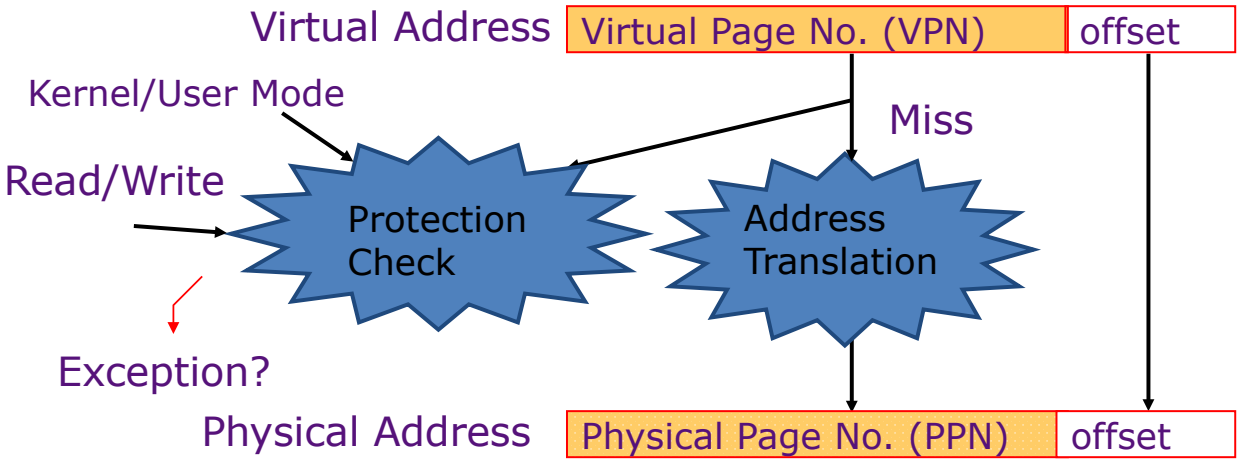
CS 61C: Great Ideas in Computer Architecture (Machine Structures)

Lecture 24: More I/O: DMA, Disks, Networking

Instructors: Krste Asanović & Randy H. Katz

http://inst.eecs.berkeley.edu/~cs61c/fa17

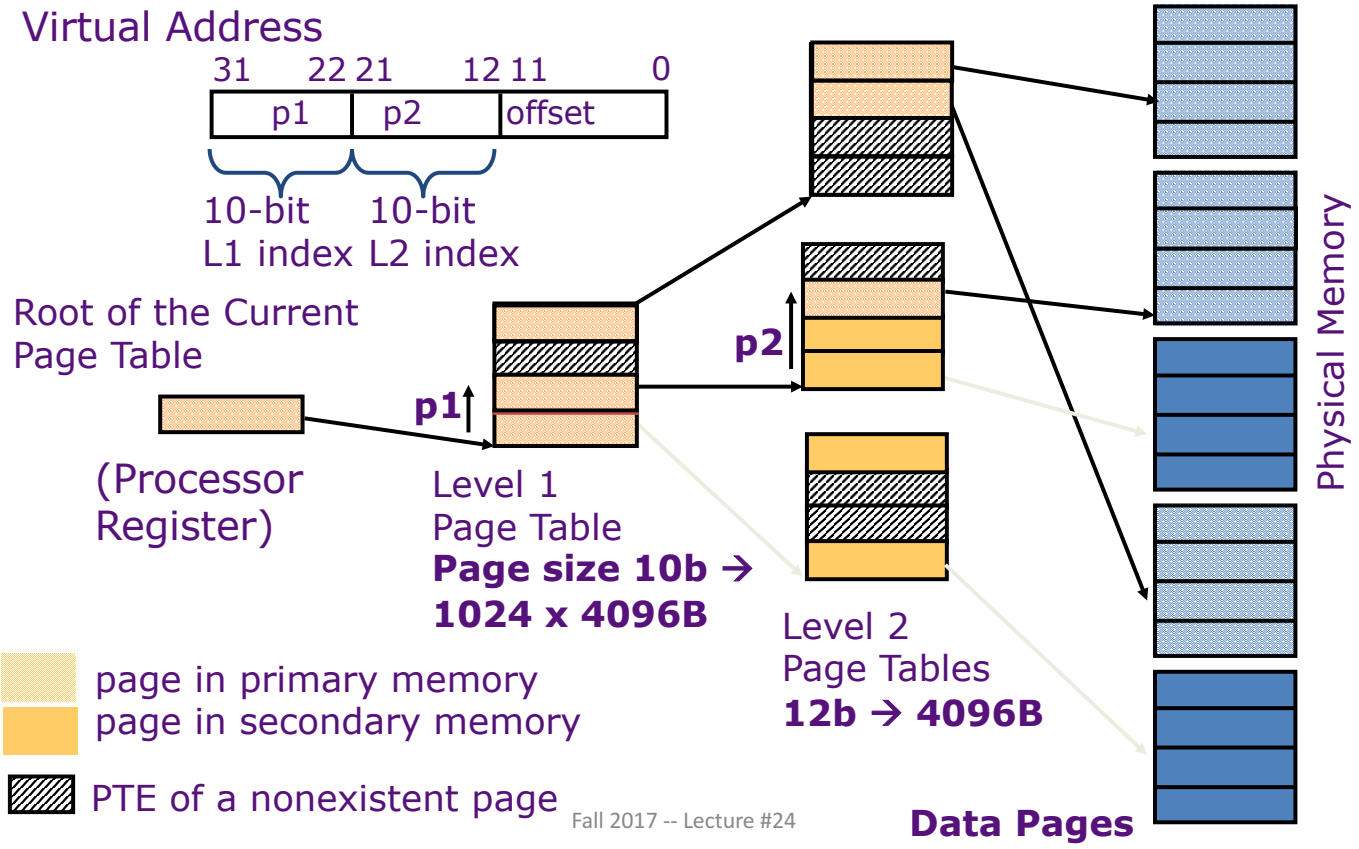
Review: Address Translation and Protection



- Every instruction and data access needs address translation and protection checks

Good VM design should be fast (~one cycle) and space efficient

Review: Hierarchical Page Tables



Review: Translation Lookaside Buffers (TLB)

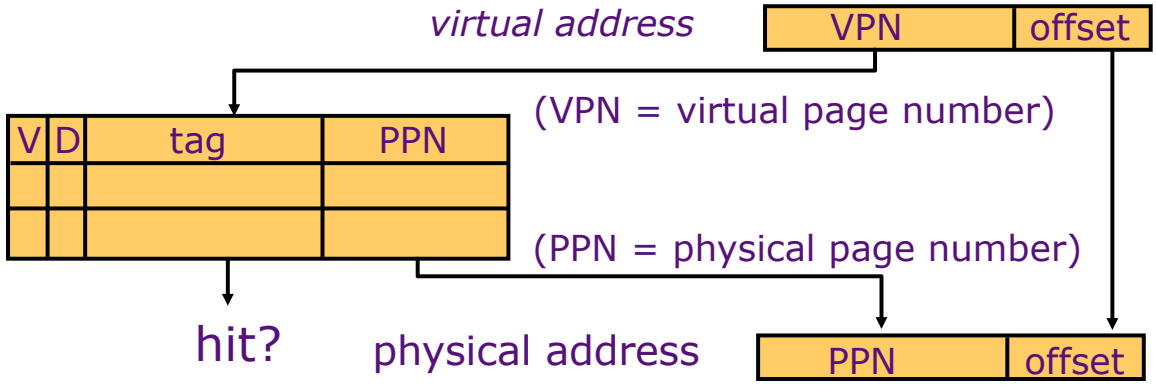
Address translation is very expensive!

In a two-level page table, each reference becomes three memory accesses

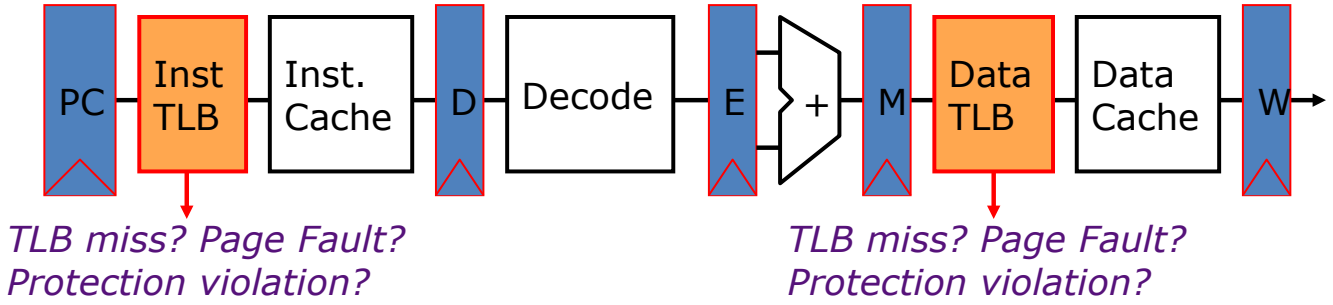
Solution: *Cache some translations in TLB*

