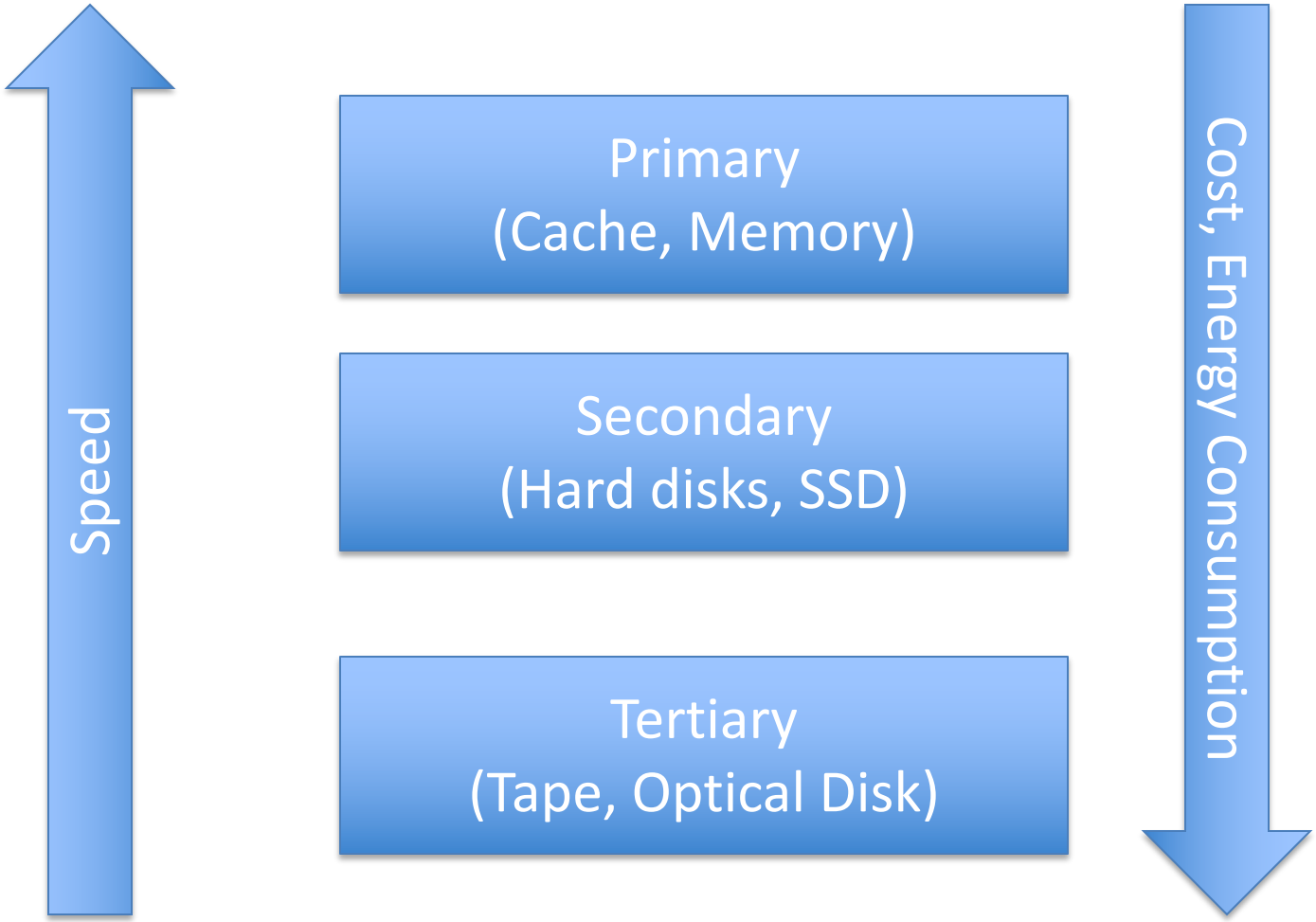
Storage Systems

-or-

Where do the bits live?

