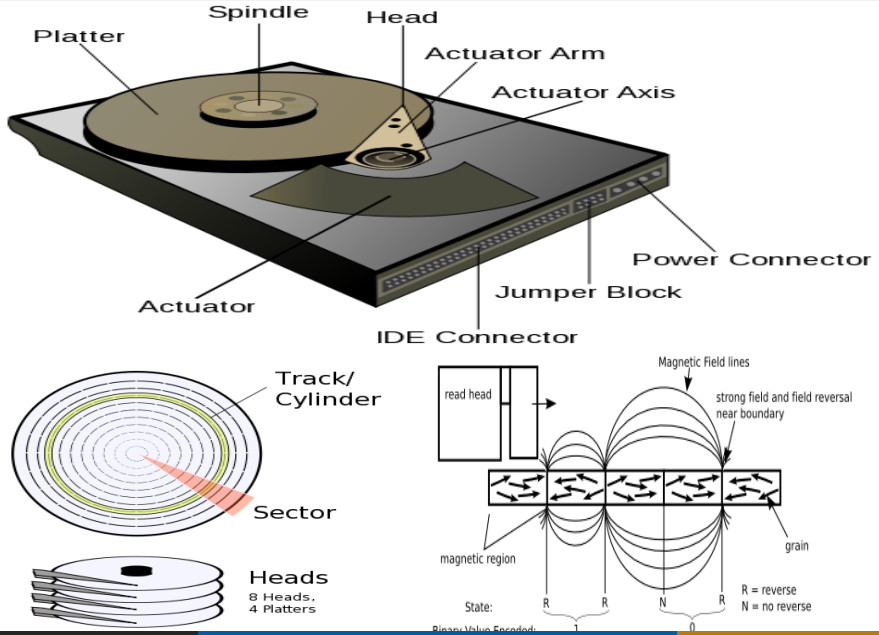
$$T_{service} = T_{controller} + T_{seek} + T_{rotation} + T_{transfer}$$

- Purpose
  - Long time, non-volatile storage
  - Large, inexpensive, slow level in the memory hierarchy
- Characteristics:
  - Seek time (3 - 8 -15 ms)
  - Rotational latency (2 - 4 - 8 ms)
- Transfer rate
  - 10 - 100 - 200 Mbyte/s
- Capacity
  - Terabytes
  - Quadruples every 3 years