TLB hit ⇒ *Single-Cycle Translation*

TLB miss ⇒ *Page-Table Walk to refill*

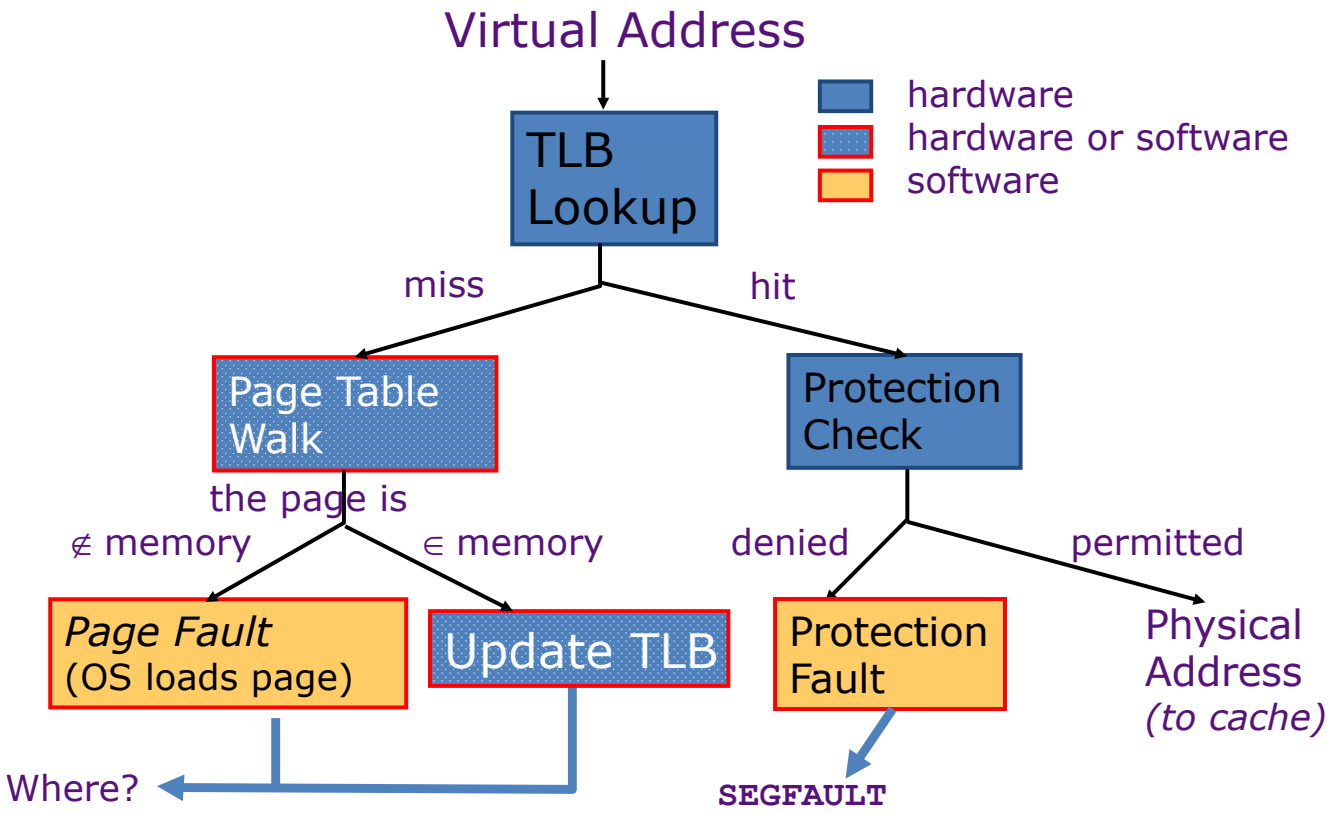


VM-related Events in Pipeline



- Handling a TLB miss needs a hw or sw mechanism to refill TLB
  - Usually done in hardware
- Handling a page fault (e.g., page is on disk) needs ***precise*** trap so software handler can easily resume after retrieving page
- Protection violation may abort process

Address Translation: Putting it all Together



Review: I/O

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- "Programmed I/O":
  - CPU execs lw/sw instructions for all data movement to/from devices
  - CPU spends time doing two things:
    1. Getting data from device to main memory
    2. Using data to compute

Reality Check!

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- ~~"Programmed I/O":~~ **DMA**
  - ~~CPU execs lw/sw instructions for all data movement to/from devices~~
  - CPU spends time doing ~~2 things~~:
    1. ~~Getting data from device to main memory~~
    2. Using data to compute

Outline

- Direct Memory Access
- Review: Disks
- Networking
- Storage Attachment Evolution
- Rack Scale Memory
- And in Conclusion …

Outline

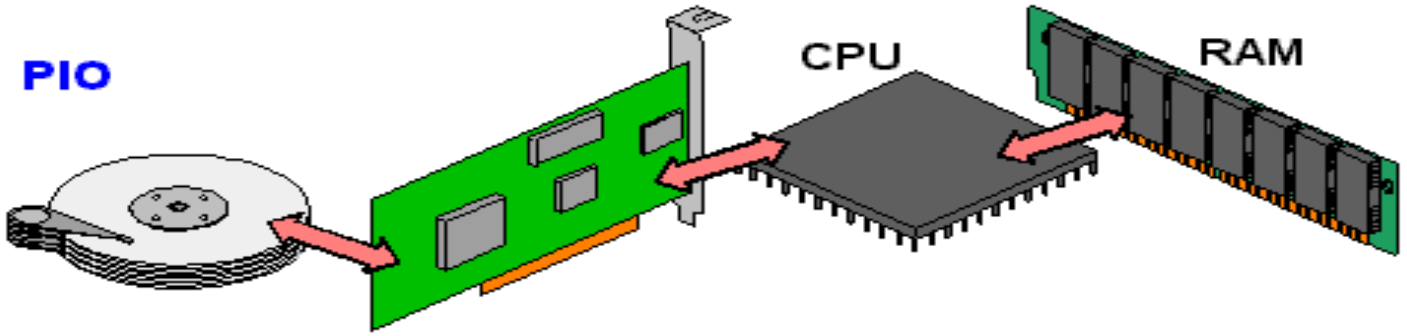
- Direct Memory Access
- Disks
- Networking
- Storage Attachment Evolution
- Rack Scale Memory
- And in Conclusion …

What's Wrong with Programmed I/O?

- Not ideal because …
  1. CPU has to execute all transfers, could be doing other work
  2. Device speeds don't align well with CPU speeds
  3. Energy cost of using beefy general-purpose CPU where simpler hardware would suffice
- Until now CPU has sole control of main memory
- 5% of CPU cycles on Google Servers spent in memcpy() and memmove() library routines!*

  *Kanev et al., "Profiling a warehouse-scale computer," ICSA 2015, (June 2015), Portland, OR.

PIO vs. DMA



Direct Memory Access (DMA)

- Allows I/O devices to directly read/write main memory
- New Hardware: the DMA Engine
- DMA engine contains registers written by CPU:
  - Memory address to place data
  - # of bytes
  - I/O device #, direction of transfer
  - unit of transfer, amount to transfer per burst

Operation of a DMA Transfer

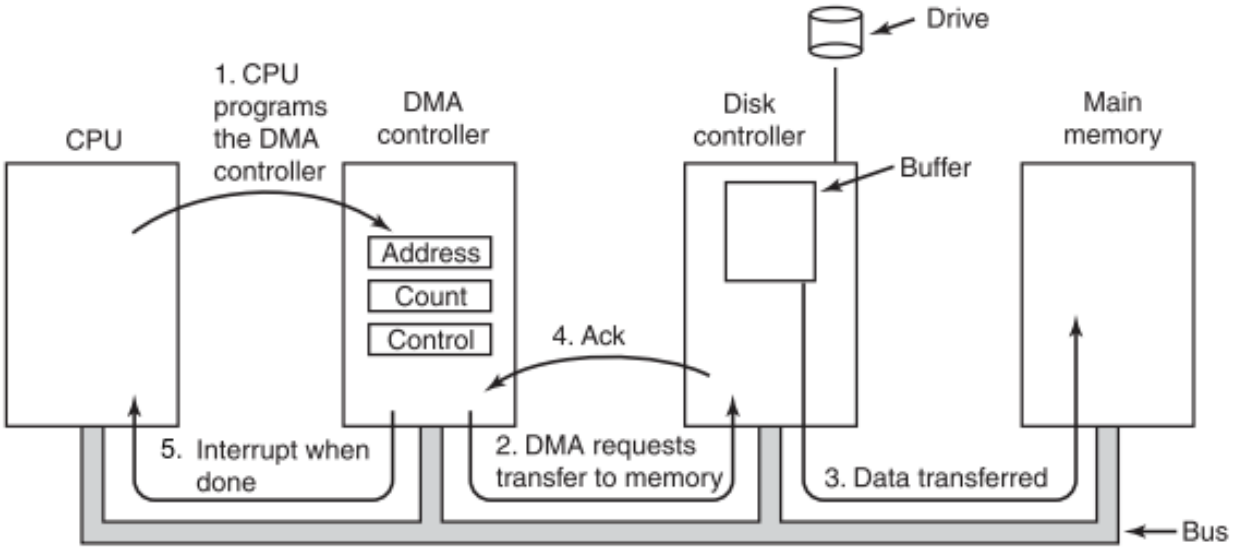


Figure 5-4. Operation of a DMA transfer.

[From Section 5.1.4 Direct Memory Access in *Modern Operating Systems* by Andrew S. Tanenbaum, Herbert Bos, 2014]

DMA: Incoming Data

1. Receive interrupt from device
2. CPU takes interrupt, begins transfer
   – Instructs DMA engine/device to place data @ certain address
3. Device/DMA engine handle the transfer
   – CPU is free to execute other things
4. Upon completion, Device/DMA engine interrupt the CPU again

DMA: Outgoing Data

1. CPU decides to initiate transfer, confirms that external device is ready
2. CPU begins transfer
   – Instructs DMA engine/device that data is available @ certain address
3. Device/DMA engine handle the transfer
   – CPU is free to execute other things
4. Device/DMA engine interrupt the CPU again to signal completion

