Storage Systems

INF-2201 Operating Systems Fundamentals – **Spring 2017**

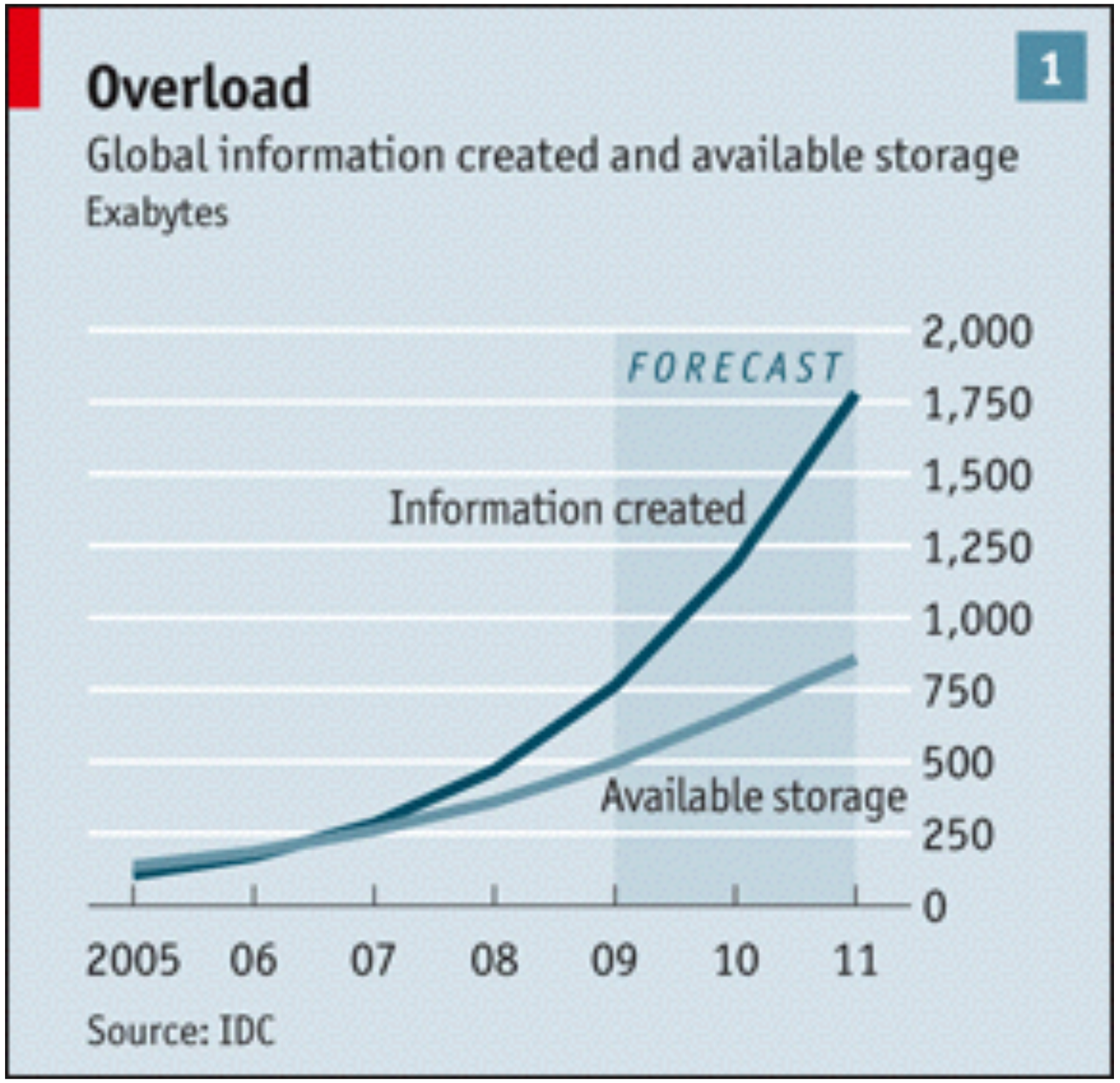
Lars Ailo Bongo, larsab@cs.uit.no

Bård Fjukstad, Daniel Stødle

And Kai Li and Andy Bavier, Princeton
(http://www.cs.princeton.edu/courses/cos318/)

Tanenbaum & Bo,Modern Operating Systems:4th ed.

Increasing volume and use

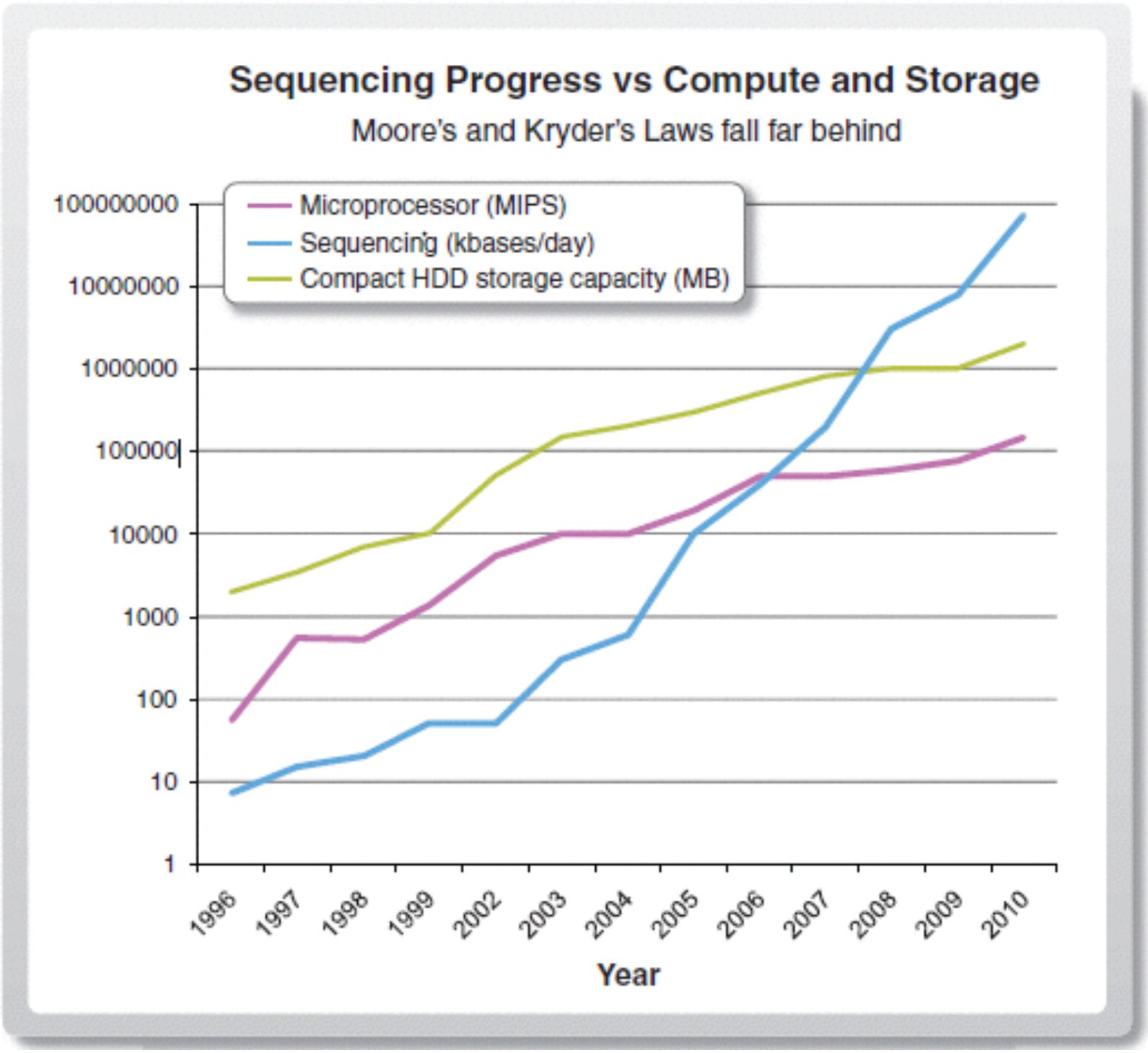


Source: The Economist [http://www.economist.com/node/15557443?story_id=15557443]

Big Data Sources

- Voluntary human produced content
  - Videos, photos, audio…
- Involuntary produced content
  - Online activity logging, tax records…
- Scientific instruments
  - CERN LHC, Sloan Digital Sky Survey, brain simulations, DNA sequencers…

Big Data in Life Science



Storage

- Reliability
  - Archival
  - Reliable
  - Persistent
  - Temporal
- Access pattern
  - Write or read intensive
  - Sequential or random access
  - Low-latency or high throughput
- Cost
- Power

The Memory Hierarchy

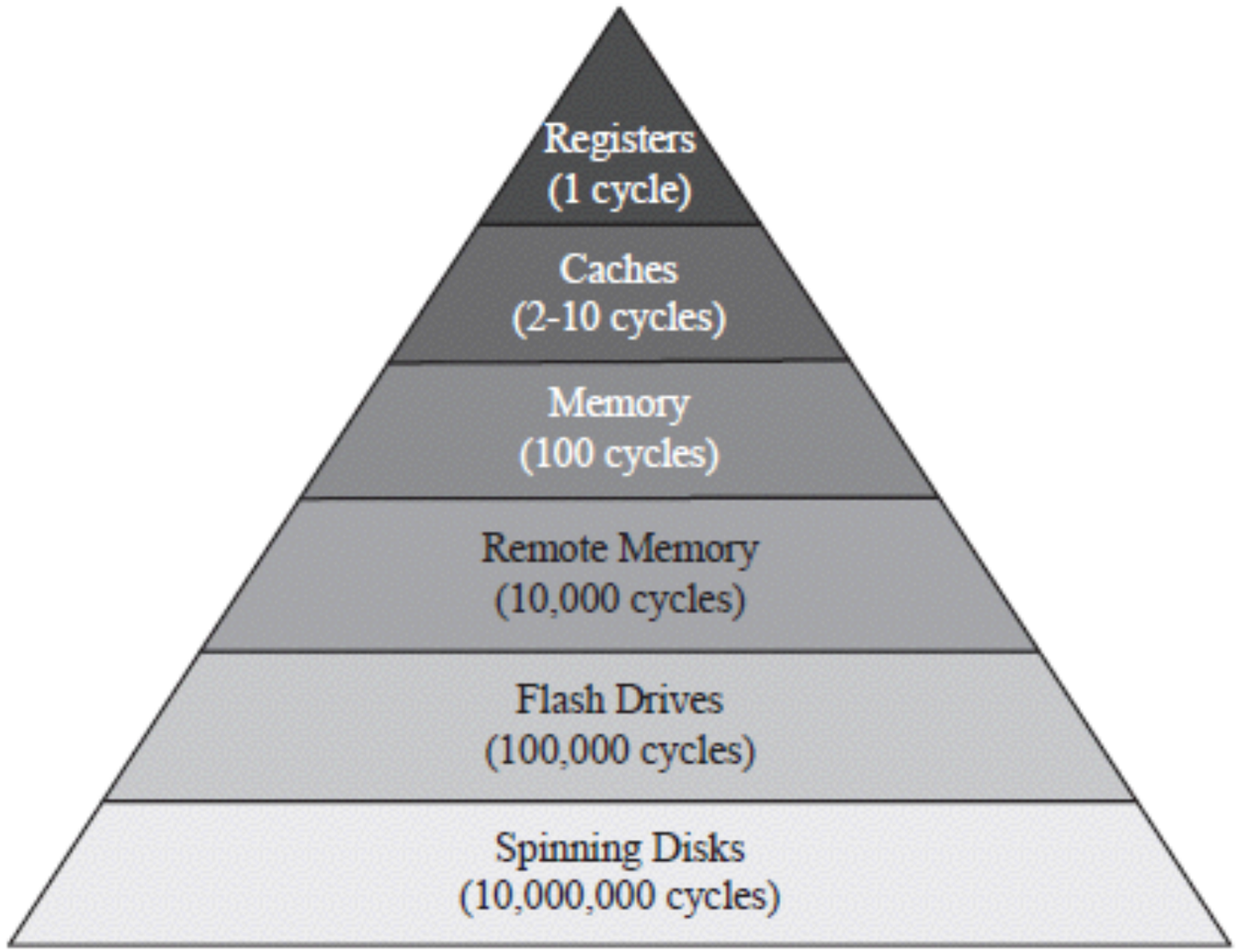


Figure 1. The memory hierarchy. *Each level shows the typical access latency in processor cycles. Note the five-orders-of-magnitude gap between main memory and spinning disks.*

Jiahua He, Arun Jagatheesan, Sandeep Gupta, Jeffrey Bennett, Allan Snavely, "**DASH: a Recipe for a Flash-based Data Intensive Supercomputer,**" sc, pp.1-11, 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, 2010