New virtual organization

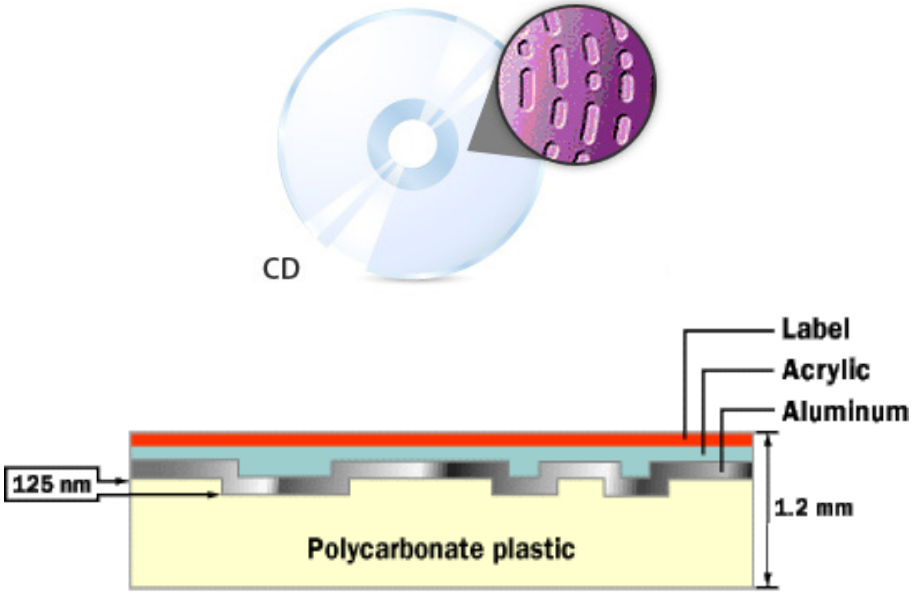


Hard disk revealed

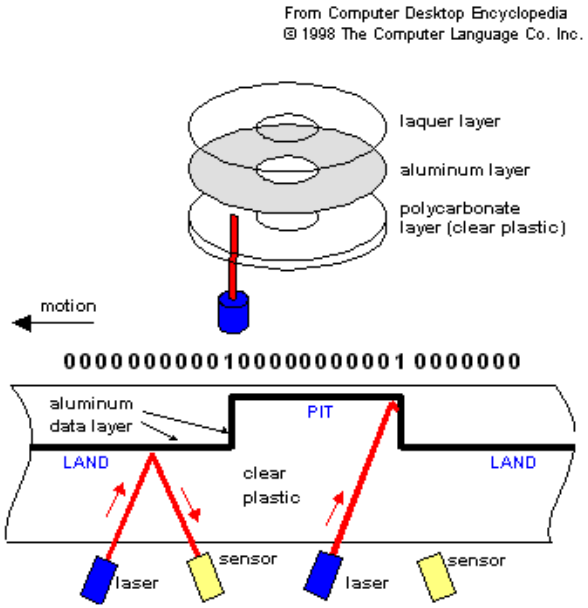


Hard disk anatomy

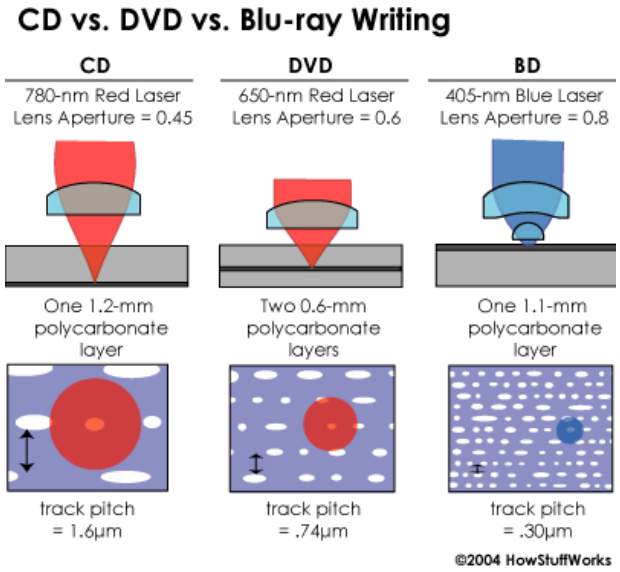
Optical storage - CD/DVD



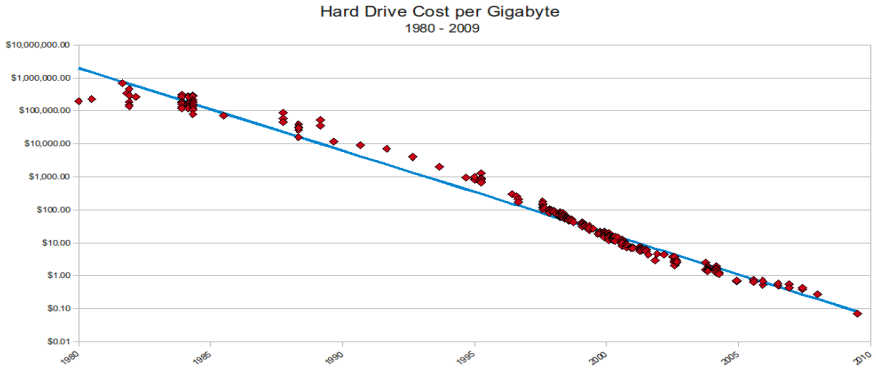
Reading CD



CD vs DVD vs Blue-Ray

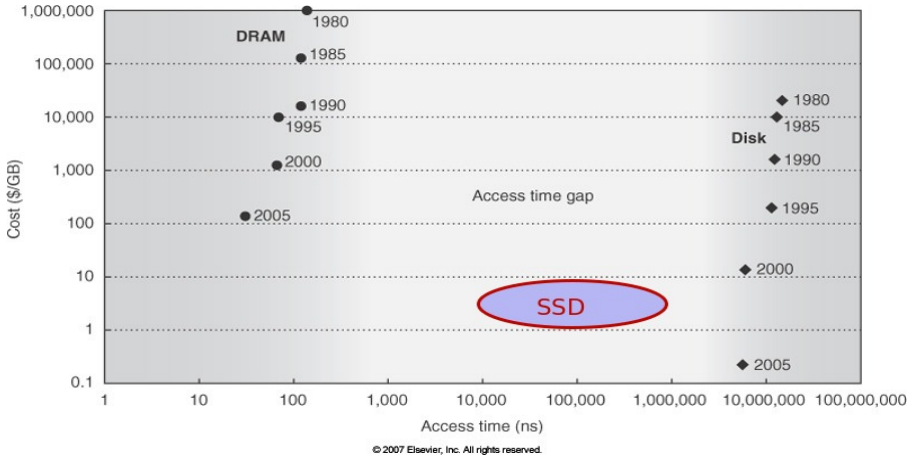


Disk technology trends



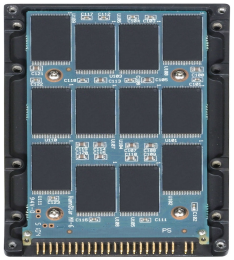
- Processing power doubles every 18 months
- Memory size doubles every 18 months
- Disk capacity doubles every 18 months
- ***Disk positioning rate (seek & rotate) doubles every ten years!***

DRAM disks?



- Can the ***access time gap*** be filled with other technologies?
- Cost is higher but SSD comming strong!
- Use disk with a small SSD as internal cache.

HDD vs SSD



HDD vs SSD

Product	Price	Rand write	Rand read	Seq. read	Seq. write	Read latency
	$/GB	MB/s				ms
Intel X-25 M (2009)	3.5	35.8	64.3	258		0.11
Intel SSD 330	0.79	300	175	550	530	
OCZ Vertex 4	0.78	350	300	560	510	
Samsung 840 Pro	1.19	350	350	550	550	
WD Caviar	0.12	1.26		100		
WD VelociRaptor	0.3	1.63	0.7	160	160	6.83

Designing a disk I/O system

There are several design alternatives to consider:

- Device performance?
- How is the disk *interfaced* to the CPU/memory
  - Interconnect technology:
    - System back-plane bus / I/O-bus?
    - Is the bus synchronous/asynchronous
  - Polled / Interrupt driven / DMA ?
  - Directly to the cache/memory?
- How do we determine performance?
  - Throughput
  - Response time
  - Utilization
  - Reliability / Availability

And of course: price, power, environment, ...

Outline

DMA performance

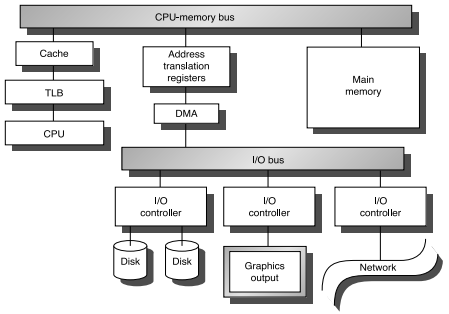
Example: 1000 transfers of 1 byte
10 Mbyte/s transfer rate $\Longrightarrow$ 0.1 μs/byte
1000 bytes $\Longrightarrow$ 100 μs

- Interrupt driven
  - 1000 interrupts at 2 μs each
  - 1000 interrupt service routines at 98 μs each
  - ***Totals 0.1 CPU seconds***
- DMA
  - 1 DMA set-up sequence at 50 μs
  - 1 interrupt at 2 μs
  - 1 interrupt service sequence at 98 μs
  - ***Totals 0.00015 CPU seconds***

DMA and virtual memory

If DMA works with physical addresses:

- Transferring buffers larger than 1 page problem since the virtual addressed buffer probably is not contiguous in physical memory
- What if the OS replaces a page used in a DMA I/O transfer?