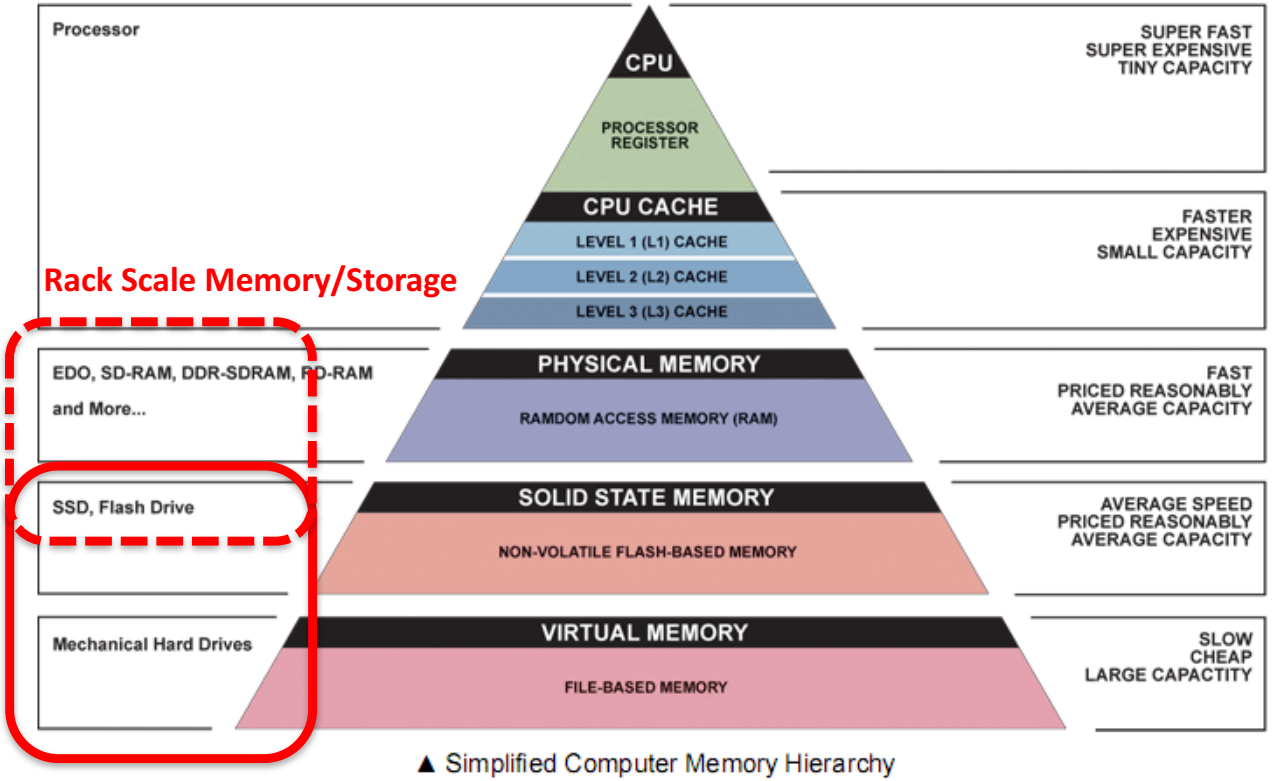
DMA: Some New Problems

- Where in the memory hierarchy do we plug in the DMA engine? Two extremes:
  - Between L1$ and CPU:
    - Pro: Free coherency
    - Con: Trash the CPU's working set with transferred data
  - Between Last-level cache and main memory:
    - Pro: Don't mess with caches
    - Con: Need to explicitly manage coherency

Outline

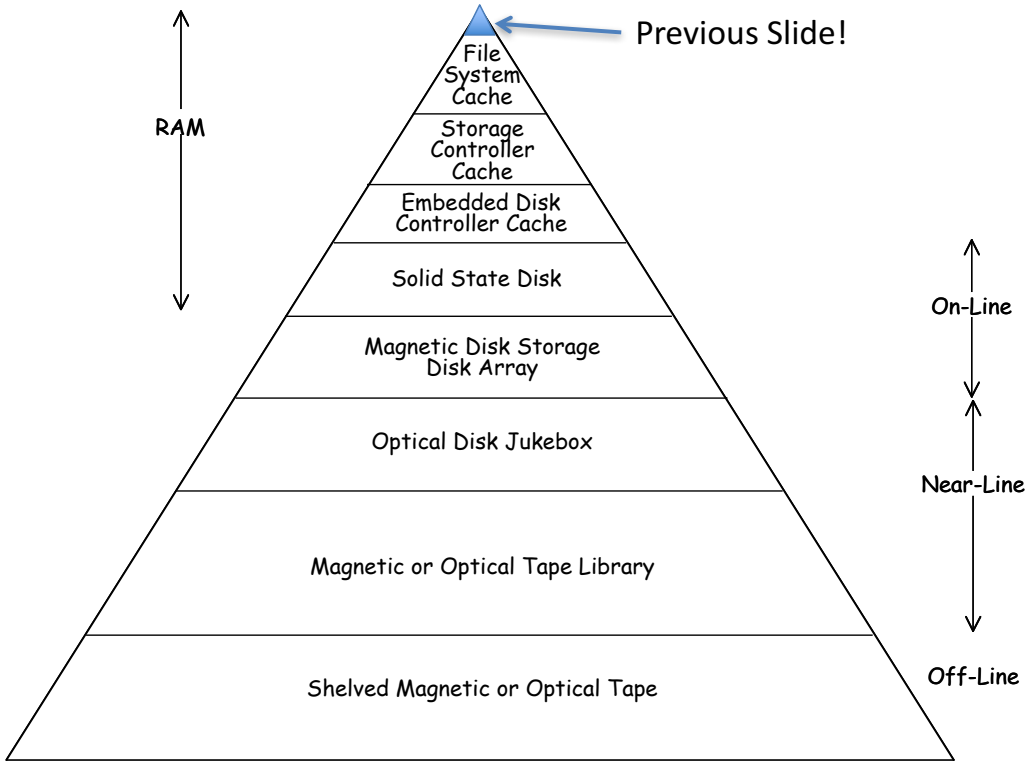
- Direct Memory Access
- Disks
- Networking
- And in Conclusion …

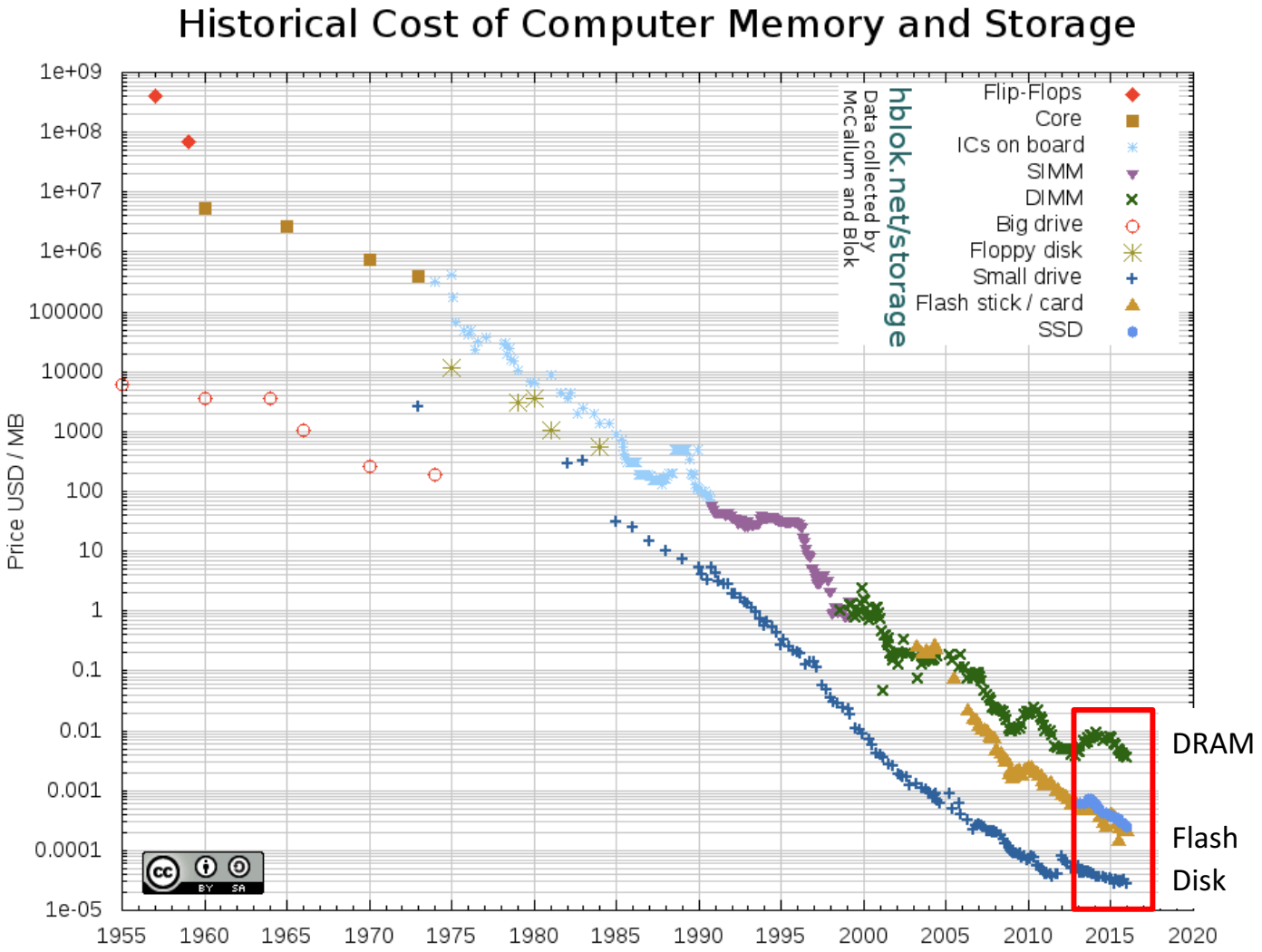
Computer Memory Hierarchy
One of our "Great Ideas"



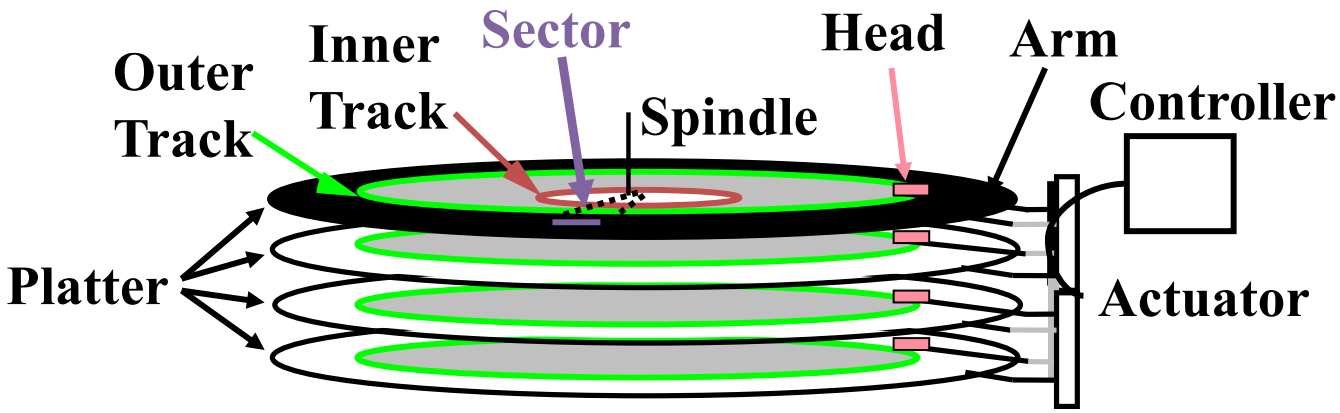
▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng

Storage-Centric View of the Memory Hierarchy




Historical Cost of Computer Memory and Storage
Price USD / MB
1e+09
1e+08
1e+07
1e+06
100000
10000
1000
100
10
1
0.1
0.01
0.001
0.0001
1e-05
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
hblok.net/storage
Data collected by McCallum and Blok
Flip-Flops
Core
ICs on board
SIMM
DIMM
Big drive
Floppy disk
Small drive
Flash stick / card
SSD
DRAM
Flash
Disk

Disk Device Performance (1/2)



- **Disk Access Time = Seek Time + Rotation Time + Transfer Time + Controller Overhead**
  - Seek Time = time to position the head assembly at the proper cylinder
  - Rotation Time = time for the disk to rotate to the point where the first sectors of the block to access reach the head
  - Transfer Time = time taken by the sectors of the block and any gaps between them to rotate past the head

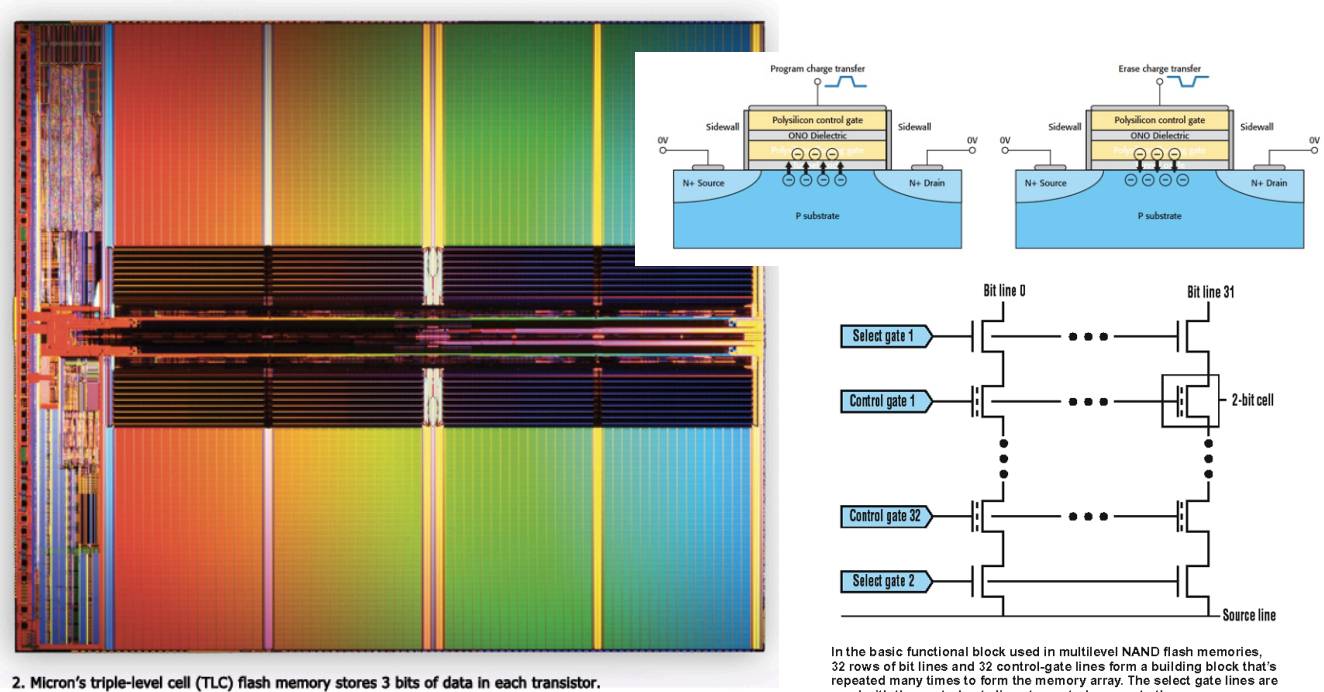
Disk Device Performance (2/2)

- Average values to plug into the formula:
- Rotation Time: Average distance of sector from head?
  - 1/2 time of a rotation
    - 7200 Revolutions Per Minute ⇒ 120 Rev/sec
    - 1 revolution = 1/120 sec ⇒ 8.33 milliseconds
    - 1/2 rotation (revolution) ⇒ 4.17 ms
- Seek time: Average no. tracks to move arm?
  - Number of tracks/3 (see CS186 for the math)
  - Then, seek time = number of tracks moved × time to move across one track

But wait!

- Performance estimates are different in practice
- Modern disks have on-disk caches, which are hidden from the outside world
  - Generally, what limits real performance is the on-disk cache access time

Flash Memory / SSD Technology



2. Micron's triple-level cell (TLC) flash memory stores 3 bits of data in each transistor.

In the basic functional block used in multilevel NAND flash memories, 32 rows of bit lines and 32 control-gate lines form a building block that's repeated many times to form the memory array. The select gate lines are used with the control gate lines to control access to the array.

- NMOS transistor with an additional conductor between gate and source/drain which "traps" electrons. The presence/absence is a 1 or 0
- Memory cells can withstand a limited number of program-erase cycles. Controllers use a technique called *wear leveling* to distribute writes as evenly as possible across all the flash blocks in the SSD.

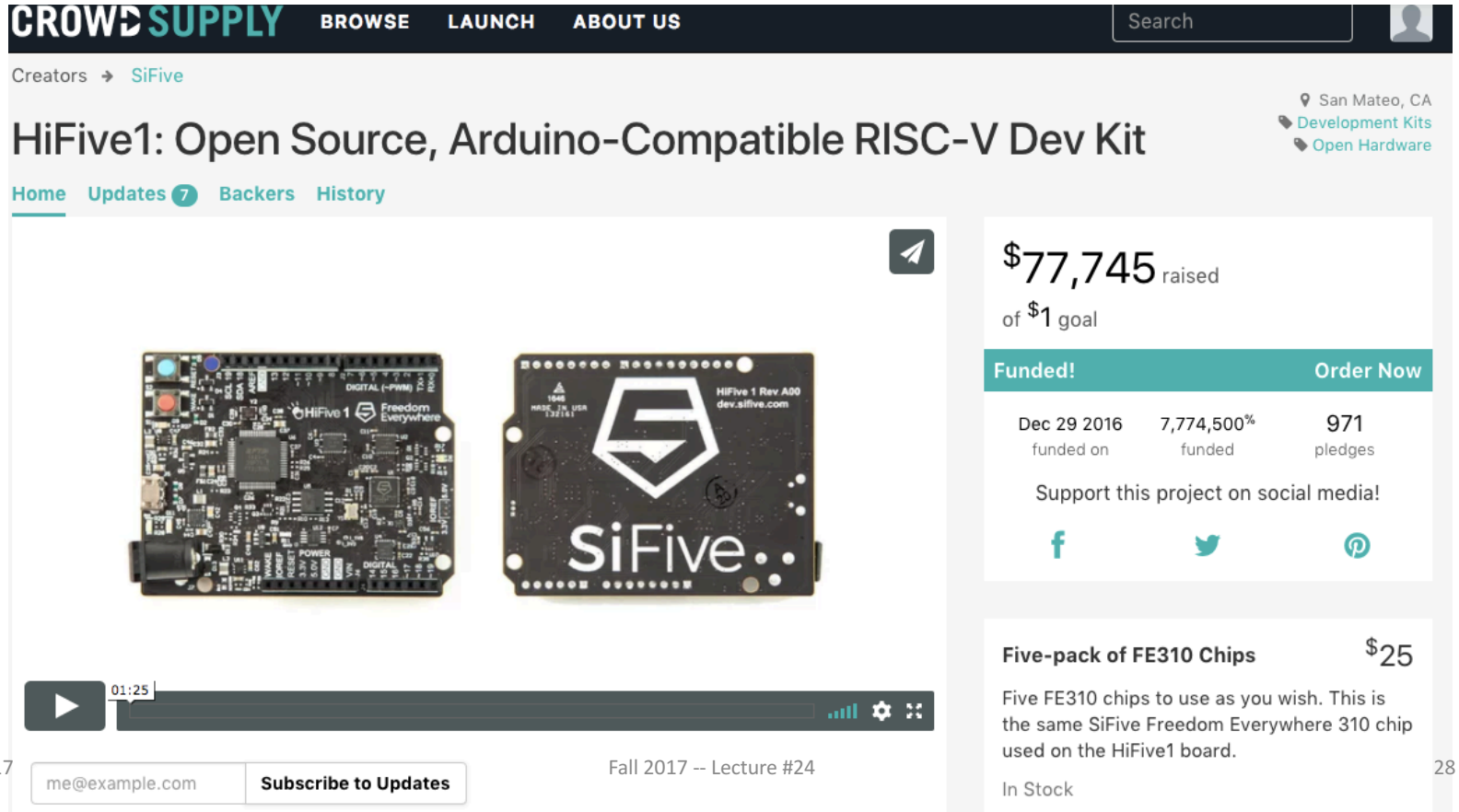
Administrivia (1/2)

- Project 3.2 (Performance Contest) has been released
  - Up to 5 extra credit points for the highest speedups
- Final exam:
  - 14 December, 7-10 PM @ TBA
  - Contact head TA (Steven Ho) about conflicts if you haven't been contacted yet
  - Review Lectures and Book with eye on the important concepts of the course
- Review Session Fri Dec 8, 5-8 PM @TBA
- Electronic Course Evaluations starting this week! See https://course-evaluations.berkeley.edu

Administrivia (2/2)

- HW6 party tonight night (Nov 21) in the Woz from 5-8 PM
- No discussions or labs this week!
  - Labs resume Monday after Thanksgiving
    - Lab 11 due in any lab before December 1
    - Lab 13 due in any OH before December 8
- Homework 6 due tomorrow night
- Project 4 to be released Friday latest
- Homework 7 to be released Monday after break

Winners of the Project 3 Performance Competition!



A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A. Patterson, Garth A. Gibson and Randy H. Katz

EECS Department
University of California, Berkeley
Technical Report No. UCB/CSD-87-391
December 1987

http://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf

Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O. While the capacity of Single Large Expensive Disk (SLED) has grown rapidly, the performance improvement of SLED has been modest. Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability.