Disk vs. Flash vs. DRAM

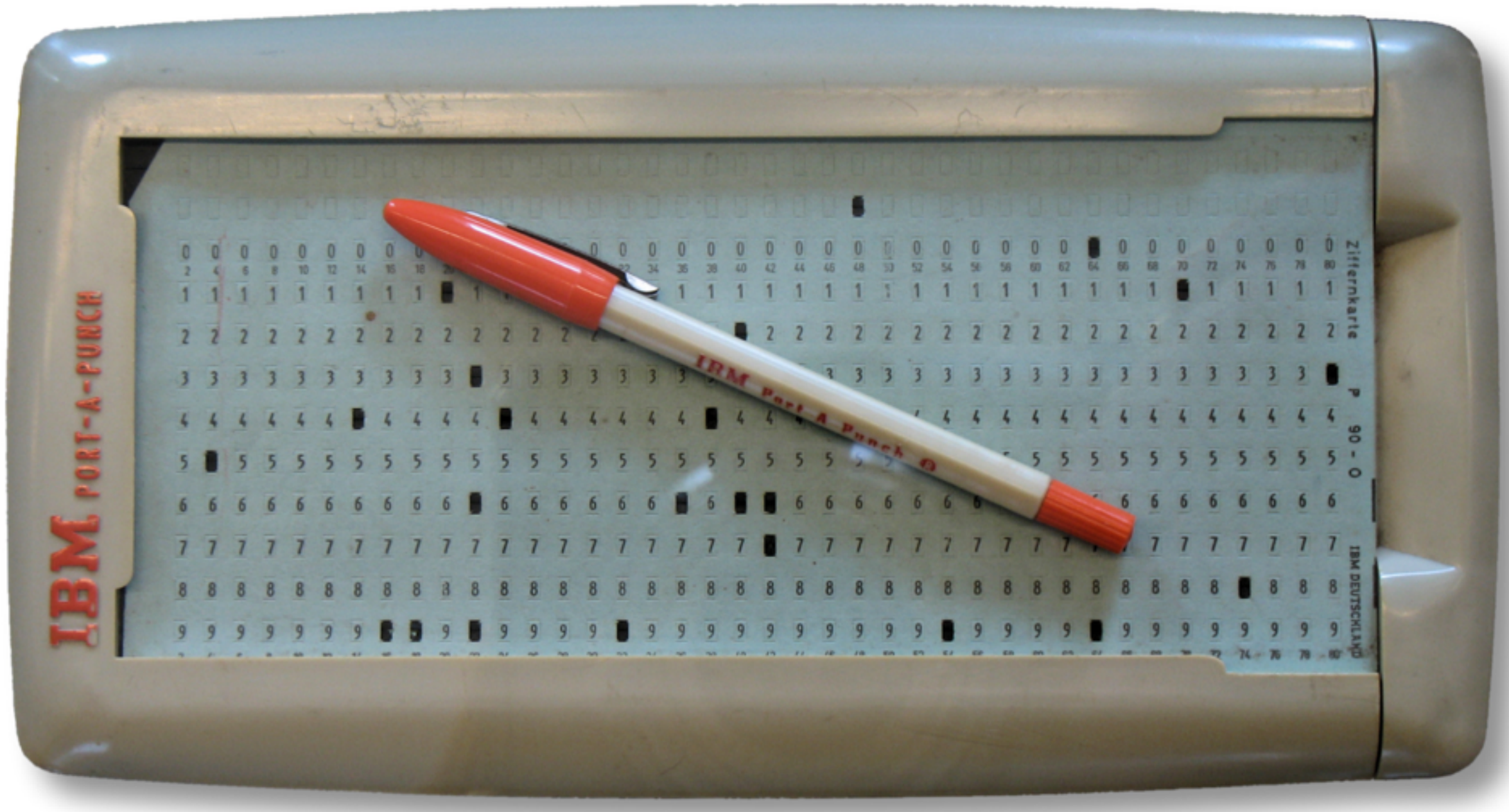
	Disk	Flash	DRAM
Access time (relative)	1	100-1,000	0.000001 (1 / 100,000)
Cost (relative)	1	15-25	30-150
Bandwidth (relative)	1	1	80
Bandwidth/ GB (relative)	1		6,000
Bandwidth/ $ (relative)	1		160

Source: Computer Architecture A Quantitative Approach

Overview

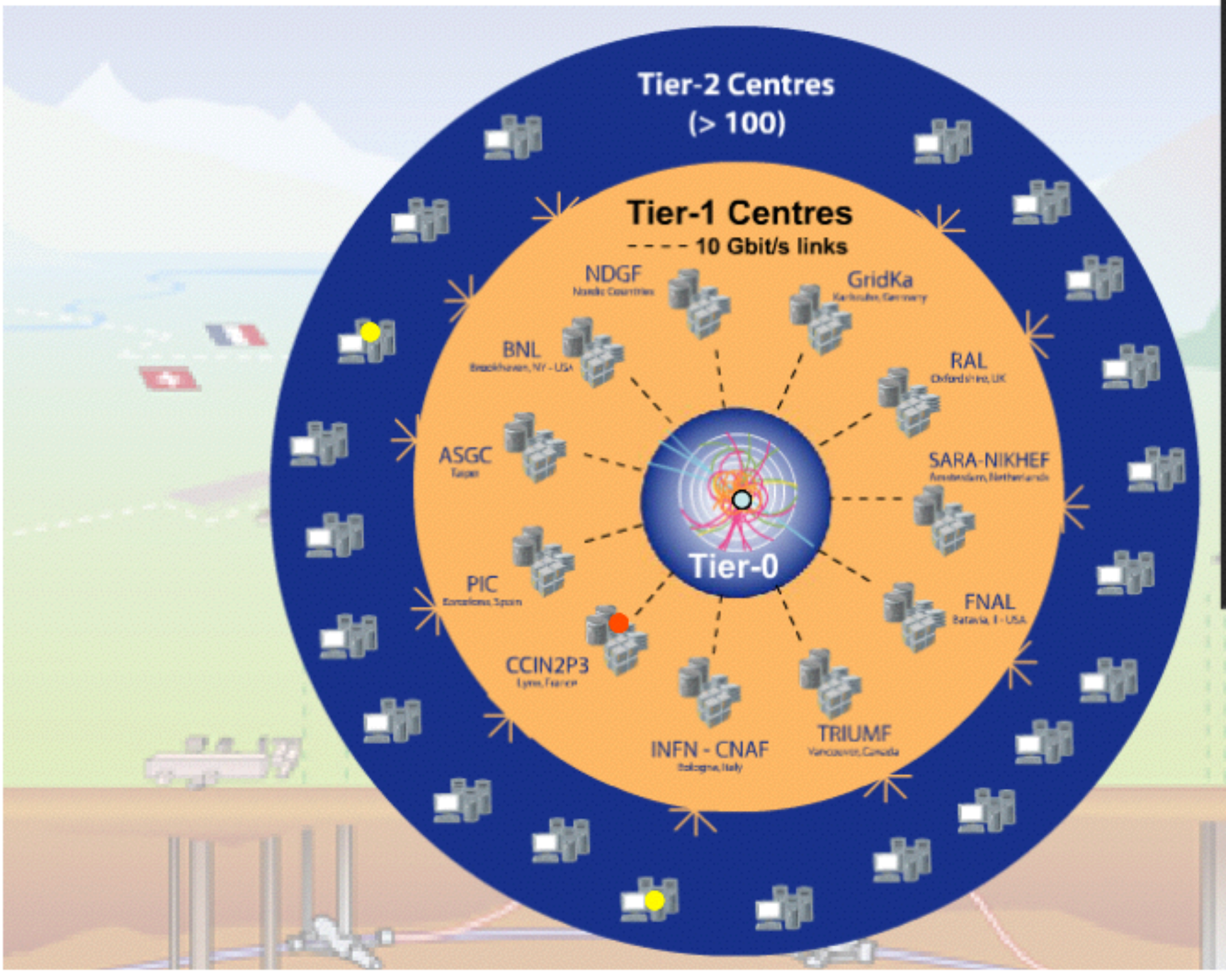
- Magnetic disks
- Disk arrays
- Flash storage
- DRAM storage
- Storage hierarchy

Punch Cards


IBM PORT-A-PUNCH
Zifternkarte
IBM Port A Punch
IBM DEUTSCHLAND

Magnetophone

Tape Library



Tier-0 (CERN):
- Data recording
- Initial data reconstruction
- Data distribution

Tier-1 (11 centres):
- Permanent storage
- Re-processing
- Analysis

Tier-2 (~130 centres):
- Simulation
- End-user analysis

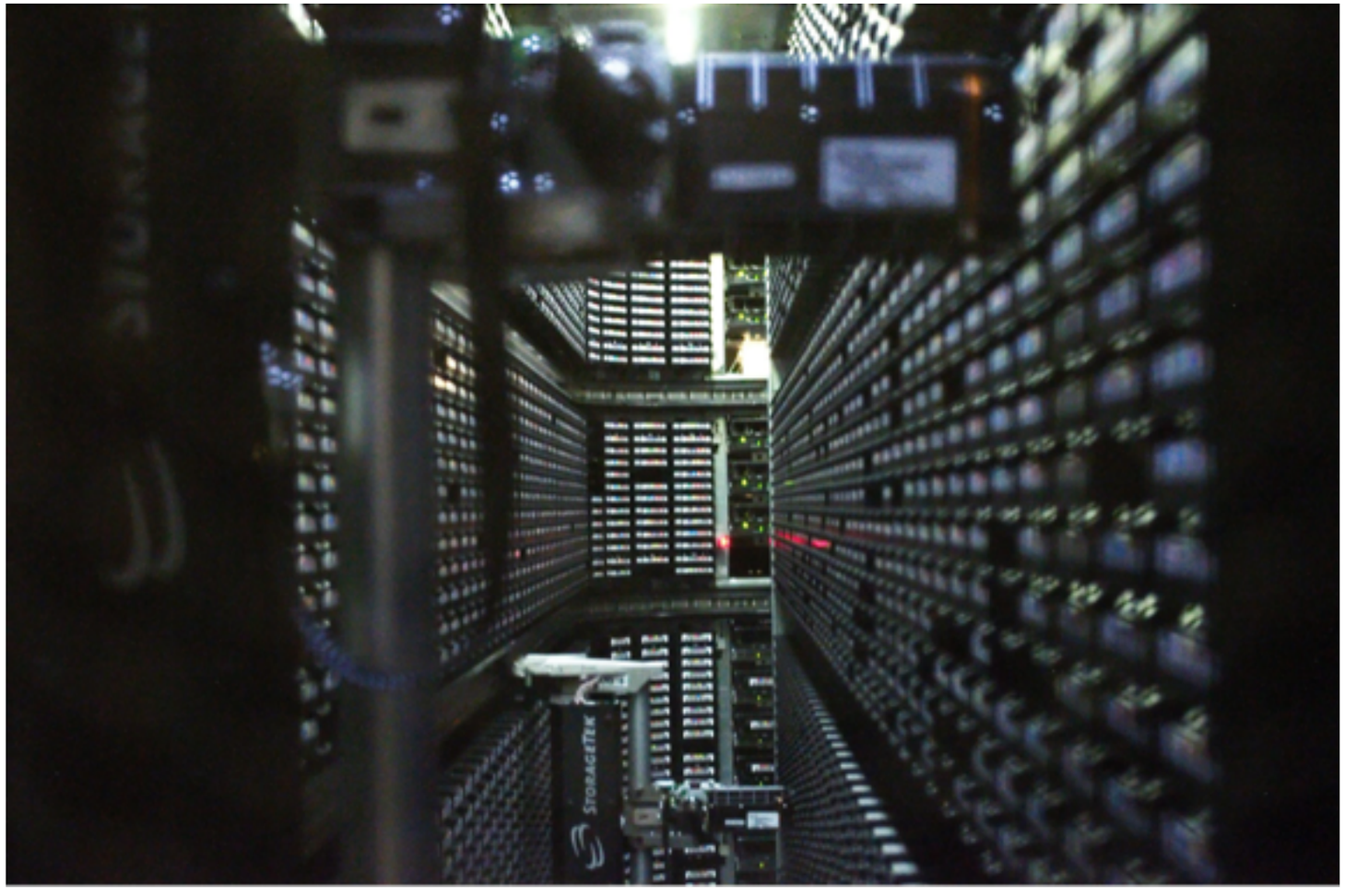
22

Source: http://www.usenix.org/events/lisa10/tech/slides/cass.pdf

CERN > 100 PB (petabytes)

ECMWF

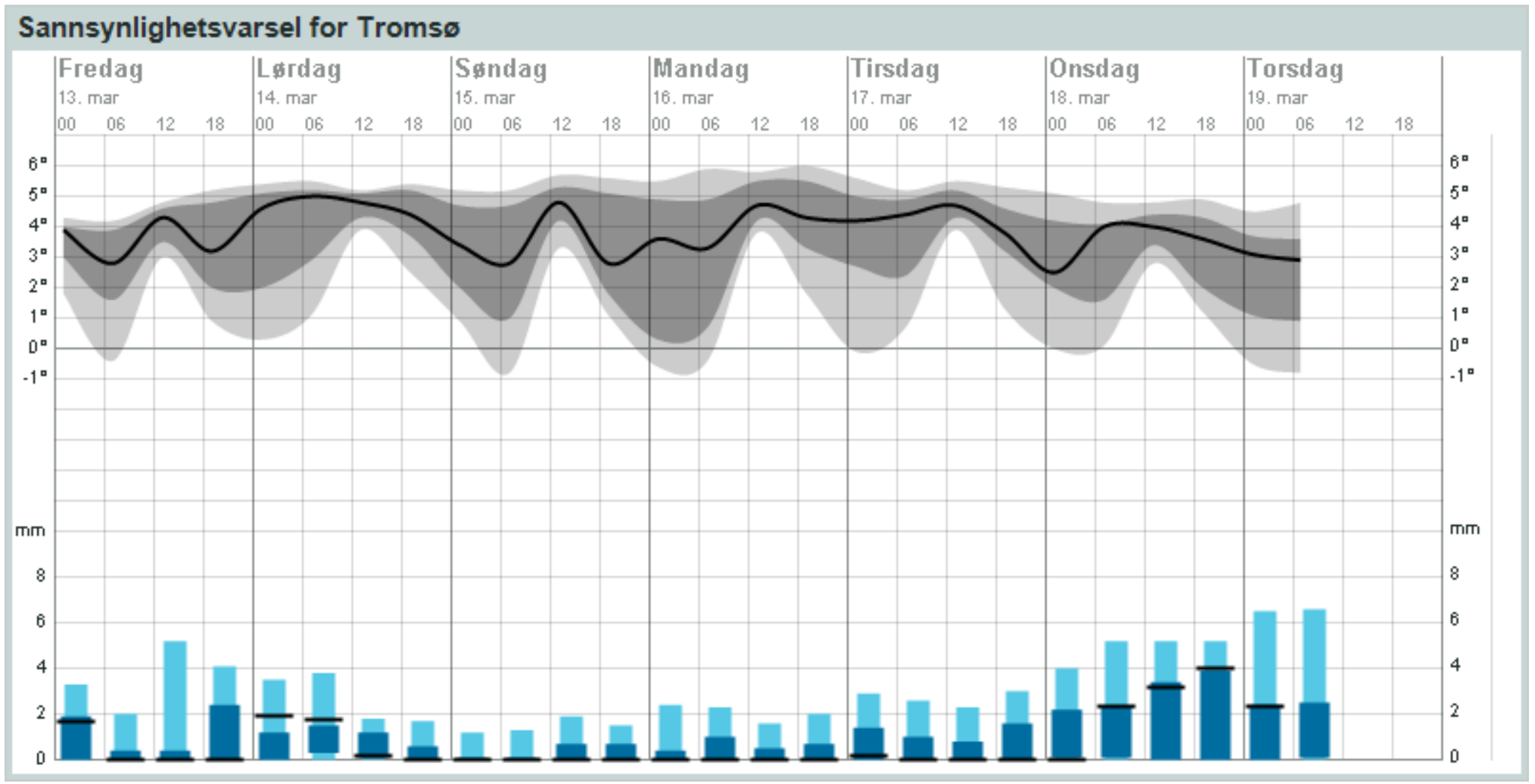
European Centre for Medium-Range Weather Forecasts