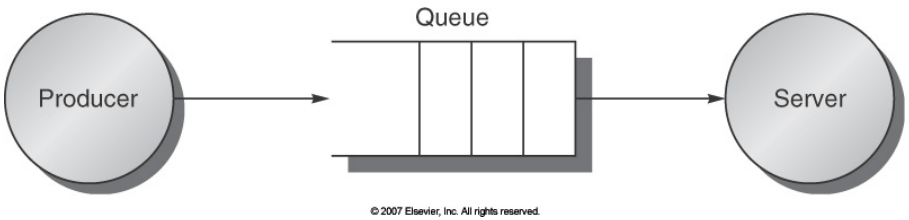
Solutions:

- Transfer into OS address space (requires one extra copying of data)
- Use virtual DMA

Synchronous/Asynchronous I/O

- Synchronous I/O
  - Request data
  - Wait for data
  - Use data
- Asynchronous I/O
  - Request data
  - Continue with other things
  - Block when trying to use data
  - Compare non-blocking caches in out-of-order CPUs
  - Multiple outstanding I/O requests

Producer-server model

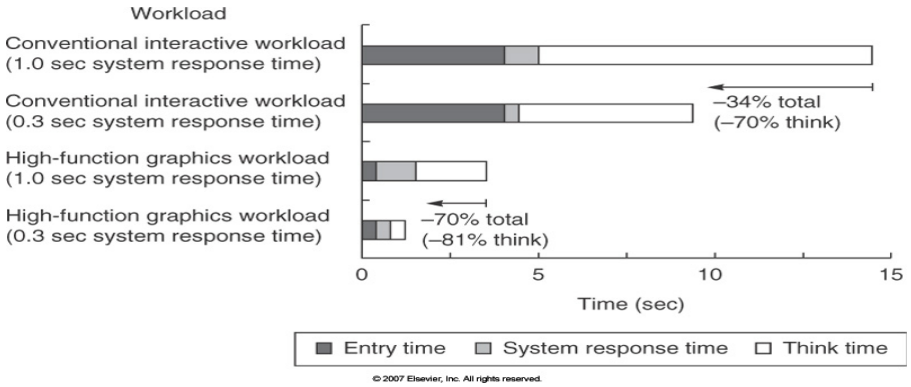


Response time: Time from placed in queue until server is finished

Throughput: Average no of tasks completed per time unit

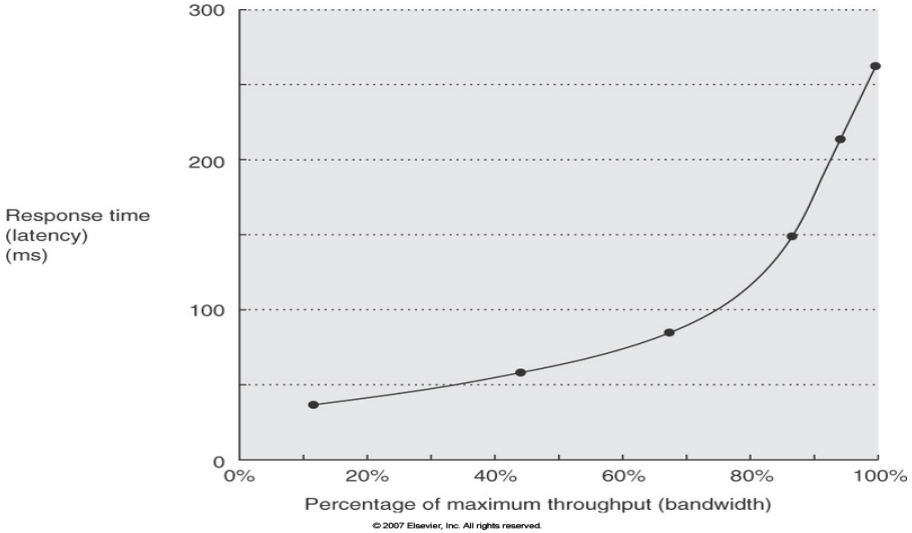
Interlude - System response time vs Think time