This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAIDs to an IBM 3380 and a Fugitsu Super Eagle.

BibTeX citation:

```
@techreport{Patterson:CSD-87-391,
    Author = {Patterson, David A. and Gibson, Garth A. and Katz, Randy H.},
    Title = {A Case for Redundant Arrays of Inexpensive Disks (RAID)},
    Institution = {EECS Department, University of California, Berkeley},
    Year = {1987},
    Month = {Dec},
    URL = {http://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/5853.html},
    Number = {UCB/CSD-87-391},
    Abstract = {Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increa
}
```

Happy Birthday Internet! 11/21/69

www.computerhistory.org/tdih/November/21/

EXHIBITS AT THE MUSEUM

- Overview
- Revolution
- Deleted City
- Thinking Big: Ada Lovelace
- Where To?
- IBM 1401 Demo Lab
- PDP-1 Demo Lab

COMING SOON

- Make Software: Change the World!

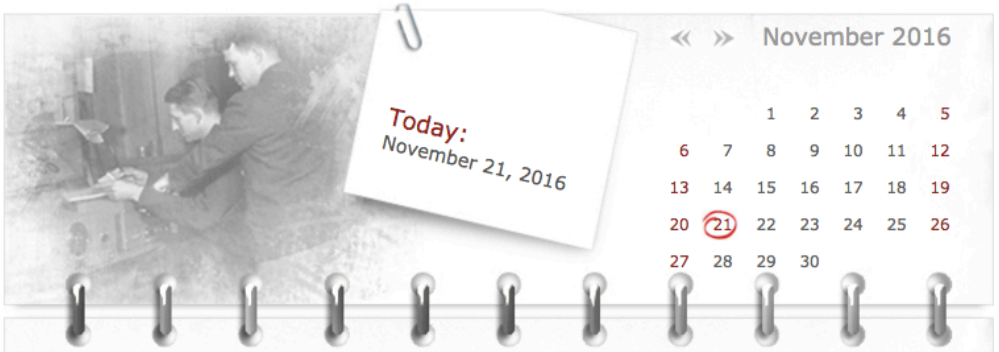
PAST EXHIBITS

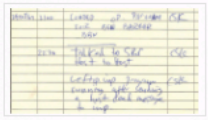
- The Babbage Engine

EXHIBITS ONLINE

- Overview
- Revolution
- Hall of Fellows
- Internet History 1962 to 1992
- Mastering the Game
- Microprocessors 1971 to 1996

This Day in History: November 21





First ARPANET IMP log--a record of the first message ever sent over the ARPANET, which was transmitted at 10:30 pm on October 29, 1969

November 21, 1969

First ARPANET Link Put Into Service

ARPANET was an early computer network developed by J.C.R. Licklider, Robert Taylor, and other researchers for the U.S. Department of Defense's Advanced Research Projects Agency (DARPA). It connected a computer at UCLA with a computer at the Stanford Research Institute, Menlo Park, CA. In 1973, the government commissioned Vinton Cerf and Robert E. Kahn to create a national computer network for military, governmental, and institutional use. The network used packet-switching, flow-control, and fault-tolerance techniques developed by ARPANET. Historians consider this worldwide network to be the origin of the Internet.

CS61c in the News: Supercomputer in a File Drawer

"The Raspberry Pi modules let developers figure out how to write this software and get it to work reliably without having a dedicated testbed of the same size, which would cost a quarter billion dollars and use 25 megawatts of electricity."

Gary Grider, leader of the High Performance Computing Division

The BitScope Pi Cluster Modules system creates an affordable, scalable, highly parallel testbed for high-performance-computing system-software developers. The system comprises five rack-mounted BitScope Pi Cluster Modules consisting of 3,000 cores using Raspberry Pi ARM processor boards, fully integrated with network switching infrastructure.

CREDIT: Bitscope


BeCool

Peer Instruction Question

- We have the following disk:
  - 15000 Cylinders, 1 ms to cross 1000 Cylinders
  - 15000 RPM = 4 ms per rotation
  - Want to copy 1 MB, transfer rate of 1000 MB/s
  - 1 ms controller processing time
- What is the access time using our model?

Disk Access Time = Seek Time + Rotation Time + Transfer Time + Controller Processing Time

A	B	C	D
10.5 ms	9 ms	8.5 ms	11.5 ms

Outline

- Direct Memory Access
- Disks
- Networking
- Storage Attachment Evolution
- Rack Scale Memory
- And in Conclusion …

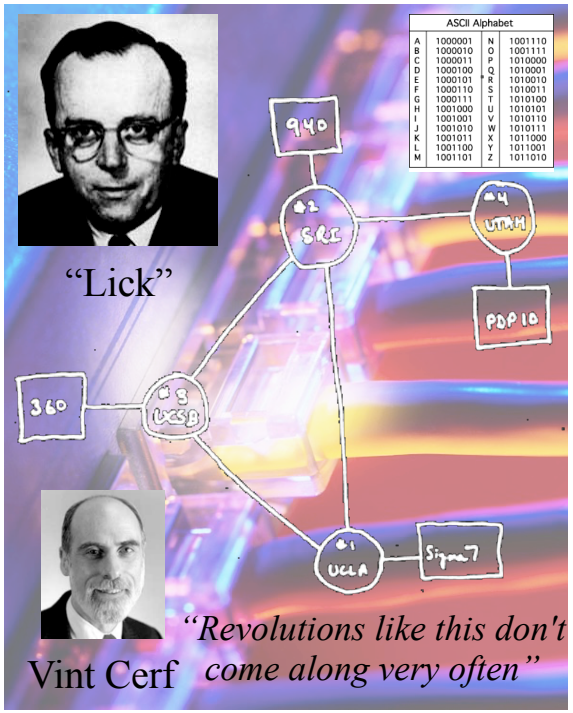
Networks: Talking to the Outside World

- Originally sharing I/O devices between computers
  - E.g., printers
- Then communicating between computers
  - E.g., file transfer protocol
- Then communicating between people
  - E.g., e-mail
- Then communicating between networks of computers
  - E.g., file sharing, www, …

www.computerhistory.org/internet_history

The Internet (1962)

- History
  - 1963: JCR Licklider, while at DoD's ARPA, writes a memo describing desire to connect the computers at various research universities: Stanford, Berkeley, UCLA, ...
  - 1969 : ARPA deploys 4 "nodes" @ UCLA, SRI, Utah, & UCSB
  - 1973 Robert Kahn & Vint Cerf invent TCP, now part of the Internet Protocol Suite
- Internet growth rates
  - Exponential since start!



www.greatachievements.org/?id=3736

en.wikipedia.org/wiki/Internet_Protocol_Suite

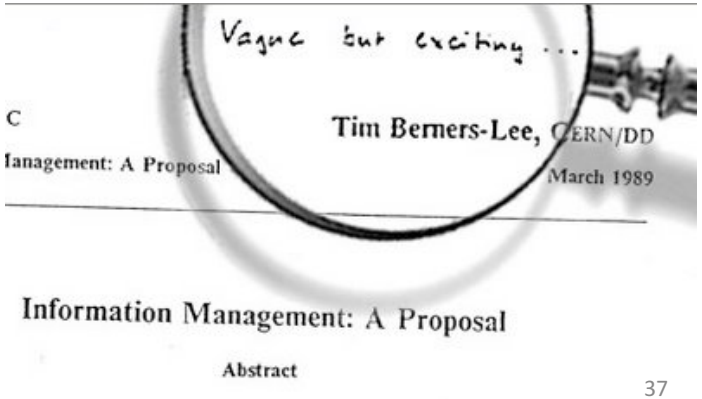
en.wikipedia.org/wiki/History_of_the_World_Wide_Web

The World Wide Web (1989)

- "System of interlinked hypertext documents on the Internet"
- History
  - 1945: Vannevar Bush describes hypertext system called "memex" in article
  - 1989: Sir Tim Berners-Lee proposed and implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server using the internet.
  - ~2000 Dot-com entrepreneurs rushed in, 2001 bubble burst
- Today : Access anywhere!

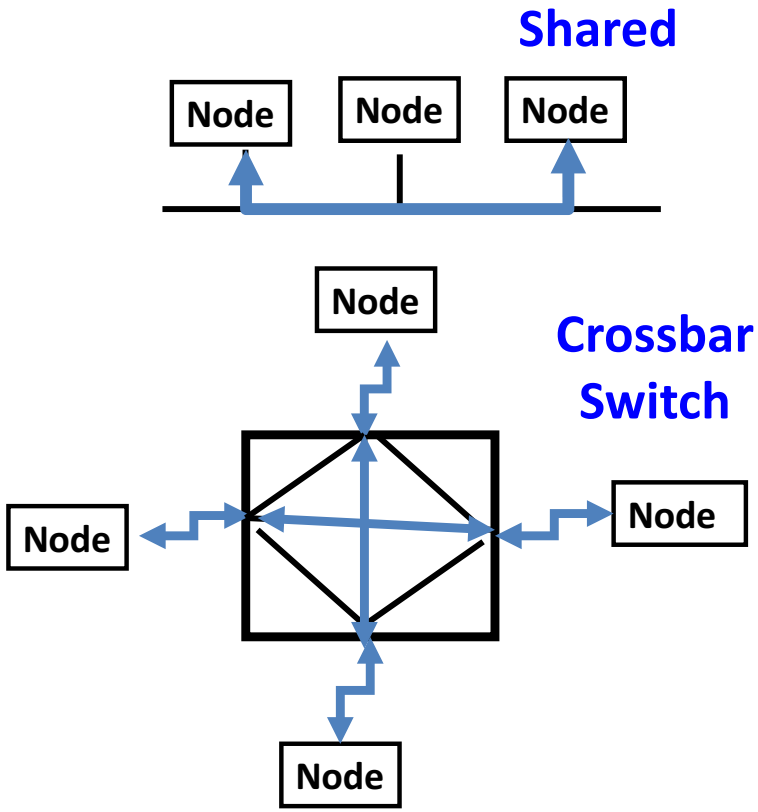
Tim Berners-Lee

World's First web server in 1990



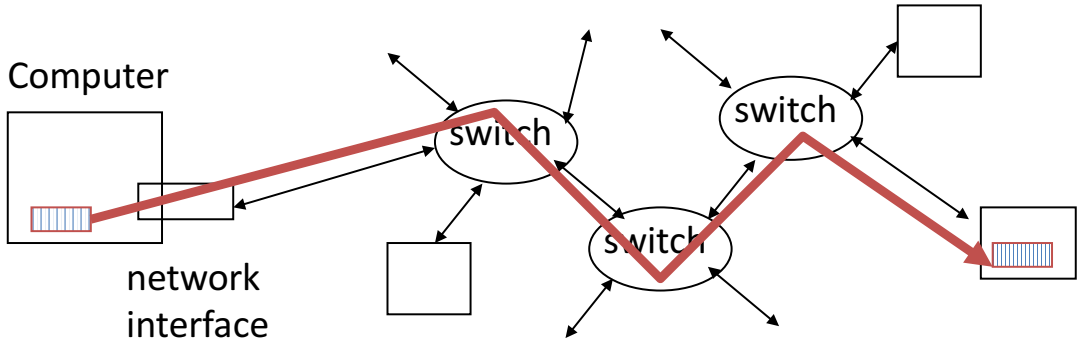
Shared vs. Switch-Based Networks

- Shared vs. Switched:
  - Shared: 1 at a time (CSMA/CD)
  - Switched: pairs ("point-to-point" connections) communicate at same time
- Aggregate bandwidth (BW) in switched network is many times that of shared:
  - Point-to-point faster since no arbitration, simpler interface



What Makes Networks Work?