9,000 tape mounts/day
65 TB/day added
59 PB of primary data + 14 PB backups

Detour: ECMWF
European Centre for Medium-Range Weather Forecasts

Detour:
Products from the ECMWF



Long-term weather forecast for Scandinavia.
Both short- and long-term forecasts for the whole globe

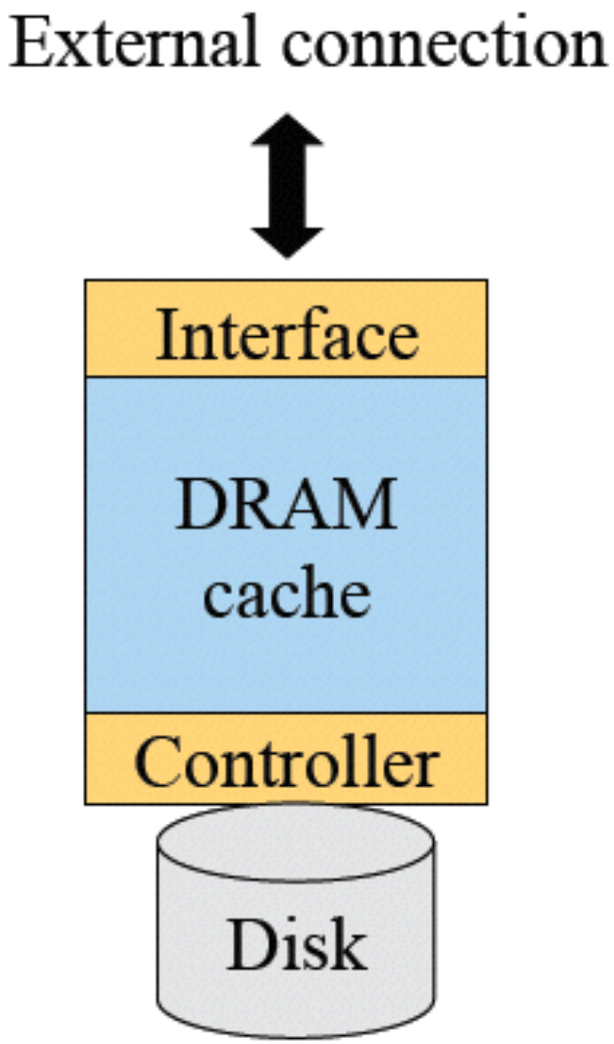
Hard Drive Teardown

- http://www.youtube.com/watch?v=Wiy_eHdj8kg

«The Engineer Guy»

A Typical Magnetic Disk Controller

- External connection
  - Parallel ATA (aka IDE or EIDE), Serial ATA, SCSI, Serial Attached SCSI (SAS), Fibre Channel, FireWire, USB
- Cache
  - Buffer data between disk and interface
- Controller
  - Read/write operation
  - Cache replacement
  - Failure detection and recovery

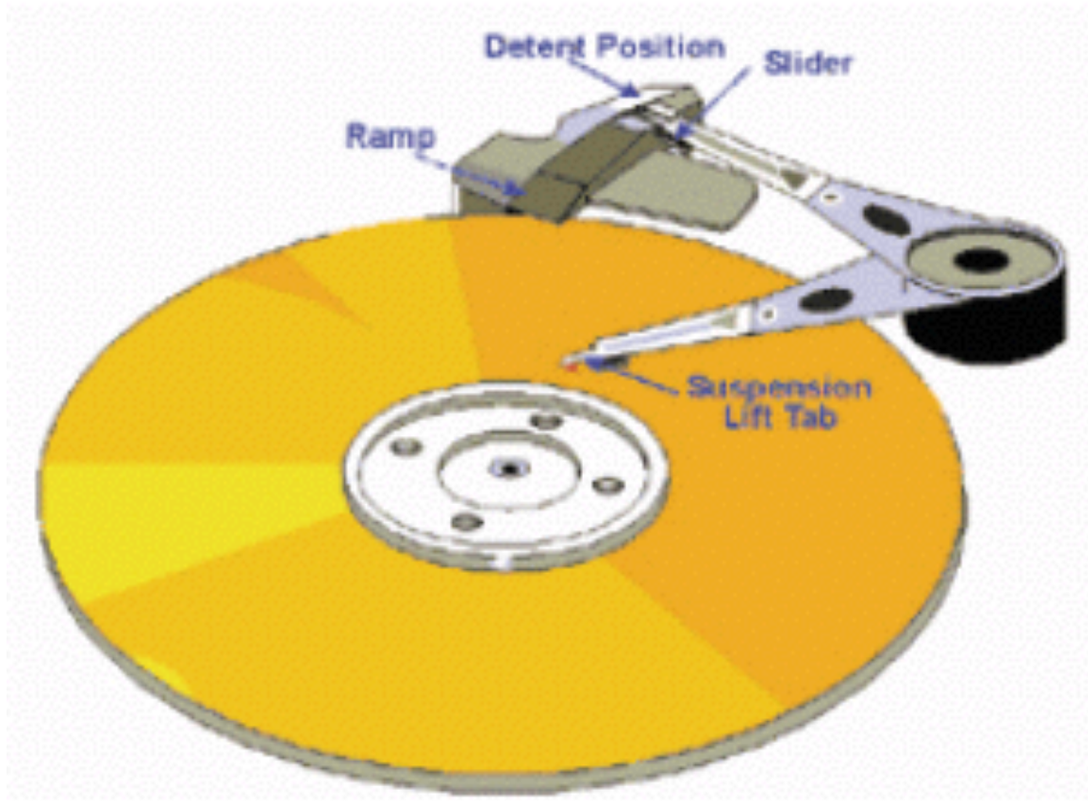


Disk Caching

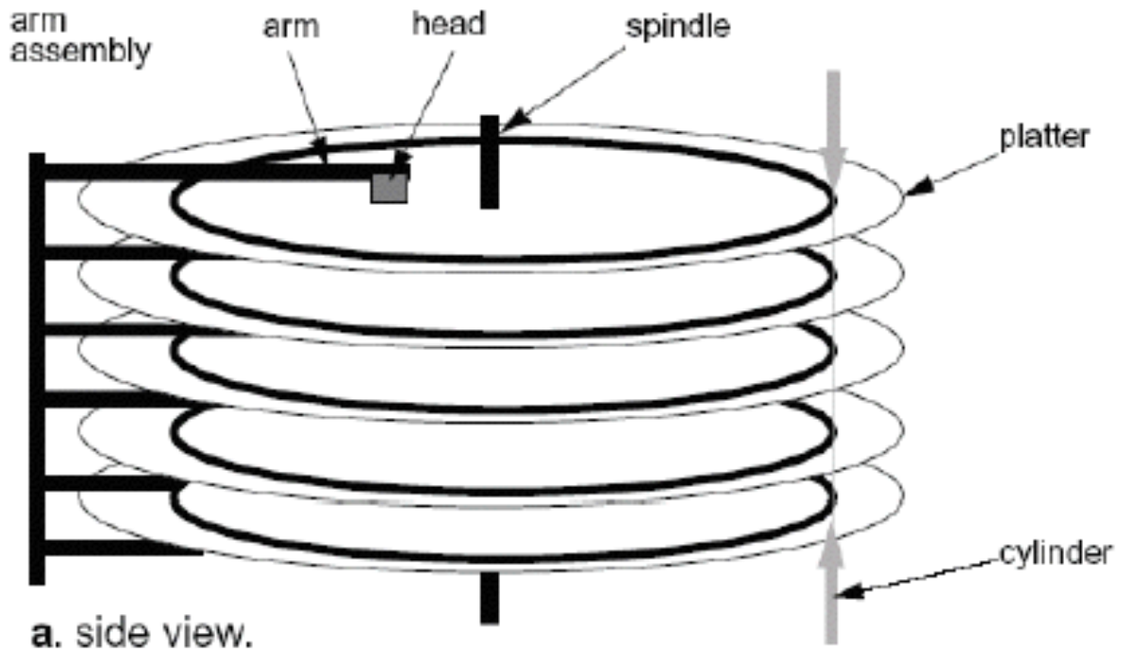
- Method
  - Use DRAM to cache recently accessed blocks
    - Most disks has 32MB
    - Some of the RAM space stores “firmware” (an embedded OS)
    - Blocks are replaced usually in an LRU order
- Pros
  - Good for reads if accesses have locality
- Cons
  - Cost
  - Need to deal with reliable writes

Disk Arm and Head

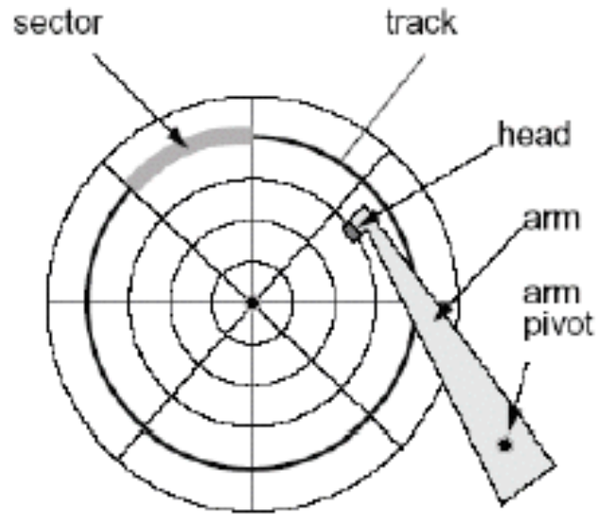
- Disk arm
  - A disk arm carries disk heads
- Disk head
  - Mounted on an actuator
  - Read and write on disk surface
- Read/write operation
  - Disk controller receives a command with <track#, sector#>
  - Seek the right cylinder (tracks)
  - Wait until the right sector comes
  - Perform read/write



Mechanical Component of A Disk Drive



a. side view.

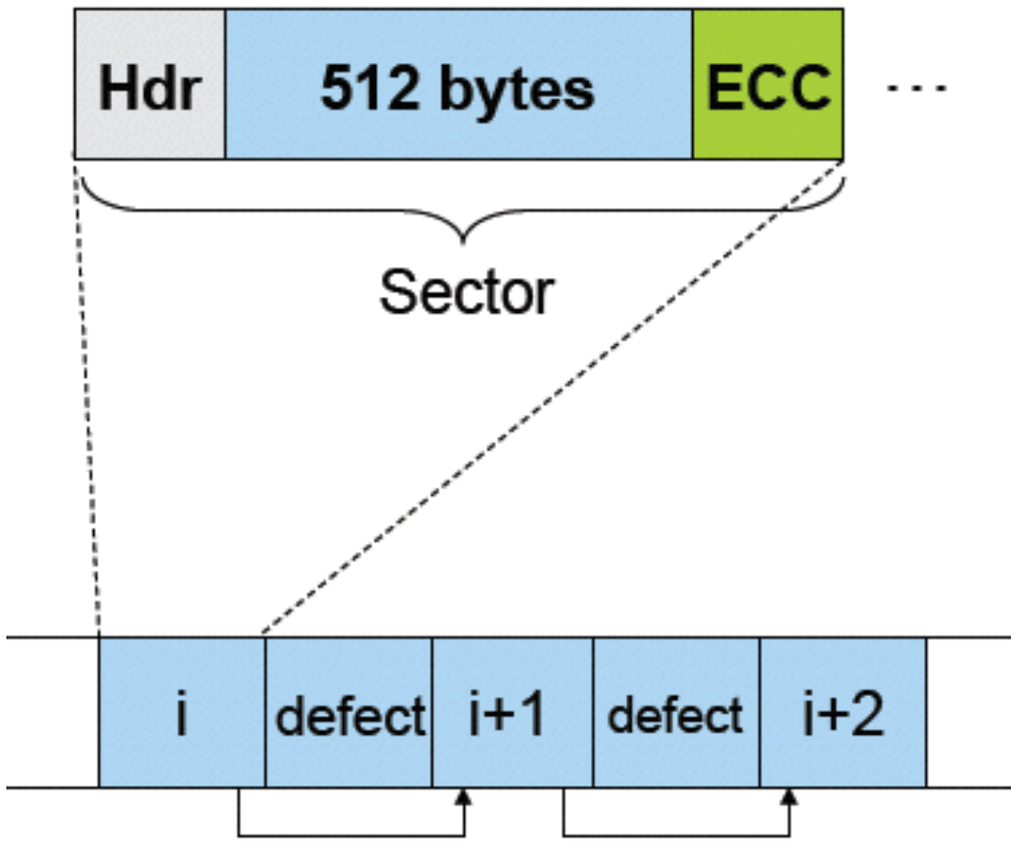


b. top view.