Computer Storage Hierarchy



Characterizing Storage

- Cost
- Performance
- Availability
- Fixity
- Permanence

Storage Operations: The CRUD model

- We should think about storage in terms of storage operations
- CRUD:
  - (C)reate
  - (R)Read
  - (U)pdate
  - (D)elete

Storage is Not Forever

- Transient vs. Permanent Errors
- Media Failures
  - One part of the storage media stops working
    - Cannot read/write to physical location in media
- Device Failures
  - Failure of entire device, data may not be recoverable
- Detecting Failure
  - Periodic scanning of media
  - On-device monitoring
  - Inability to access device

Major Performance Measures

- Capacity (bytes)
  - How much data can you hold
- Cost ($$$)
  - Price per byte of storage
- Bandwidth (bytes/sec)
  - Number of bytes/second that can be transferred
  - Read and write bandwidth may be different
- Latency (secs)
  - Time between initiating a request and an action.
  - In the case of storage, to deliver 1st Byte

Storage capacity

- 1 Bit = Binary Digit
- 8 Bits = 1 Byte
- 1000 Bytes = 1 Kilobyte (10^3)
- 1000 Kilobytes = 1 Megabyte (10^6)
- 1000 Megabytes = 1 Gigabyte (10^9)
- 1000 Gigabytes = 1 Terabyte (10^{12})
- 1000 Terabytes = 1 Petabyte (10^{15})
- 1000 Petabytes = 1 Exabyte (10^{18})
- 1000 Exabytes = 1 Zettabyte
- 1000 Zettabytes = 1 Yottabyte
- 1000 Yottabytes = 1 Brontobyte
- 1000 Brontobytes = 1 GeopbyteBytes

Confusion….

- **1000 bytes**
  - In the International System of Units (SI) the prefix *kilo-* means 1000 (10^3); therefore one kilobyte is 1000 bytes in this system. The unit symbol is **kB**.
  - This is the definition recommended by the International Electrotechnical Commission (IEC).This definition, and related definitions of prefixes mega- = 1000000, giga- = 1000000000, etc., are used for data transfer rates in computer networks, internal bus, hard drive and flash media transfer speeds, and for the capacities of most storage media. It is also consistent with the other uses of the SI prefixes in computing, such as CPU clock speeds or measures of performance.
- **1024 bytes**
  - In some fields of information technology, the kilobyte instead refers to 1024 (2^{10}) bytes. This definition, and related definitions of mega = 1048576 (=1024^2), etc., are almost invariably used for random access memory capacities, such as main memory and CPU cache sizes, due to the binary addressing of memory. These "binary meanings" of kilobyte, megabyte, etc., are also used by Windows and Linux operating systems when reporting disk capacities and file sizes.
  - The binary representation of 1024 bytes typically uses the symbol **KB** (uppercase *K*). The *B* is often omitted in informal use. For example, a processor with 65,536 bytes of cache might be said to have "64K" of cache.
- **kibibyte**
  - In December 1998, the IEC addressed such multiple usages and definitions by creating prefixes such as kibi, mebi, gibi, etc., to unambiguously denote powers of 1024.[10] Thus the kibibyte, symbol KiB, represents 2^{10} = 1024 bytes. These prefixes are now part of the International System of Quantities. The IEC further specified that the kilobyte should only be used to refer to 1000 bytes. However, the kilobyte is still commonly used to refer to 1024 bytes.

Time to Complete an Operation

- Time to complete an operation depends on both bandwidth and latency
  - CompletionTime = Latency + Size/Bandwidth
- The time will depend on technology, operation type, number of operations and access patterns

Access patterns

- Sequential
  - Data to be accessed are located next to each other on the device
- Random
  - Access data located randomly on storage device

Performance factors

- Small requests will be dominated by latency
  - Why?
- Big requests will be dominated by bandwidth
  - Why?