- Links connecting switches and/or routers to each other and to computers or devices

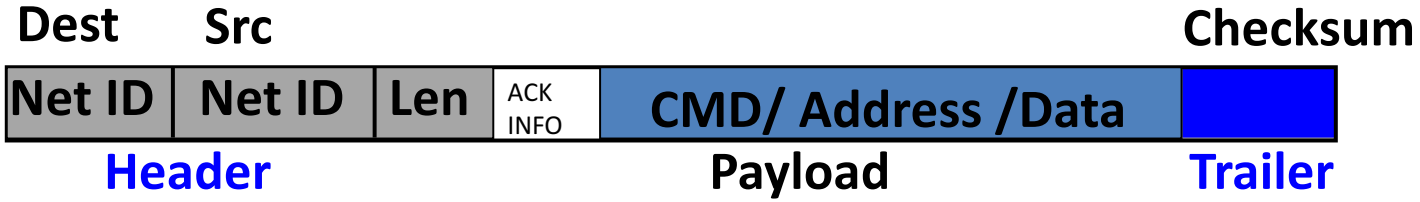


- Ability to name the components and to route packets of information - messages - from a source to a destination

- Layering, redundancy, protocols, and encapsulation as means of abstraction (61C big idea)

Software Protocol to Send and Receive

- SW Send steps
  1: Application copies data to OS buffer
  2: OS calculates checksum, starts timer
  3: OS sends data to network interface HW and says start
- SW Receive steps
  3: OS copies data from network interface HW to OS buffer
  2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
  1: If OK, OS copies data to user address space, & signals application to continue



Networks are like Ogres

https://www.youtube.com/watch?v=_bMcXVe8zIs

Protocols for Networks of Networks?

What does it take to send packets across the globe?

- Bits on wire or air
- Packets on wire or air
- Delivery packets within a single physical network
- Deliver packets across multiple networks
- Ensure the destination received the data
- Create data at the sender and make use of the data at the receiver

Protocol for Networks of Networks?

Lots to do and at multiple levels!

Use abstraction to cope with complexity of communication

- Networks are like ~~ogres~~ onions
  - Hierarchy of layers:
    - Application (chat client, game, etc.)
    - Transport (TCP, UDP)
    - Network (IP)
    - Data Link Layer (ethernet)
    - Physical Link (copper, wireless, etc.)

Protocol Family Concept

- *Protocol*: packet structure and control commands to manage communication
- *Protocol families (suites)*: a set of cooperating protocols that implement the network stack
- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer…

  …but is implemented via services at the next lower level
- Encapsulation: carry higher level information within lower level "envelope"

Inspiration …

- CEO A writes letter to CEO B
  - Folds letter and hands it to assistant
- Assistant:
  - Puts letter in envelope with CEO B's full name
  - Takes to FedEx
- FedEx Office
  - Puts letter in larger envelope
  - Puts name and street address on FedEx envelope
  - Puts package on FedEx delivery truck
- FedEx delivers to other company

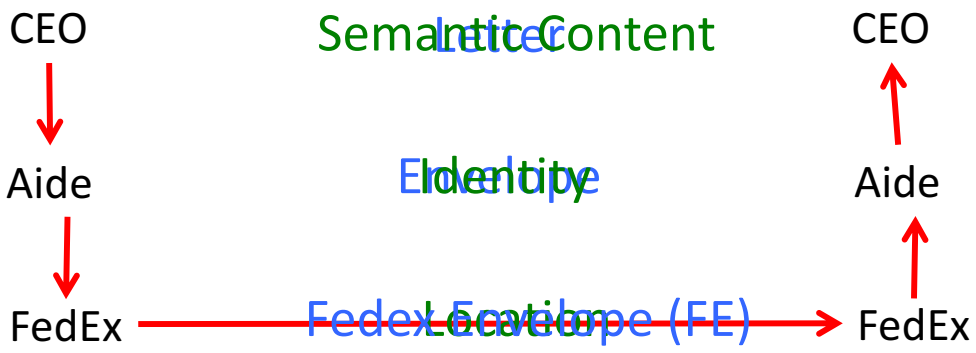
Dear Bill,

Your days are numbered.

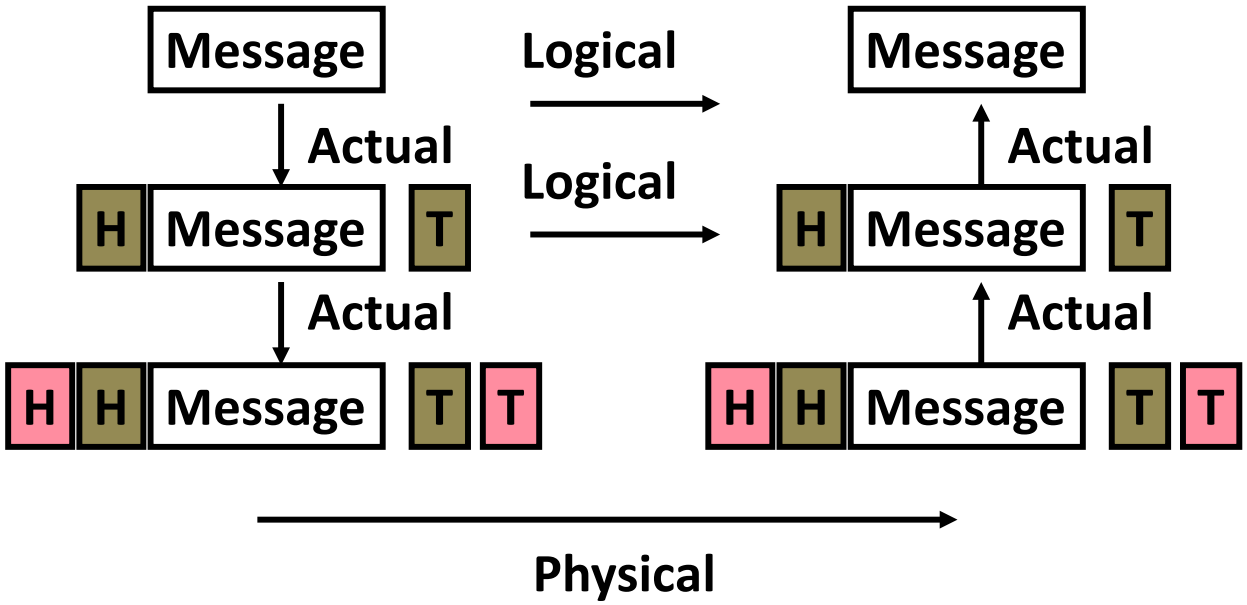
--Steve

The Path of the Letter

"Peers" on each side understand the same things
No one else needs to
Lowest level has most packaging



Protocol Family Concept



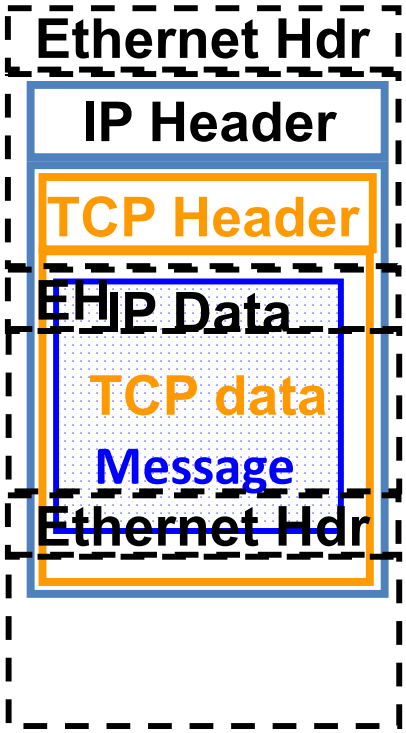
Each lower level of stack "encapsulates" information from layer above by adding header and trailer

Most Popular Protocol for Network of Networks

- Transmission Control Protocol/Internet Protocol (TCP/IP)
- This protocol family is the basis of the Internet, a WAN (wide area network) protocol
  - IP makes best effort to deliver
  - Packets can be lost, corrupted
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN (local area network)

TCP/IP Packet, Ethernet Packet, Protocols

- Application sends message
- TCP breaks into 64KiB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers

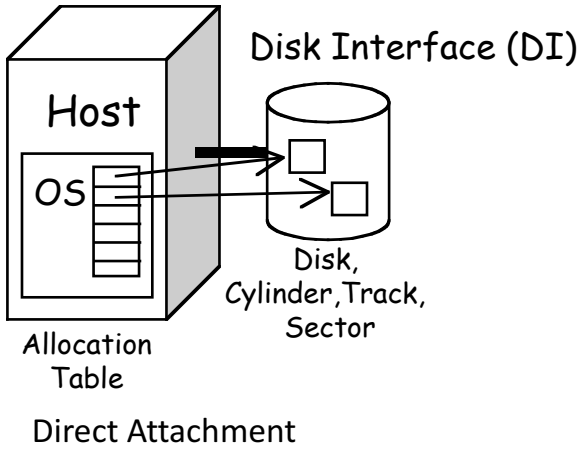


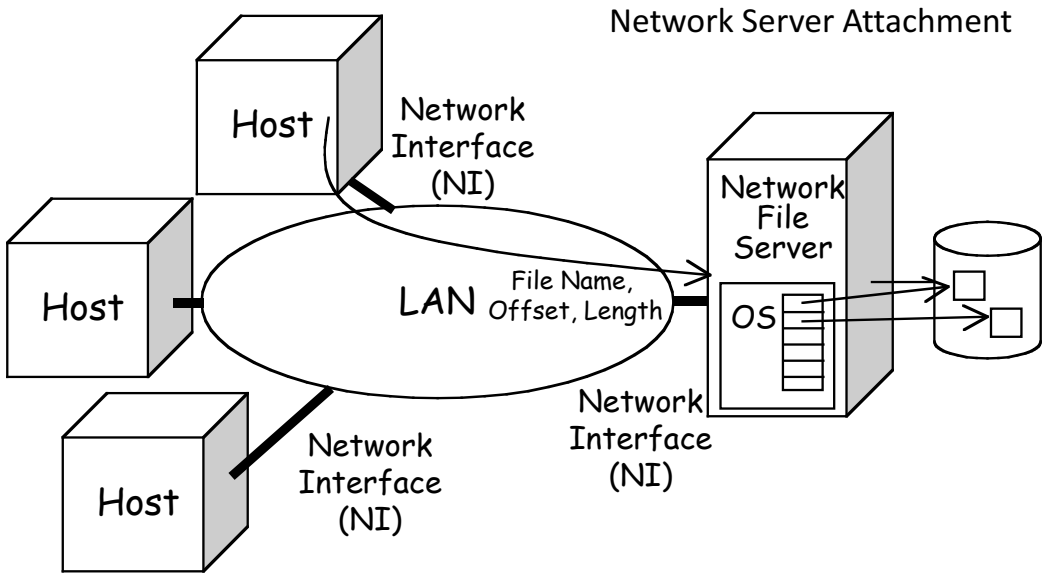

BeCool

Outline

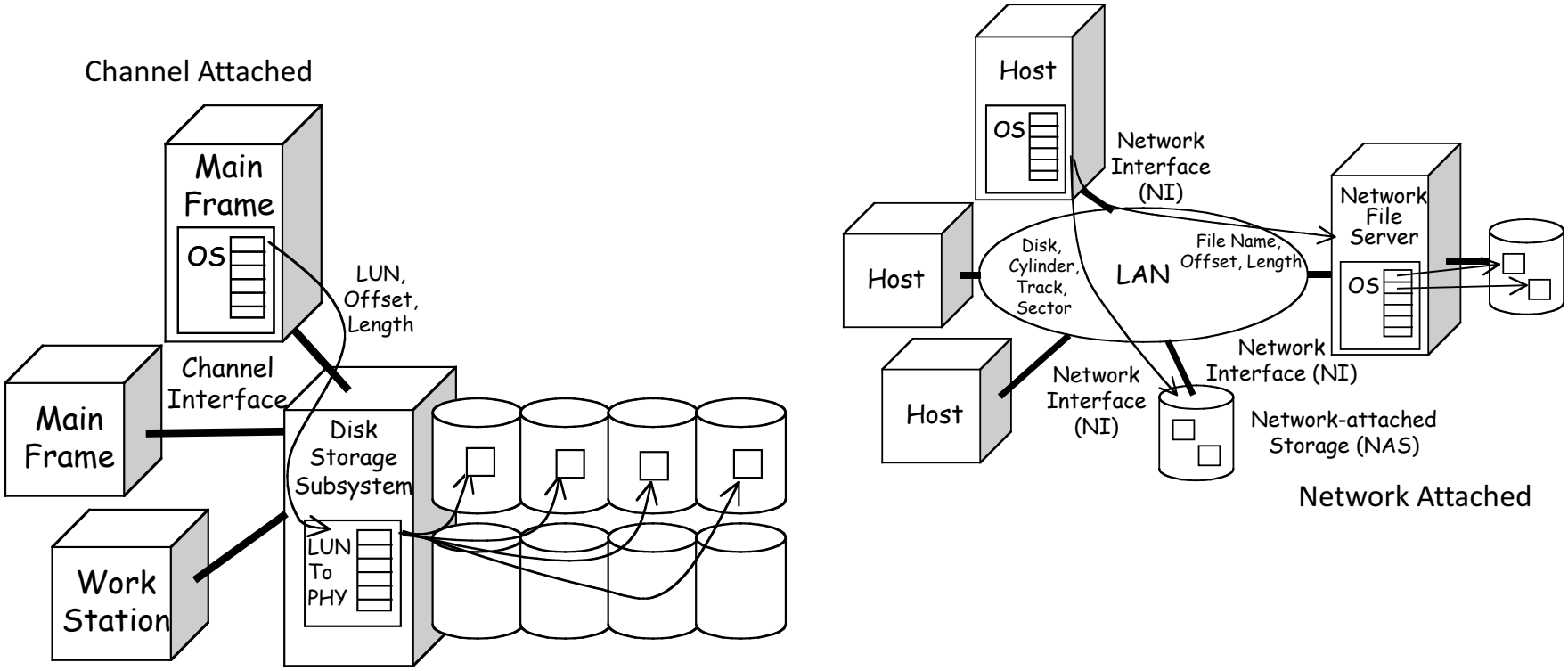
- Direct Memory Access
- Disks
- Networking
- Storage Attachment Evolution
- Rack Scale Memory
- And in Conclusion …