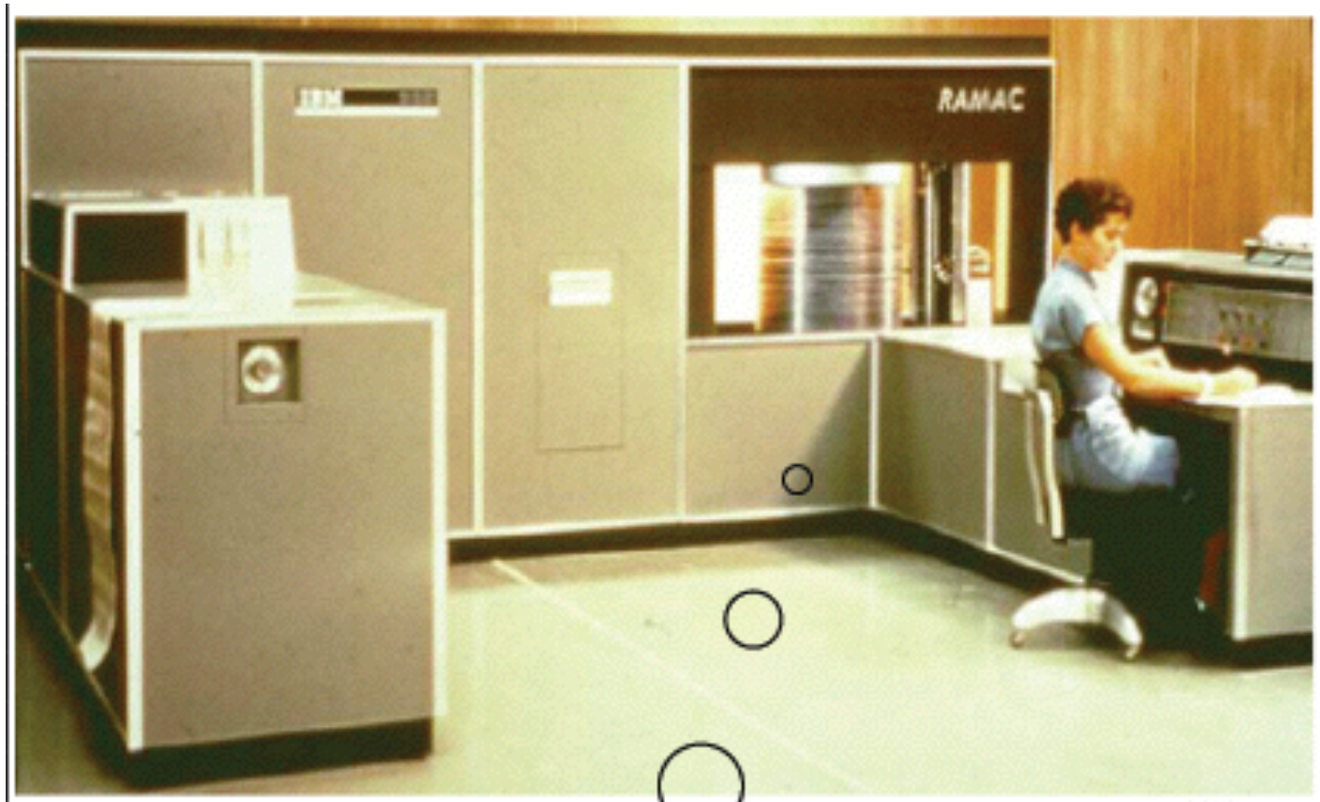
- Tracks
  - Concentric rings around disk surface, bits laid out serially along each track
- Cylinder
  - A track of the platter, 1000-5000 cylinders per zone, 1 spare per zone
- Sectors
  - Each track is split into arc of track (min unit of transfer)

Disk Sectors

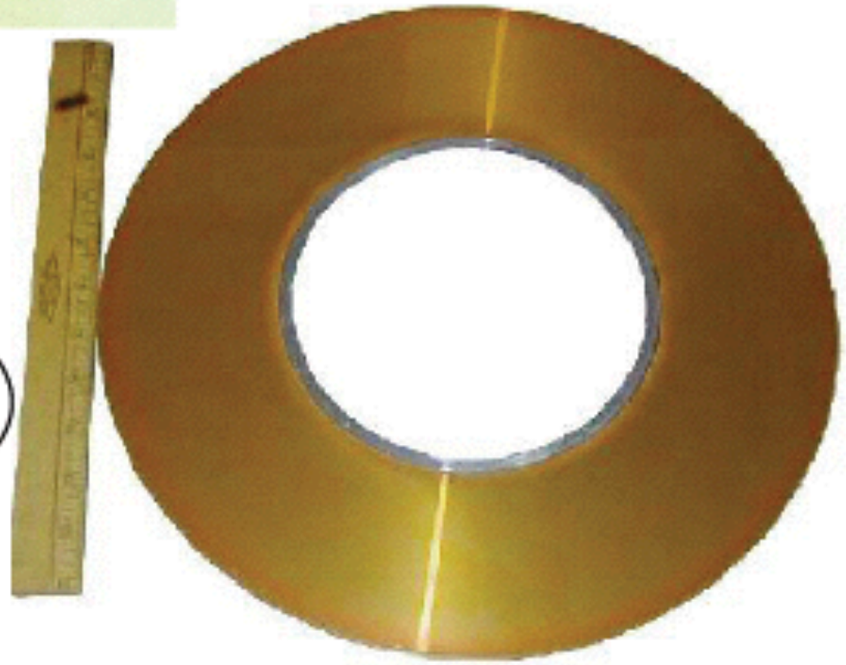
- Where do they come from?
  - Formatting process
  - Logical maps to physical
- What is a sector?
  - Header (ID, defect flag, …)
  - Real space (e.g. 512 bytes)
  - Trailer (ECC code)
- What about errors?
  - Detect errors in a sector
  - Correct them with ECC
  - If not recoverable, replace it with a spare
  - Skip bad sectors in the future



Disks Were Large



First Disk:
IBM 305 RAMAC (1956)
5MB capacity
50 disks, each 24”



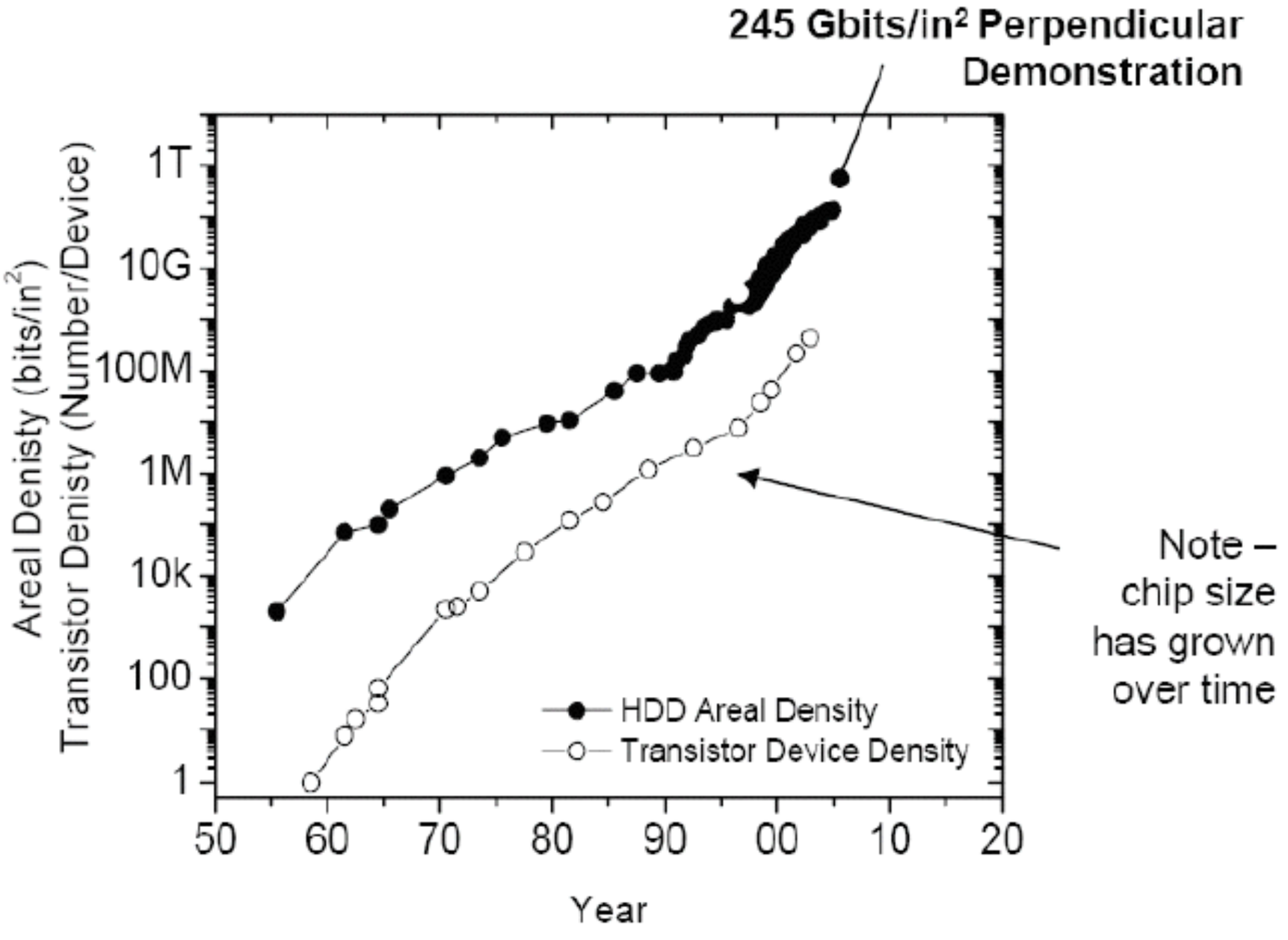
They Are Now Much Smaller

Toshiba Mobile 2,5" 2TB
SATA 3Gb/s, 8MB Cache, S.M.A.R.T 5400RPM, 15mm, 2.5»

komplett.no
1.379,-

2,5” approx 64 x 100 x 15 mm

Areal Density vs. Moore's Law



(Mark Kryder at SNW 2006)

50 Years Later (Mark Kryder at SNW 2006)

	IBM RAMAC (1956)	Seagate Momentus (2006)	Difference
Capacity	5MB	160GB	32,000
Areal Density	2K bits/in^2	130 Gbits/in^2	65,000,000
Disks	50 @ 24" diameter	2 @ 2.5" diameter	1 / 2,300
Price/MB	$1,000	$0.01	1 / 3,200,000
Spindle Speed	1,200 RPM	5,400 RPM	5
Seek Time	600 ms	10 ms	1 / 60
Data Rate	10 KB/s	44 MB/s	4,400
Power	5000 W	2 W	1 / 2,500
Weight	~ 1 ton	4 oz	1 / 9,000

Sample Disk Specs (from Seagate)

	Cheetah 15k.7	Barracuda XT
Capacity		
Formatted capacity	600	2000
Discs	4	4
Heads	8	8
Sector size (bytes)	512	512
Performance		
External interface	Ultra320 SCSI, FC, S.	SATA
Spindle speed (RPM)	15,000	7,200
Average latency	2	4.16
Seek time, read/write	3.5/3.9	8.5/9.5
Track-to-track read/	0.2-0.4	0.8/1.0
Internal transfer (MB/	1,450-2,370	600
Transfer rate (MB/sec)	122-204	138
Cache size (MB)	16	64
Reliability		
Recoverable read	1 per 1012 bits	1 per 1010 bits
Non-recoverable read	1 per 1016 bits	1 per 1014 bits

Disk Performance (2TB disk)

- Seek
  - Position heads over cylinder, typically 3.5-9.5 ms
- Rotational delay
  - Wait for a sector to rotate underneath the heads
  - Typically 8 - 4 ms (7,200 – 15,000RPM) or . rotation takes 4 - 2ms
- Transfer bytes
  - Transfer bandwidth is typically 40-138 Mbytes/sec
- Performance of transfer 1 Kbytes
  - Seek (4 ms) + half rotational delay (2ms) + transfer (0.013 ms)
  - Total time is 6.01 ms or 167 Kbytes/sec (1/360 of 60MB/sec)!

More on Performance

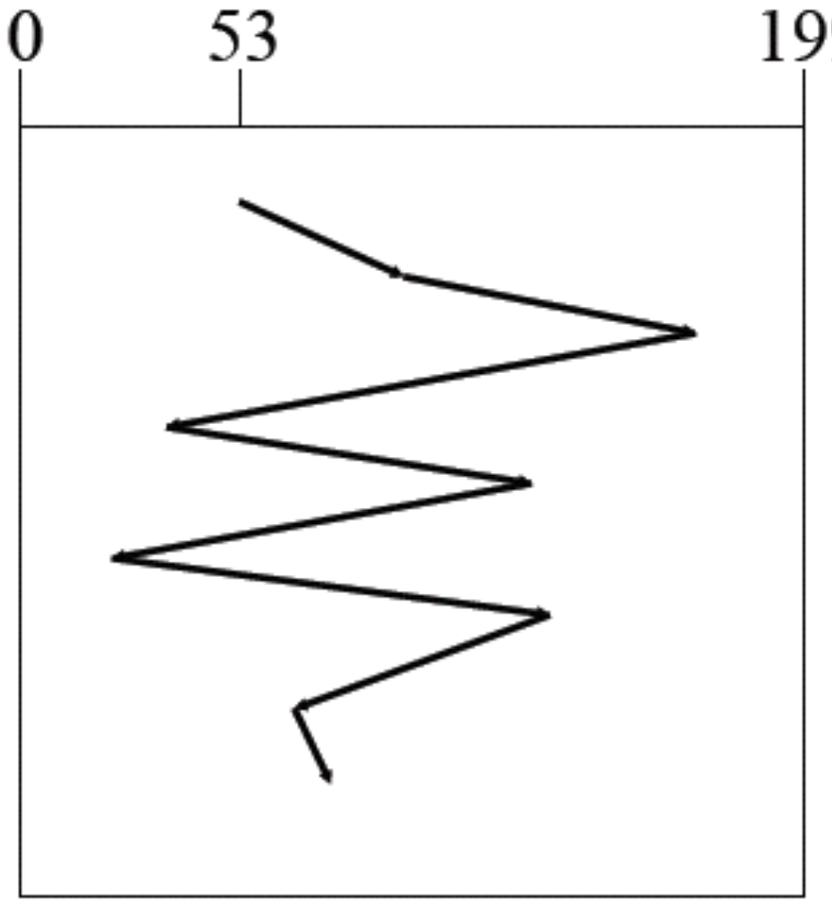
- What transfer size can get 90% of the disk bandwidth?
  - Assume Disk BW = 60MB/sec, . rotation = 2ms, . seek = 4ms
  - BW * 90% = size / (size/BW + rotation + seek) size = BW * (rotation + seek) * 0.9 / 0.1= 60MB * 0.006 * 0.9 / 0.1 = 3.24MB

Block Size	% of Disk Transfer Bandwidth
1Kbytes	0.28%
1Mbytes	73.99%
3.24Mbytes	90%

- Seek and rotational times dominate the cost of small accesses
  - Disk transfer bandwidth are wasted
  - Need algorithms to reduce seek time
- Speed depends on which sectors to access
  - Are outer tracks or inner tracks faster?