- Interactive environments:
  - Each interaction or transaction has 3 parts:
    - ***Entry Time***: time for user to enter command
    - ***System Response Time***: time between user entry & system replies
    - ***Think Time***: Time from response until user begins next command



How important is response time?

- Performance metrics: Throughput and Response time
- Improved response time (lower) leads to increased productivity!



Benchmarks

- TPC-C – Transaction processing benchmarks
- SFS – NFS benchmark
- SPECMail – Mail servers
- SPECWeb2005 – Web servers
- PCMark, SYSMark, IOMeter, ...

Outline

Reliability / Availability – Dependability

Definitions:

- Reliability – Is anything broken?
- Availability – Is the system available for the user?
- Dependability – Is the system doing what it is supposed to do?

Why is this an issue?

- Small disks and large disks cost the same / byte
- An array of *N* small disks can achieve higher bandwidth than one large disk
- However, the reliability is 1/*N* of the reliability of a single disk

Interlude - Google experience - Jeff Dean

The Joys of Real Hardware

Typical first year for a new cluster:

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
~1 network rewiring (rolling ~5% of machines down over 2-day span)
~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
~5 racks go wonky (40-80 machines see 50% packetloss)
~8 network maintenances (4 might cause ~30-minute random connectivity losses)
~12 router reloads (takes out DNS and external vips for a couple minutes)
~3 router failures (have to immediately pull traffic for an hour)
~dozens of minor 30-second blips for dns
~1000 individual machine failures
~thousands of hard drive failures
slow disks, bad memory, misconfigured machines, flaky machines, etc.

Long distance links: wild dogs, sharks, dead horses, drunken hunters, etc.

RAID

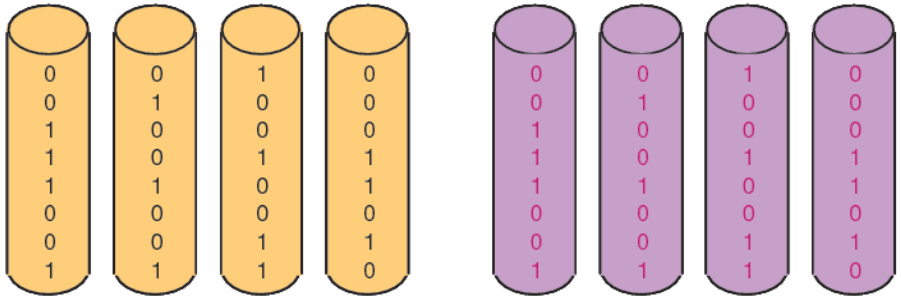
Redundant Array of Inexpensive (Independent) **Disks**

- Files are “striped” across multiple disks
- Redundancy yields high data availability
- Higher throughput
- Disks will fail!
- RAID issues:
  - Contents reconstructed from data ***redundantly*** stored in the array
  - Capacity penalty to store the redundant data
  - Bandwidth penalty to update

RAID types

	RAID level	Failures tolerated	Overhead 8 data disks	comment
0	striped	0	0	JBOD, common
1	mirrored	1-8	8	high overhead
2	ECC	1	4	not used
3	bit parity	1	1	synchronized drives
4	block parity	1	1	
5	block parity distributed	1	1	common
6	row-diagonal dual parity	2	2	high availability
01	mirrored stripes	1-8	8	
10	striped mirrors	1-8	8	