Storage Attachment Evolution

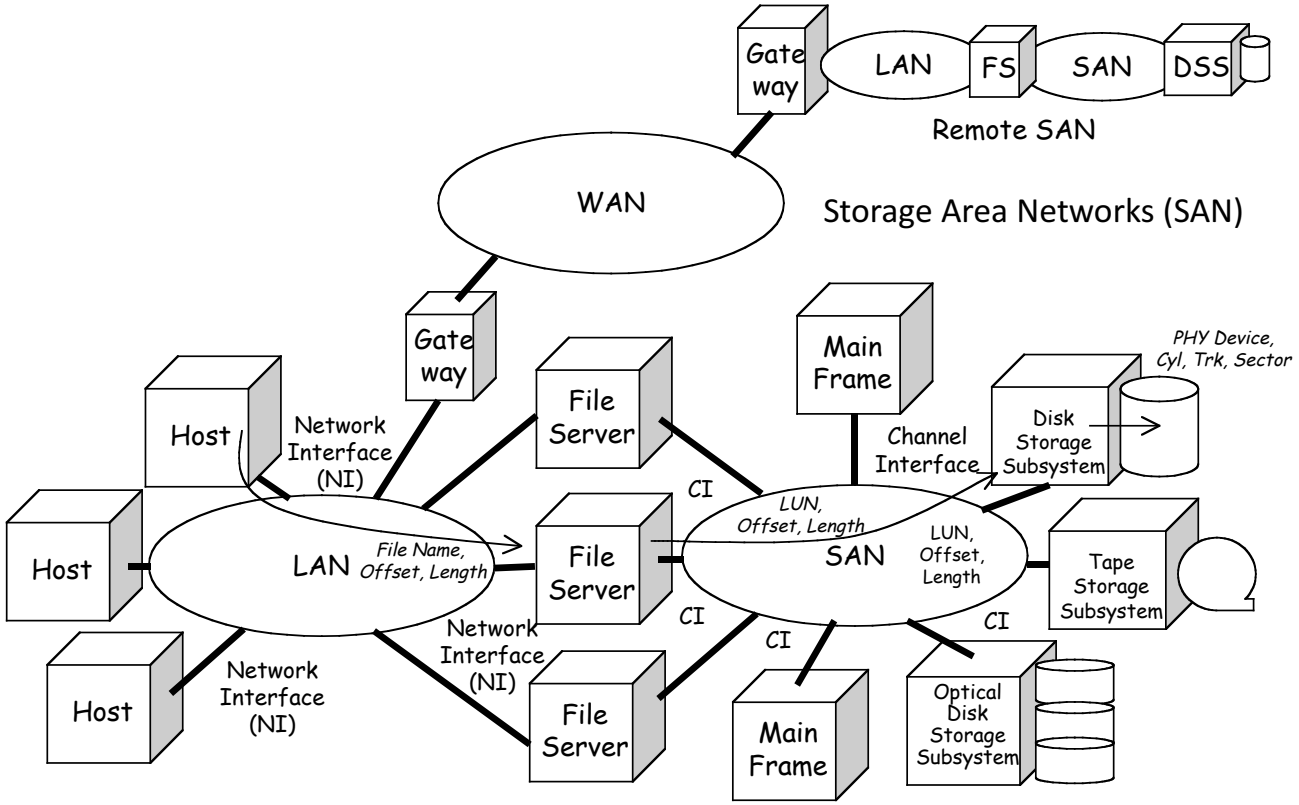




Storage Attachment Evolution



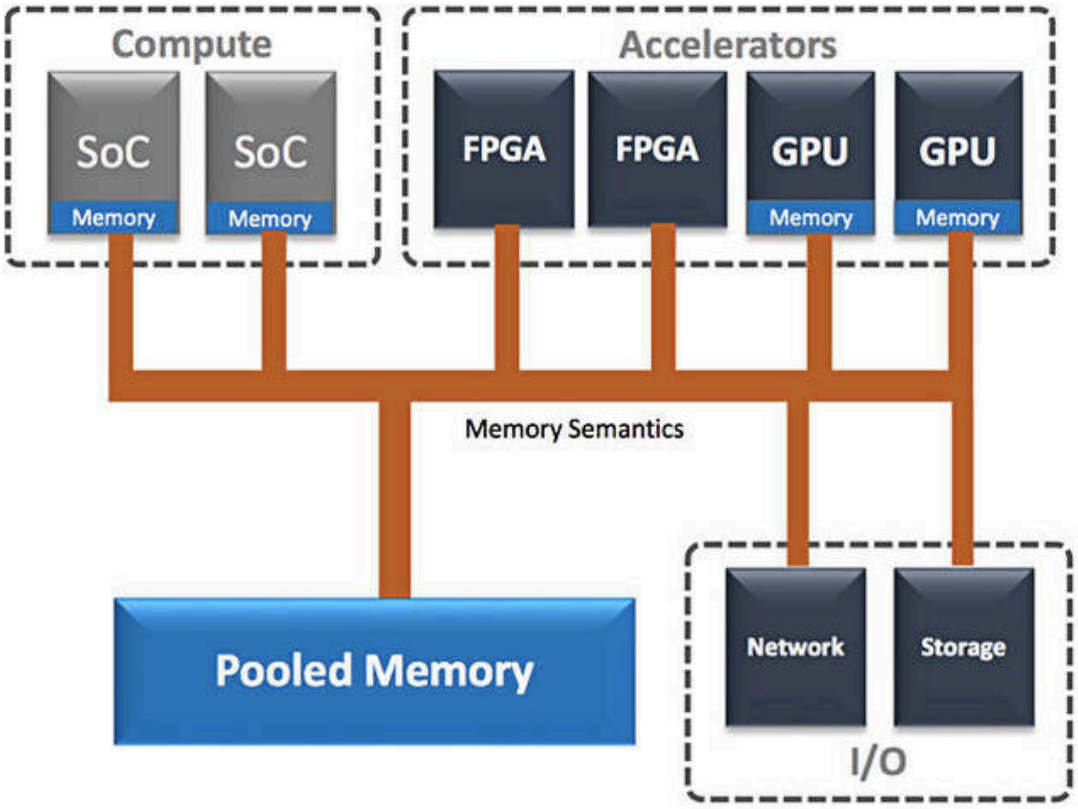
Storage Attachment Evolution



Outline

- Direct Memory Access
- Disks
- Networking
- Storage Attachment Evolution
- Rack Scale Memory
- And in Conclusion …

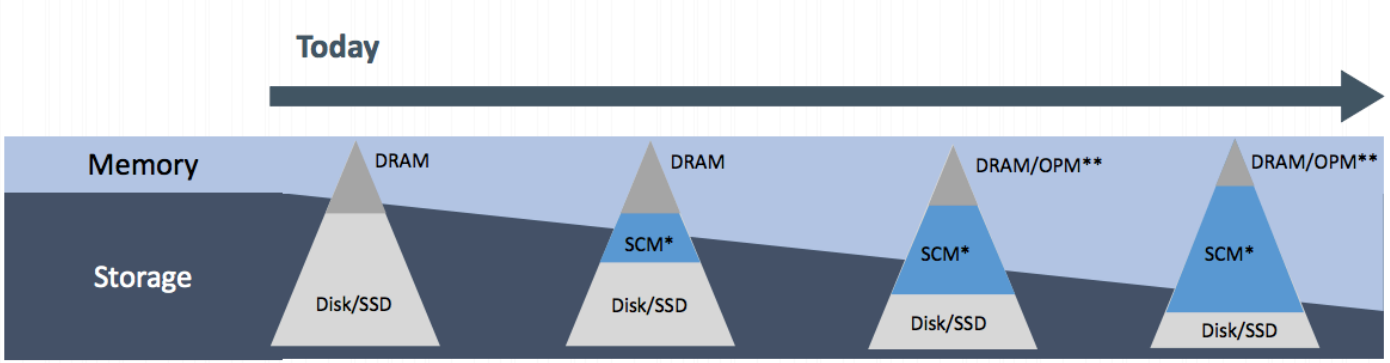
Storage Class Memory
aka Rack-Scale Memory



Storage Class Memory
aka Rack-Scale Memory

- High bandwidth and low latency through a simplified interface based on memory semantics (i.e., ld/st), scalable from tens to several hundred GB/sec of bandwidth, with sub-100 nanosecs load-to-use memory latency
- Supports scalable memory pools and resources for real-time analytics and in-memory applications (i.e., map-reduce)
- Highly software compatible with no required changes to the operating system
- Scales from simple, low cost connectivity to highly capable, rack scale interconnect

Storage Class Memory
aka Rack-Scale Memory

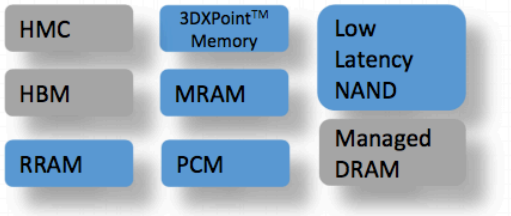


Memory Semantics will be pervasive in Volatile ***AND*** Non-Volatile Storage as these technologies continue to converge.

*SCM = Storage Class Memory

**OPM = On-Package Memory

New and Emerging Memory Technologies

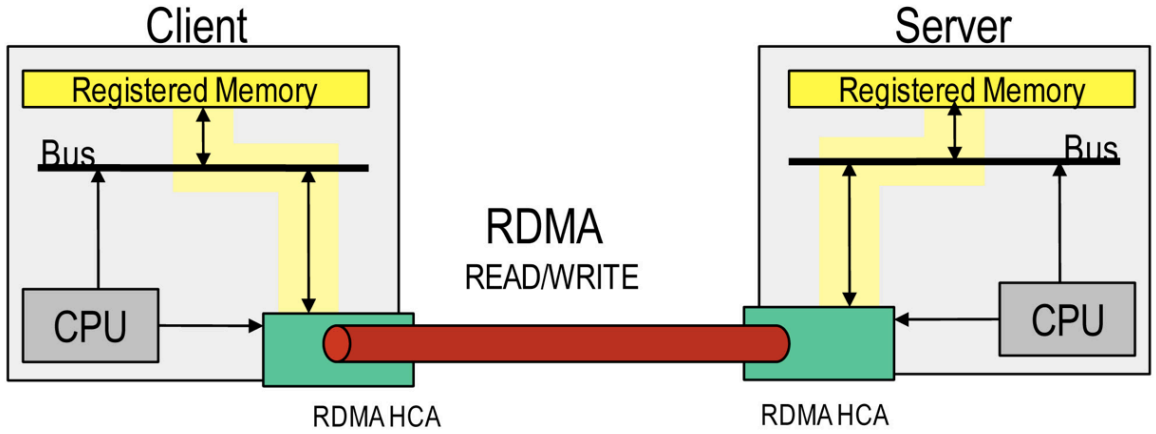


Cheaper than DRAM
More expensive than disk
Non-Volatile and faster than disk

Storage Class Memory
aka Rack Scale Memory

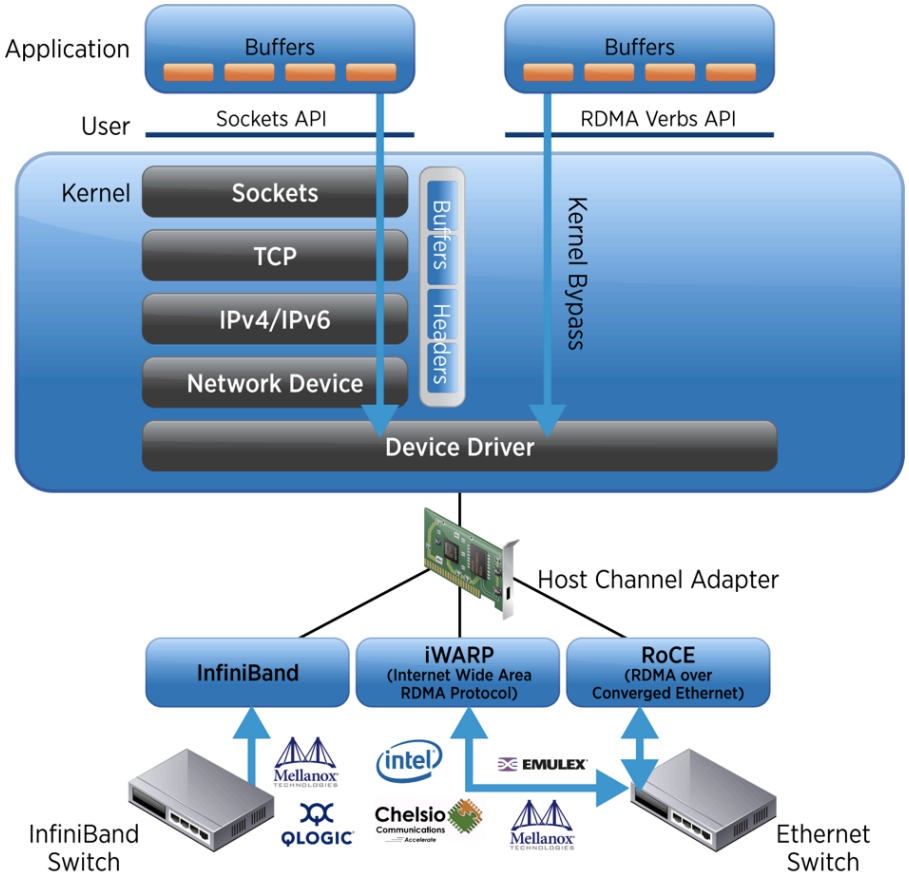
- System memory is flat or shrinking
  - Memory bandwidth per core continues to decrease
  - Memory capacity per core is generally flat
  - Memory is changing on a different cadence compared to the CPU
- Data is growing
  - Data that requires real-time analysis is growing exponentially
  - The value of the analysis decreases if it takes too long to provide insights
- Industry needs an open architecture to solve the problems
  - Memory tiers will become increasingly important
  - Rack-scale composability requires a high bandwidth, low latency fabric
  - Must seamlessly plug into existing ecosystems without requiring OS changes

Remote Direct Memory Access



Remote Direct Memory Access

Conventional Networking



Cut-through Memory access Over network

Outline

- Direct Memory Access
- Disks
- Networking
- Rack Scale Memory
- And, in Conclusion …

"And, in Conclusion…"

- I/O gives computers their 5 senses
- I/O speed range is 100-million to one
- DMA to avoid wasting CPU time on data transfers
- Disks for persistent storage, being replaced by flash and emerging "storage class memory"
- Networks: computer-to-computer I/O
  - Protocol suites allow networking of heterogeneous components. Great Idea: Layers and Abstraction
  - Emerging class: Rack-scale/Storage-class Memory accessible over RDMA or other network interconnect