FIFO (FCFS) order

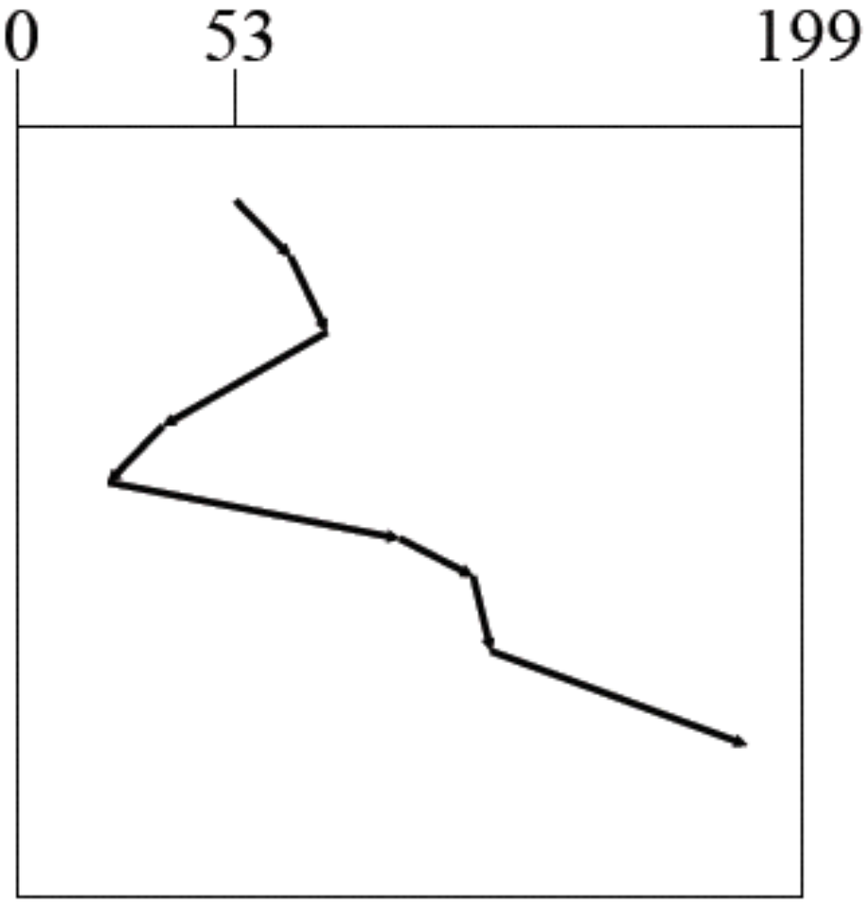
- Method
  - First come first serve
- Pros
  - Fairness among requests
  - In the order applications expe
- Cons
  - Arrival may be on random spo seeks)
  - Wild swing can happen



98, 183, 37, 122, 14, 124, 65, 67

SSTF (Shortest Seek Time First)

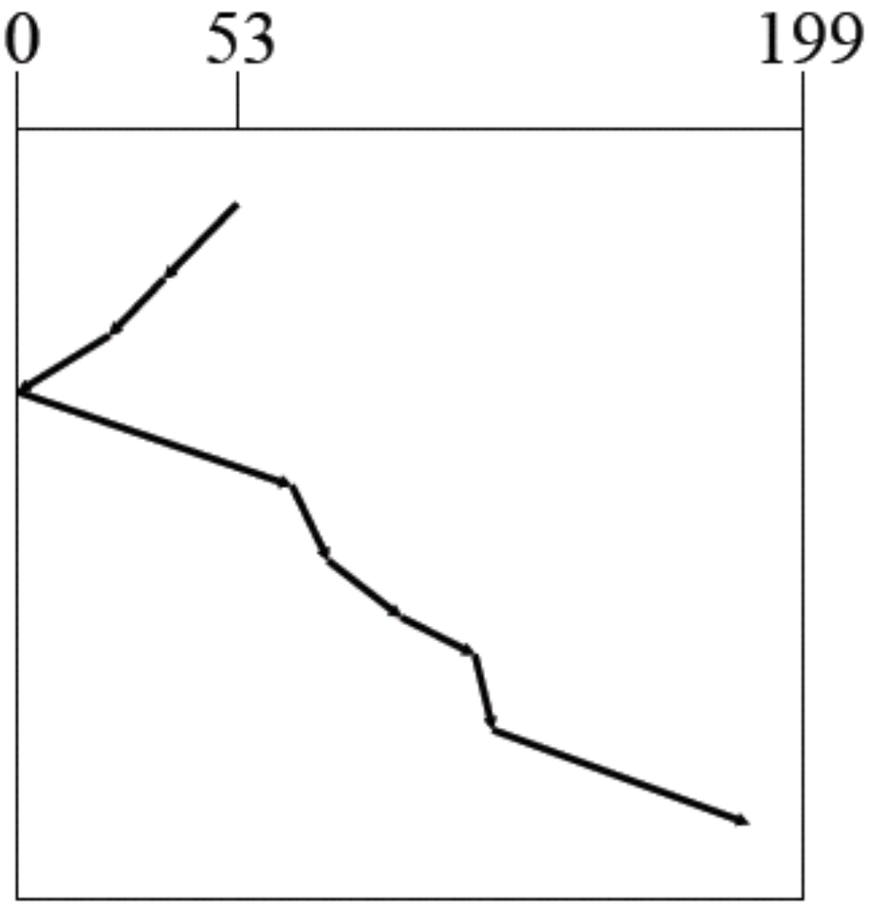
- Method
  - Pick the one closest on disk
  - Rotational delay is in calculati
- Pros
  - Try to minimize seek time
- Cons
  - Starvation
- Question
  - Is SSTF optimal?
  - Can we avoid the starvation?



98, 183, 37, 122, 14, 124, 65, 67
(65, 67, 37, 14, 98, 122, 124, 183)

Elevator (SCAN)

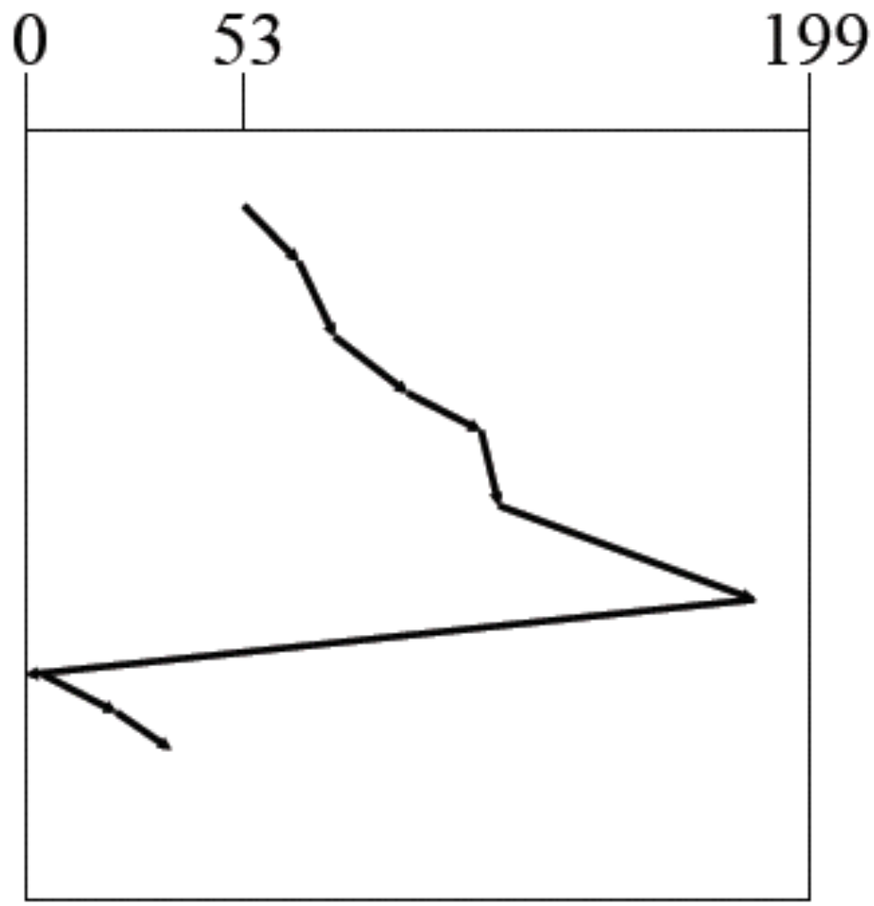
- Method
  - Take the closest request in the
  - Real implementations do not
    LOOK)
- Pros
  - Bounded time for each reque
- Cons
  - Request at the other end will



98, 183, 37, 122, 14, 124, 65, 67
(37, 14, 65, 67, 98, 122, 124, 183)

C-SCAN (Circular SCAN)

- Method
  - Like SCAN
  - But, wrap around
  - Real implementation doesn't g
- Pros
  - Uniform service time
- Cons
  - Do nothing on the return



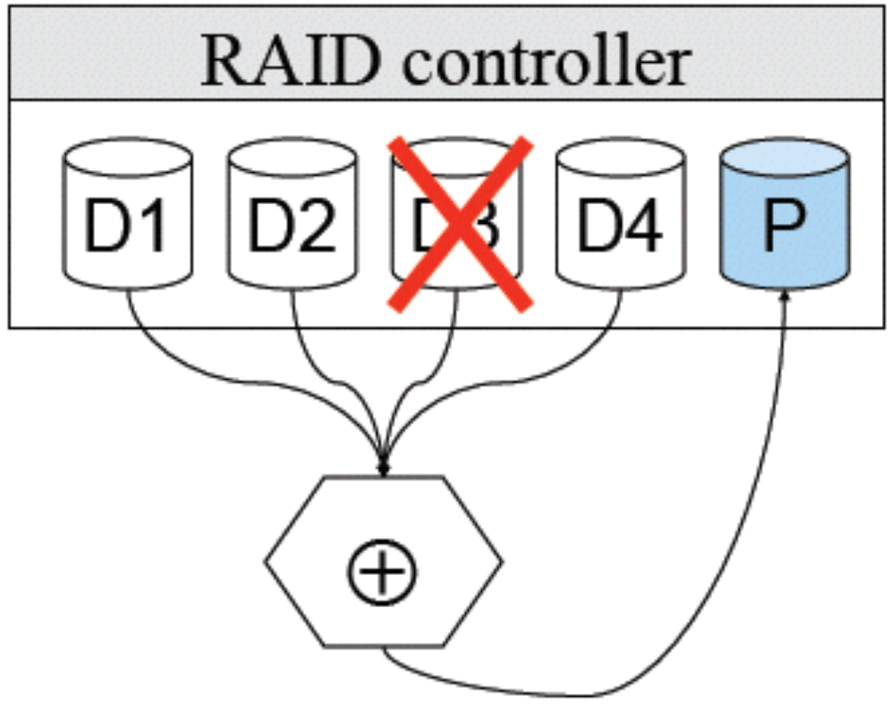
98, 183, 37, 122, 14, 124, 65, 67
(65, 67, 98, 122, 124, 183, 14, 37)

Storage System

- Network connected box with many disks
- Entry level box has 12 disks
  - Mean time to failure?
  - How to improve reliability?
  - What if there are 1000 disks?

RAID (Redundant Array of Independent Disks)

- Main idea
  - Store the error correcting codes on other disks
  - General error correcting codes are too powerful
  - Use XORs or single parity
  - Upon any failure, one can recover the entire block from the spare disk (or any disk) using XORs
- Pros
  - Reliability
  - High bandwidth
- Cons
  - The controller is complex



$$P = D1 \oplus D2 \oplus D3 \oplus D4$$

$$D3 = D1 \oplus D2 \oplus P \oplus D4$$

Synopsis of RAID Levels

RAID Level 0: Non redundant

RAID Level 1:
Mirroring

RAID Level 2:
Byte-interleaved, ECC

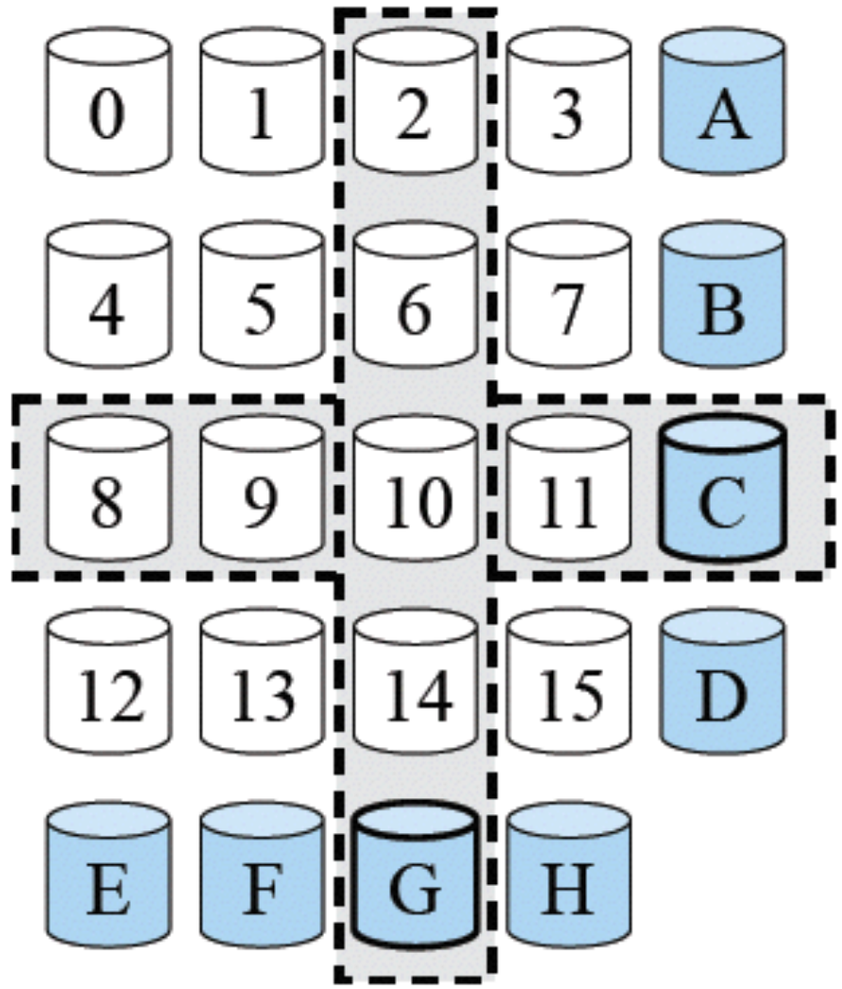
RAID Level 3:
Byte-interleaved, parity

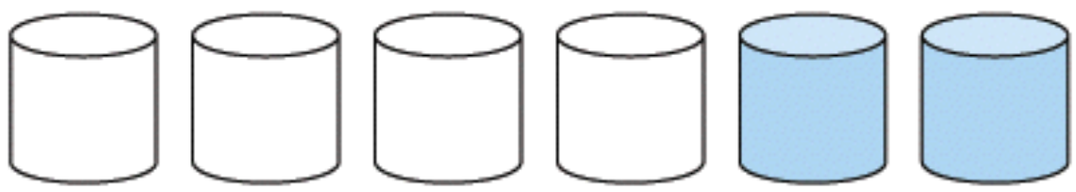
RAID Level 4:
Block-interleaved, parity

RAID Level 5:
Block-interleaved, distributed parity

RAID Level 6 and Beyond

- Goals
  - Less computation and fewer updates per random writes
  - Small amount of extra disk space
- Extended Hamming code
- Specialized Eraser Codes
  - IBM Even-Odd, NetApp RAID-DP, …
- Beyond RAID-6
  - Reed-Solomon codes, using MOD 4 equations
  - Can be generalized to deal with k (>2) disk failures