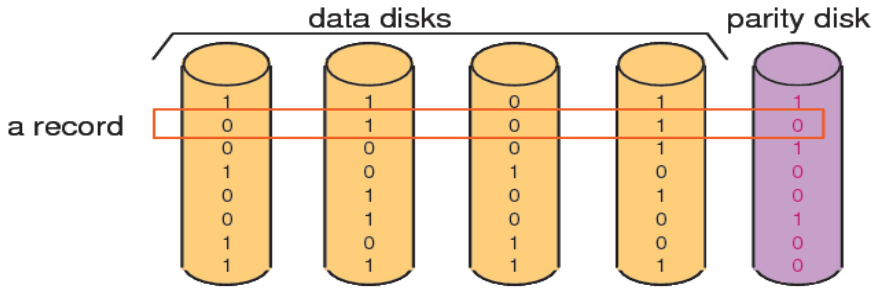
Disk mirroring – **RAID-1**



- Each disk is fully duplicated onto its “shadow”
- Logical write = two physical writes
- Easy recovery, just use the copy
- 100% capacity overhead

Targeted for high I/O rate, high availability systems

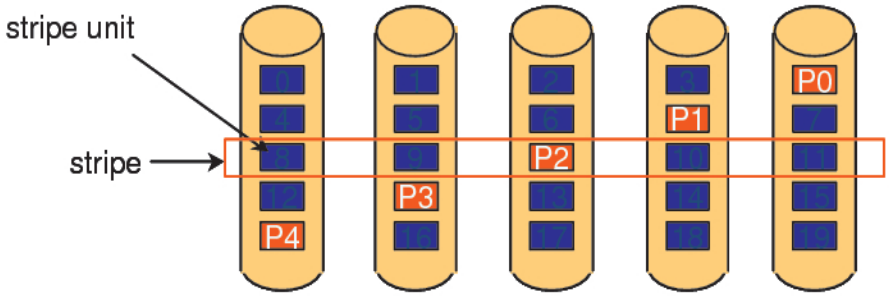
Bit Parity – **RAID 3**



- Parity is computed across recovery group
- 20% capacity cost (this config), wider arrays reduce capacity cost but also decreases expected availability
- Logically a single high capacity, high transfer rate disk

Targeted for high bandwidth applications: Scientific, image processing

Block-Interleaved Parity – **RAID 5**



- “Small” reads can occur in parallel
- A logical write becomes four physical I/O-ops
- Independent writes possible due to interleaved parity

Targeted for mixed applications with high-performance I/O needs

RAID 6

Stripe	Disk 0	1	2	3	4	5
0	$D_{(0,0)}$	$D_{(0,1)}$	$D_{(0,2)}$	$D_{(0,3)}$	P_0	Q_0
1	$D_{(1,0)}$	$D_{(1,1)}$	$D_{(1,2)}$	P_1	Q_1	$D_{(1,3)}$
2	$D_{(2,0)}$	$D_{(2,1)}$	P_2	Q_2	$D_{(2,2)}$	$D_{(2,3)}$
3	$D_{(3,0)}$	P_3	Q_3	$D_{(3,1)}$	$D_{(3,2)}$	$D_{(3,3)}$
4	P_4	Q_4	$D_{(4,0)}$	$D_{(4,1)}$	$D_{(4,2)}$	$D_{(4,3)}$
5	Q_5	$D_{(5,3)}$	$D_{(5,0)}$	$D_{(5,1)}$	$D_{(5,2)}$	P_5

- 2 blocks of parity for each stripe (XOR, Reed-Solomon)
- Can recover from 2 failed disks
- Similar to RAID 5
- Good for arrays with many disks

Targeted for high-availability applications

Outline

Summary I/O

I/O:

- I/O performance is important!
- The task of the I/O system designer:
  - meet performance needs
  - cost-effective
  - reliability, availability
- I/O system parts
  - CPU interface
  - Interconnect technology
  - Device performance

Disks:

- Disks have moving parts leading to long service times
- RAID disk arrays provide high bandwidth, high capacity disk storage at a reasonable cost
- SSD is faster and more expensive