RAID level		Disk failures tolerated, check space overhead for 8 data disks	Pros	Cons	Company products
0	Nonredundant striped	0 failures, 0 check disks	No space overhead	No protection	Widely used
1	Mirrored	1 failure, 8 check disks	No parity calculation; fast recovery; small writes faster than higher RAIDs; fast reads	Highest check storage overhead	EMC, HP (Tandem), IBM
2	Memory-style ECC	1 failure, 4 check disks	Doesn't rely on failed disk to self-diagnose	~ Log 2 check storage overhead	Not used
3	Bit-interleaved parity	1 failure, 1 check disk	Low check overhead; high bandwidth for large reads or writes	No support for small, random reads or writes	Storage Concepts
4	Block-interleaved parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads	Parity disk is small write bottleneck	Network Appliance
5	Block-interleaved distributed parity	1 failure, 1 check disk	Low check overhead; more bandwidth for small reads and writes	Small writes → 4 disk accesses	Widely used
6	Row-diagonal parity, EVEN-ODD	2 failures, 2 check disks	Protects against 2 disk failures	Small writes → 6 disk accesses; 2× check overhead	Network Appliance

Figure D.4 RAID levels, their fault tolerance, and their overhead in redundant disks. The paper that introduced the term *RAID* [Patterson, Gibson, and Katz 1987] used a numerical classification that has become popular. In fact, the nonredundant disk array is often called *RAID 0*, indicating that the data are striped across several disks but without redundancy. Note that mirroring (RAID 1) in this instance can survive up to eight disk failures provided only one disk of each mirrored pair fails; worst case is both disks in a mirrored pair fail. In 2011, there may be no commercial implementations of RAID 2; the rest are found in a wide range of products. RAID 0 + 1, 1 + 0, 01, 10, and 6 are discussed in the text.

Figure from: Computer Architecture A Quantitative Approach

An Alternative to RAID

- Google File System
  - Open source version: Hadoop file system
- Distributed file system built on top of Linux FS
  - Special purpose for data-intensive computing (MapReduce)
  - Not intended to replace Linux FS for ordinary jobs
- Run on commodity components (clusters)
  - Each node in cluster has storage and computation resources
  - High aggregate I/O bandwidth
- Large blocks (64MB)
- Typically 3x replication for blocks
- MapReduce jobs

Dealing with Disk Failures

- What failures
  - Power failures
  - Disk failures
  - Human failures
- What mechanisms required
  - NVRAM for power failures
  - Hot swappable capability
  - Monitoring hardware
- RAID reconstruction
  - Reconstruction during operation
  - What happens if a reconstruction fail?
  - What happens if the OS crashes during a reconstruction

New Generation: FLASH

Retail prices March 2015(Komplett):

960 GB SSD	NOK 5795
240GB SSD	NOK 1299
2TB disk	NOK 789
64GB SD:	NOK 349

- Flash chip density increases on the Moore's law curve
  - 1995 16 Mb NAND flash chips
  - 2005 16 Gb NAND flash chips
  - 2009 64 Gb NAND flash chips
  - Doubled each year since 1995
- Market driven by Phones, Cameras, iPod,…
  Low entry-cost,
  ~$30/chip → ~$3/chip

Flash Memory

- NOR
  - Byte addressable
  - Often used for BIOS
  - Much higher price than for NAND
- NAND
  - Dominant for consumer and enterprise devices
  - Single Level Cell (SLC) vs. Multi Level Cell (MLC):
    - SLC is more robust but expensive
    - MLC offers higher density and lower price

NAND Memory Organization

- Organized into a set of erase blocks (EB)
- Each erase block has a set of pages
- Example configuration for a 512 MB NAND device:
  - 4096 EB's, 64 pages per EB, 2112 bytes per page (2KB user data + 64 bytes metadata)
- Read:
  - Random access on any page, multiple times
  - 25-60µs
- Write
  - Data must be written sequentially to pages in an erase block
  - Entire page should be written for best reliability
  - 250-900 µs
- Erase:
  - Entire erase block must be erased before re-writing
  - Up to 3.5ms

Flash Translation Layer

- Emulate a hard disk by exposing an array of blocks
- Logic block mapping
  - Map from logical to physical blocks
  - Cannot do random writes
  - Granularity: block vs. page (read-modify-write block vs. large RAM for storing map table
  - Hybrid approach often used: maintain a small set of log blocks for page level mapping
- Garbage collection
  - Maintain an allocation pool of clean blocks (blocks must be erased before reuse)
  - Recycle invalidated pages
  - Merge blocks if page level mapping
- Wear leveling
  - Most writes are to a few pages (metadata)
  - Even out writes over blocks

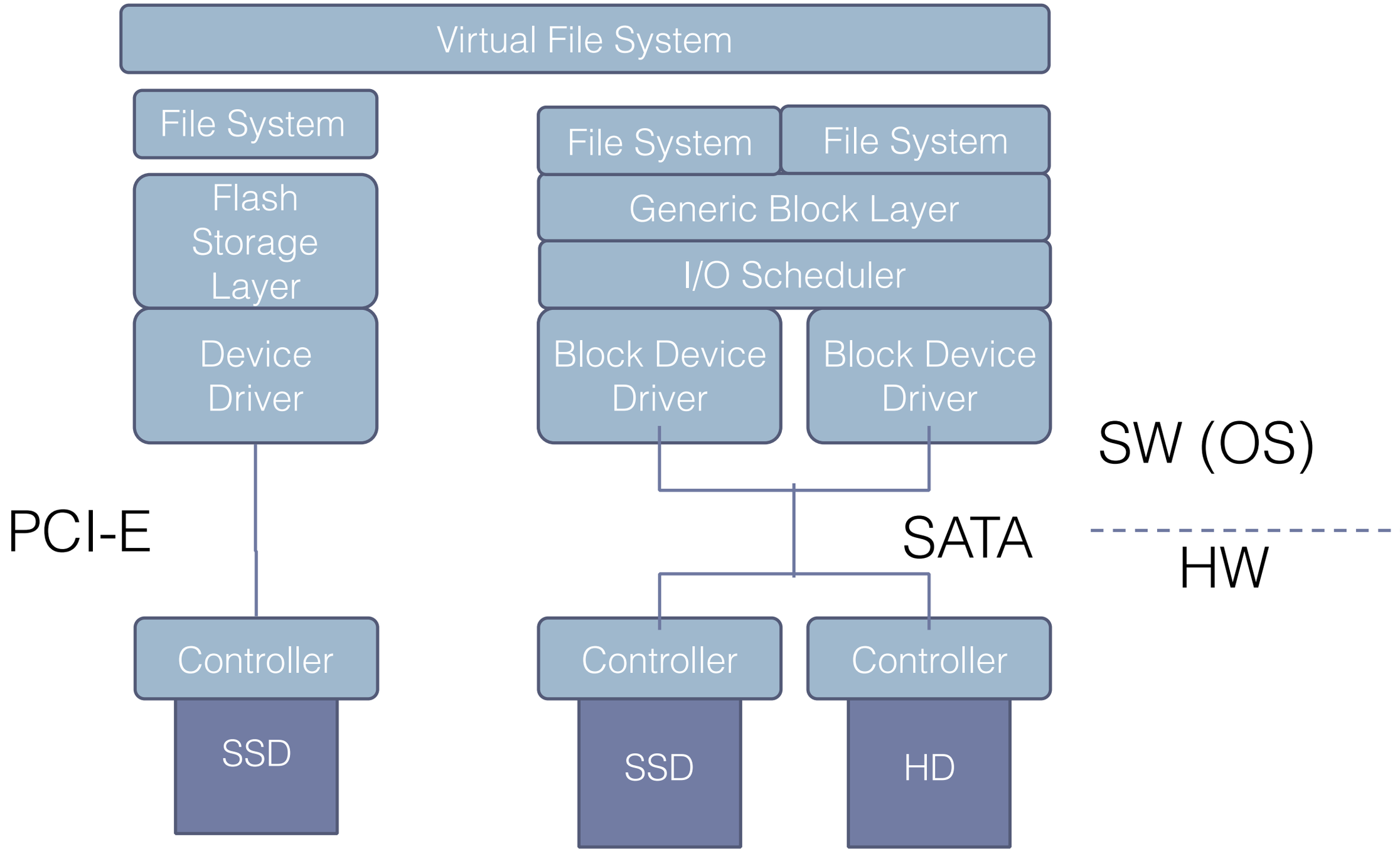
What's Wrong With FLASH?

- Expensive: $/GB
  - 2x less than cheap DRAM
  - 10-20x more than disk today
  - Limited lifetime
    - ~100k to 1M writes / page (single cell)
    - ~15k to 1M writes / page (single cell)
    - requires "wear leveling"
      but, if you have 1,000M pages,
      then 15,000 years to "use" the pages.
- Current performance limitations
  - Slow to write can only write 0's, so erase (set all 1) then write
  - Large (e.g. 128K) segments to erase

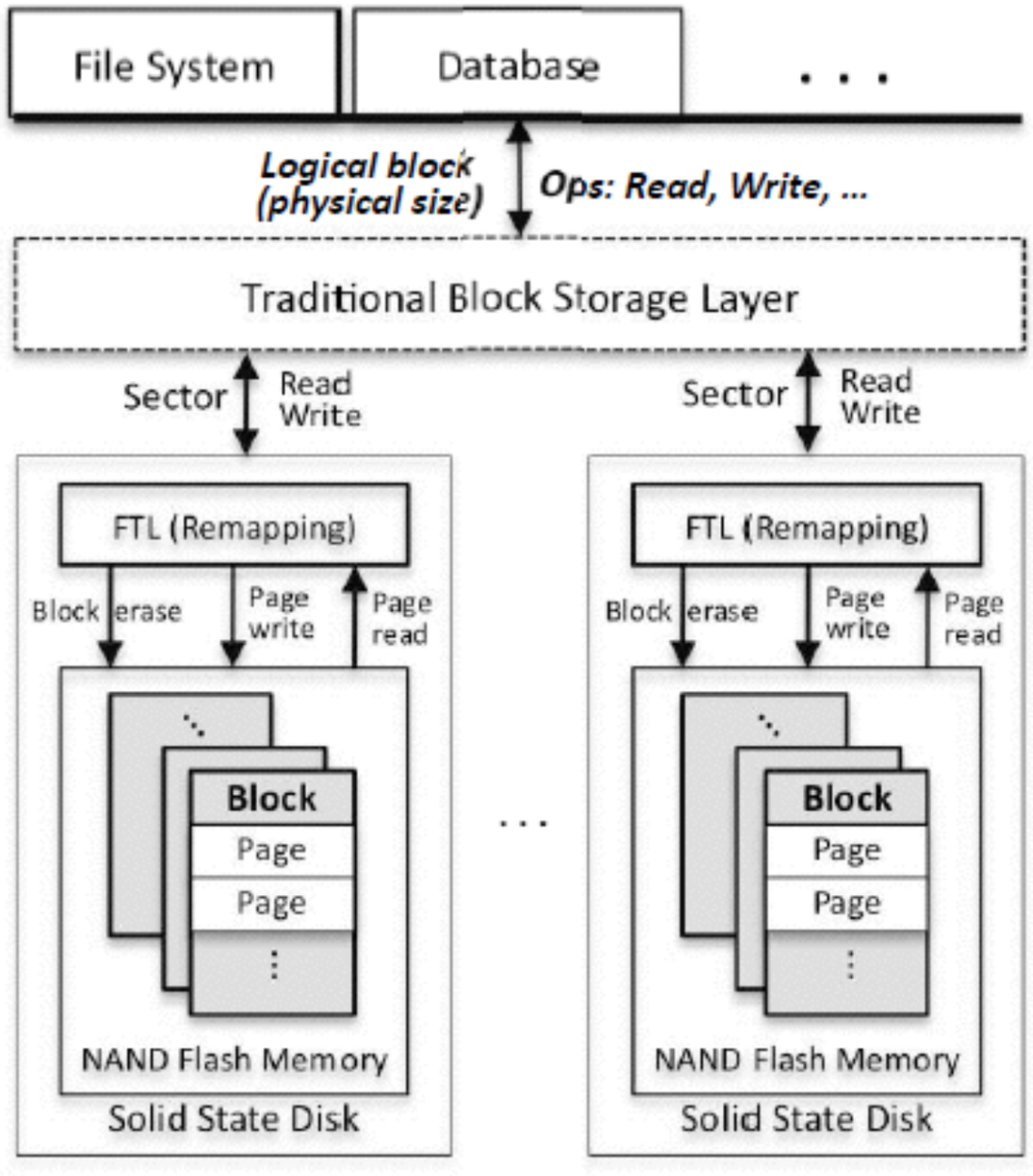
Current Development

- Flash Translation Layer (FTL)
  - Remapping
  - Wear-leveling
  - Write faster
- Form factors
  - SSD
  - USB, SD, Stick,…
  - PCI cards
- Performance
  - Fusion-IO cards achieves 200K IOPS

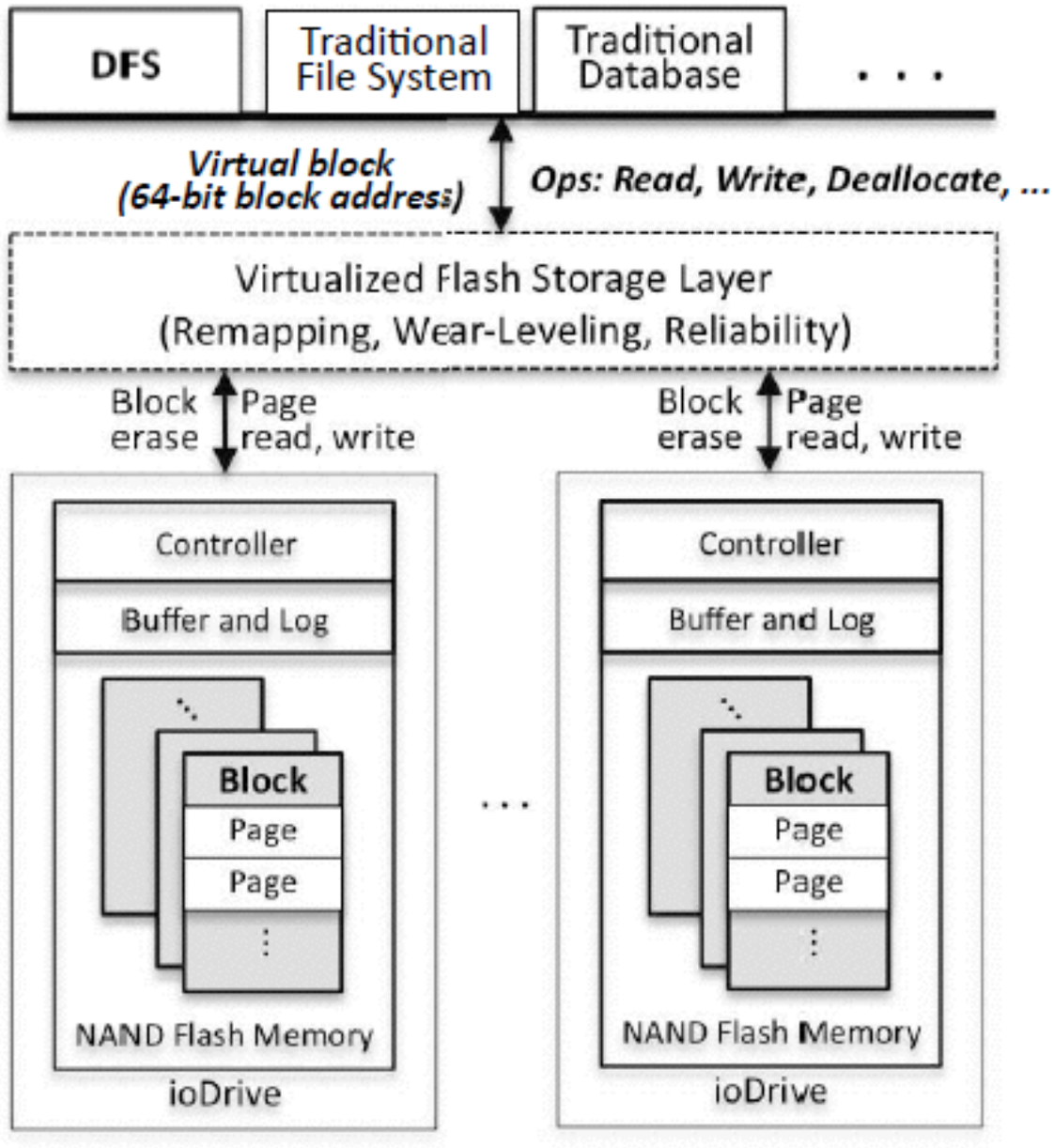
Hardware/Software Architecture



DFS: A File System for Flash Storage



(a) Traditional layers of abstractions



(b) Our layers of abstractions

Figure 1: Flash Storage Abstractions

Issues

- Where to put a flash drive in the storage hierarchy?
- Which new algorithms need to be developed?
- Should the OS treat flash drive as a hard drive?

Good Paper

Understanding Intrinsic Characteristics and System Implications of Flash Memory based Solid State Drives

Feng Chen[1], David A. Koufaty[2], and Xiaodong Zhang[1]

[1]Dept. of Computer Science & Engineering
The Ohio State University
Columbus, OH 43210
{fchen, zhang}@cse.ohio-state.edu

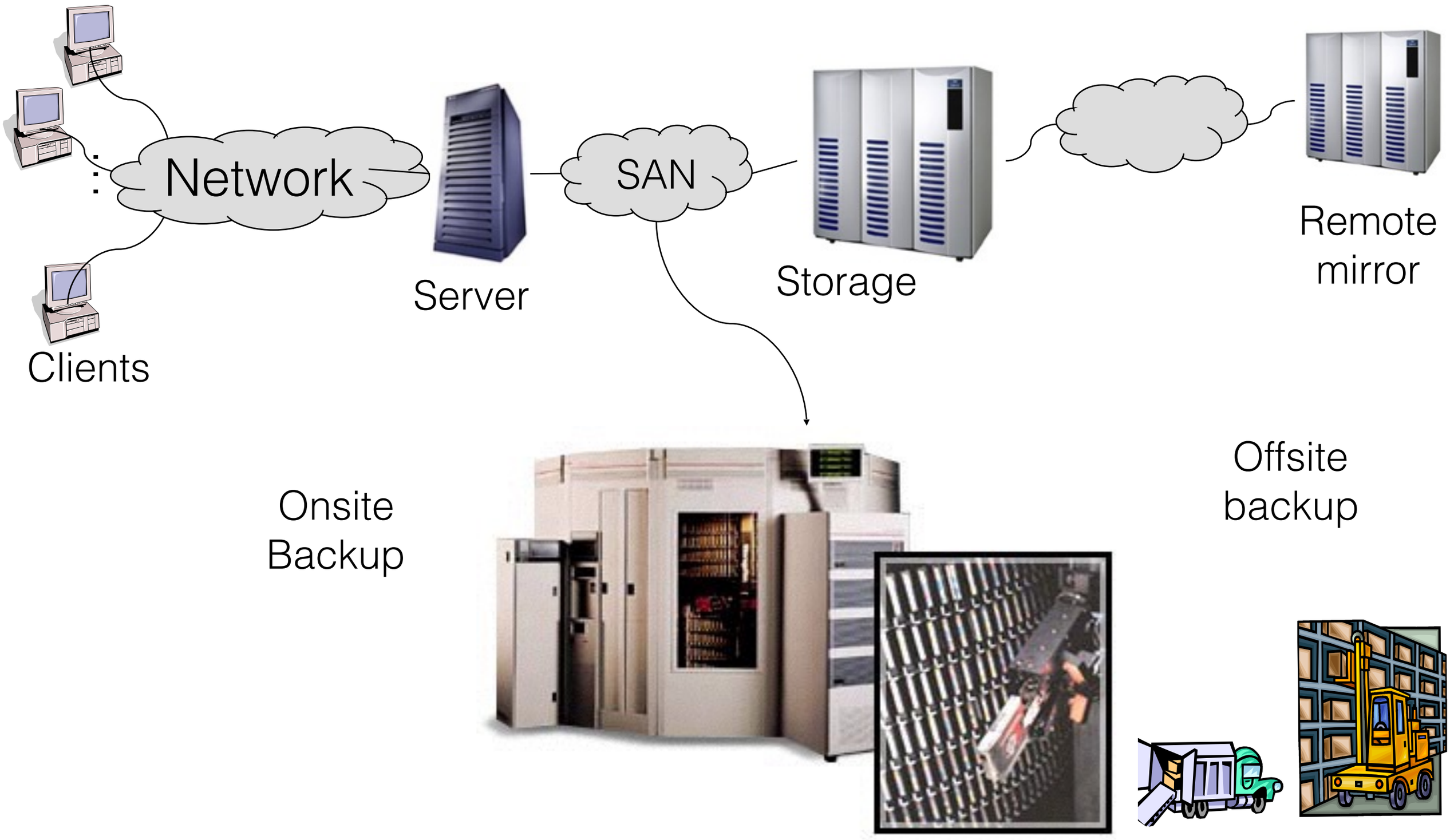
[2]System Technology Lab
Intel Corporation
Hillsboro, OR 97124
david.a.koufaty@intel.com

SIGMETRICS/Performance'09, June 15–19, 2009, Seattle, WA, USA.

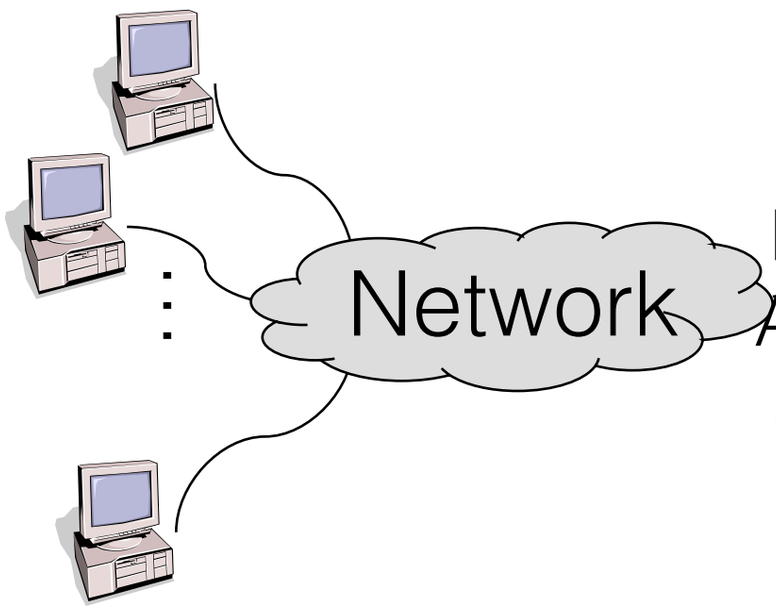
DRAM Storage Systems

- Currently increasing interest in DRAM only storage systems
  - All data structures in memory
  - Data is backed on disk in background
- In-memory databases
  - Database entirely in memory
  - Often used for analytics
  - Business critical data stored on ordinary databases
- RAM Cloud
  - A big compute cluster
  - Data structures replicated over many computers
  - Data also written to disk
  - http://www.sigops.org/sosp/sosp11/current/2011-Cascais/03-ongaro-online.pdf

Traditional Data Center Storage Hierarchy



Evolved Data Center Storage Hierarchy



Clients

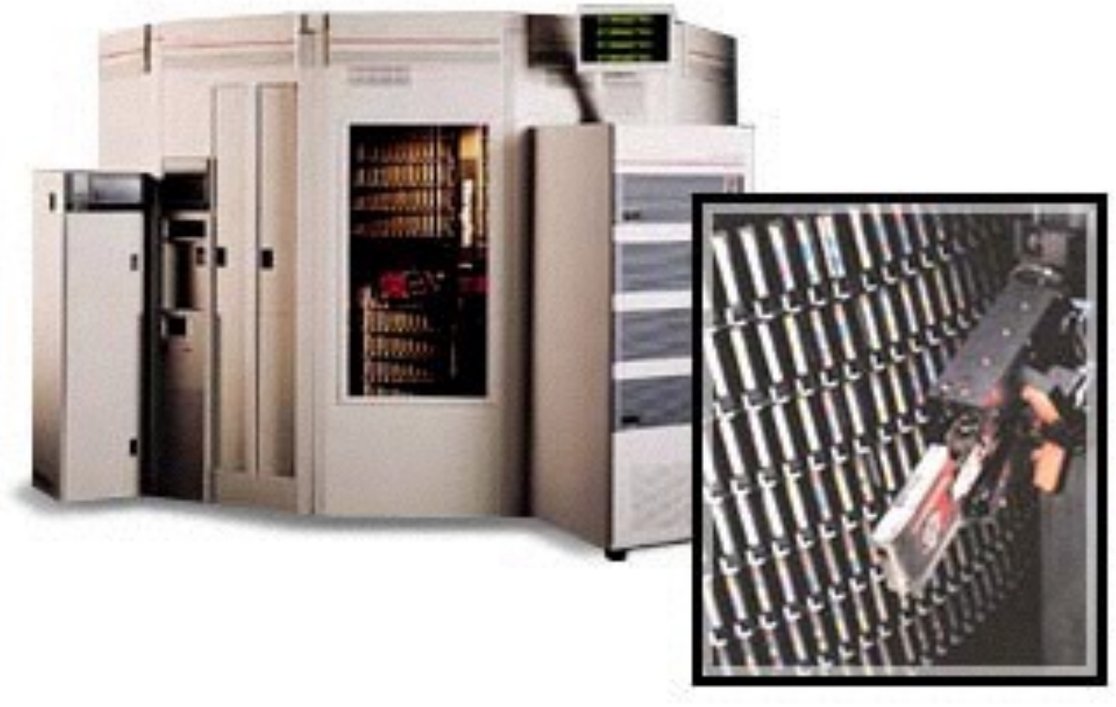
Network

Network Attached Storage (NAS)

w/ snapshots to protect data

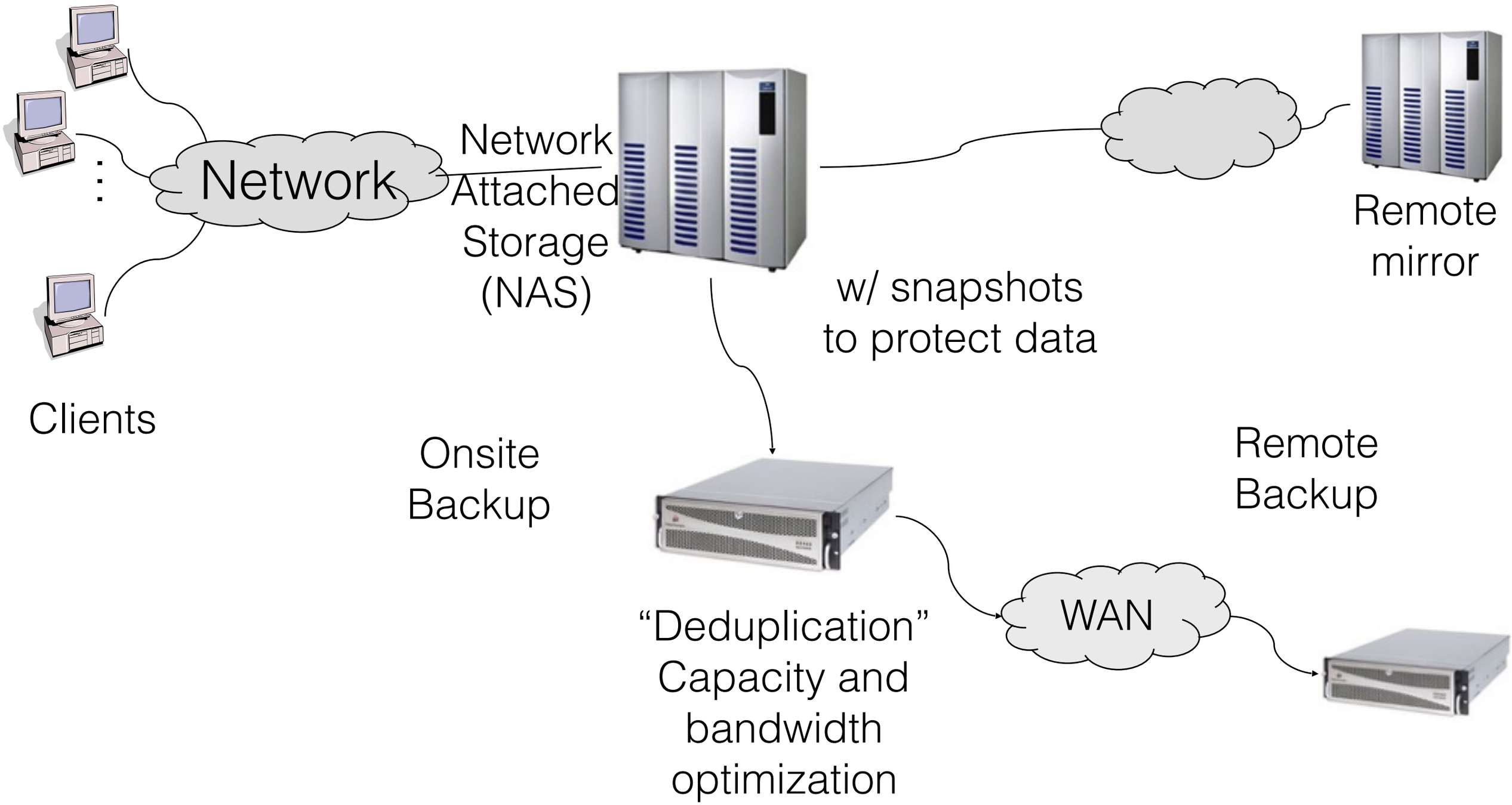
Remote mirror

Onsite Backup

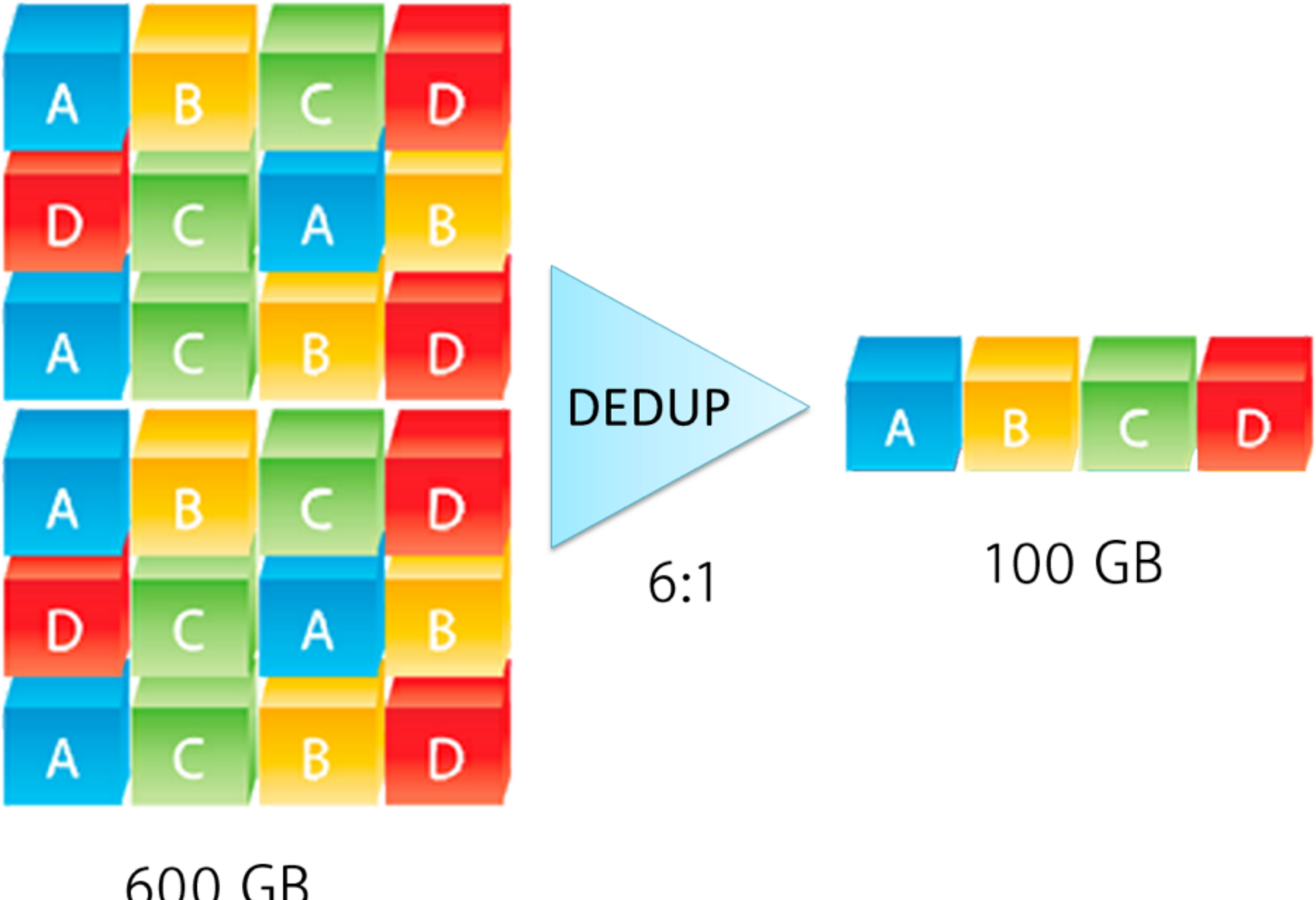


Offsite backup

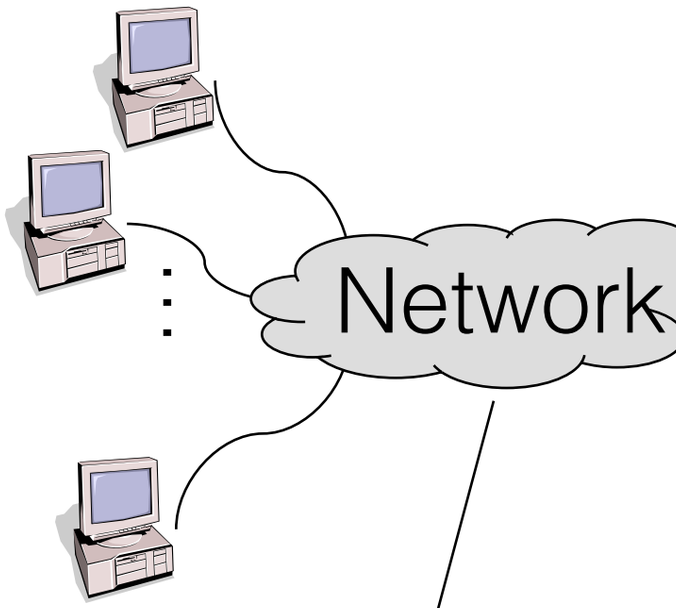
Modern Data Center Storage Hierarchy



Detour: Deduplication

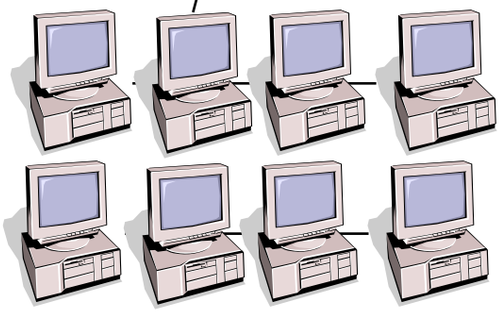


Very Large Dataset Storage Hierarchy



Clients

Network



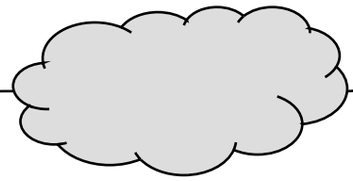
Distributed Storage
w. internal replication

Network
Attached
Storage
(NAS)

Onsite
Backup

"Deduplication"
Capacity and
bandwidth
optimization

w/ snapshots
to protect data



WAN

Remote
mirror

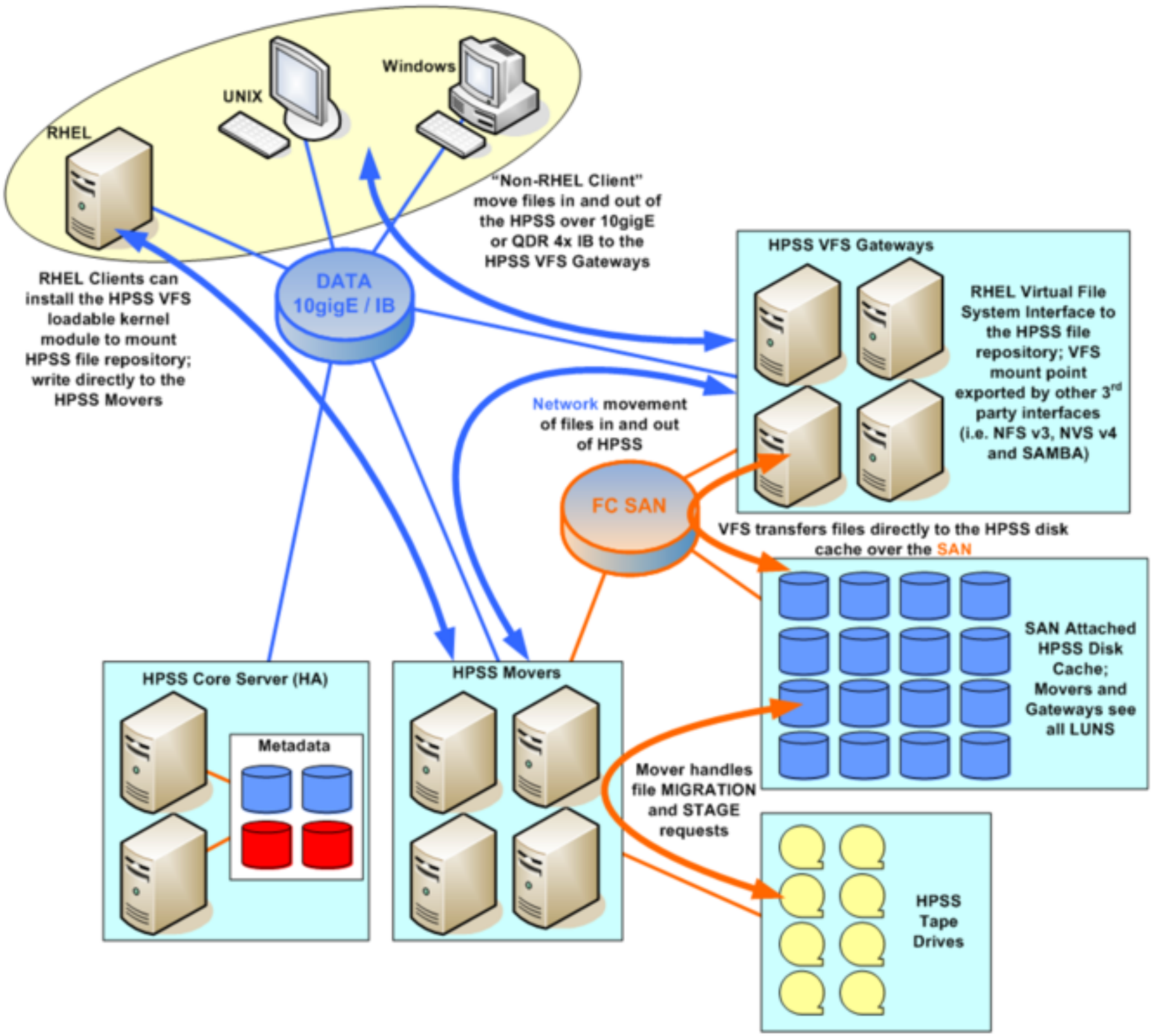
Remote
Backup

ECFC
ECMWF File System

- Provides a logical view of a seemingly very large file system (PB)

- Unix-like commands for accessing files
  - epwd, ecd, ecp, emkdir, els

IBM HPSS
High Performance Storage System



10.000 Years Storage Systesm

- How to build a storage system with a mean time to failure = 10.000 years?

Towards practical, high-capacity, low-maintenance information storage in synthesized DNA

Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. LeProust, Botond Sipos & Ewan Birney

- Uses lots of basic storage system principles
- I/O performance not great

Summary

- Disk is complex
- Disk real density is on Moore's law curve
- Need large disk blocks to achieve good throughput
- OS needs to perform disk scheduling
- RAID improves reliability and high throughput at a cost
- Careful designs to deal with disk failures
- Flash memory has emerged at low and high ends
- DRAM only storage systems are emerging
- Storage hierarchy is complex