Factors impacting performance

- Performance is influenced by the way the storage device is used
- Ratio of reads, writes, updates and deletes
- Size of the data being manipulated
- Request rates and concurrency
- Access patterns
  - Sequential vs random

Why does this Matter?

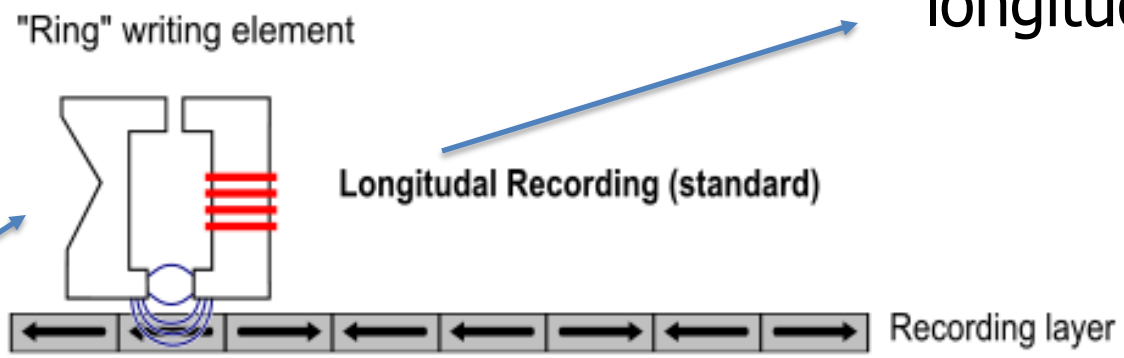
- Data management strategy is will often be driven by performance of storage system
  - Optimize for reads, writes
  - Big chunks of data, small chunks of data
  - Random access, sequential access
- Blend approaches to optimize performance
  - Store on device, on cloud, on local drive, on network drive…
  - Structure of storage systems

Storage technologies

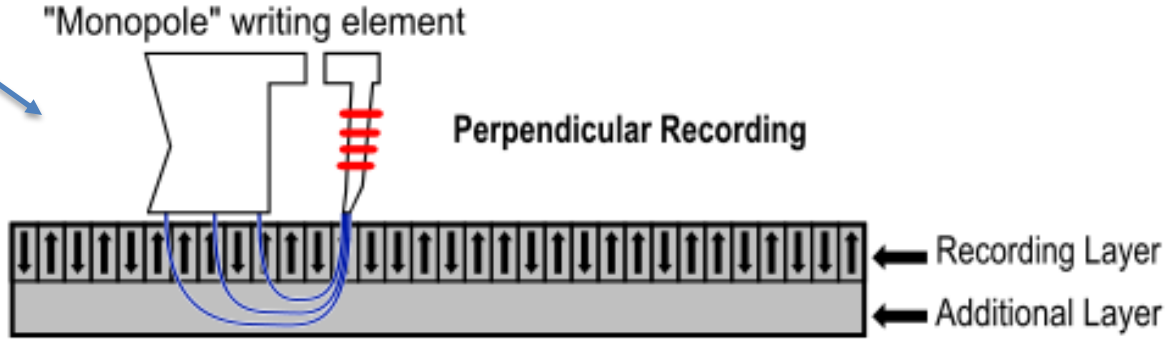
- Tapes
- Hard disk
- Solid state disk

Magnetic recording

longitudinal



Read/write head



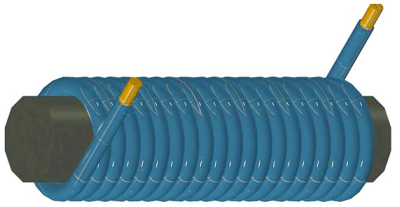
Change of direction of current => change of direction of field

Read and write heads

- Read/write head is composite:
  - separate components for read and write
- Read head
  - Conductive material changes its resistance in presence of magnetic field
  - A sensor is used to detect the changes in resistance

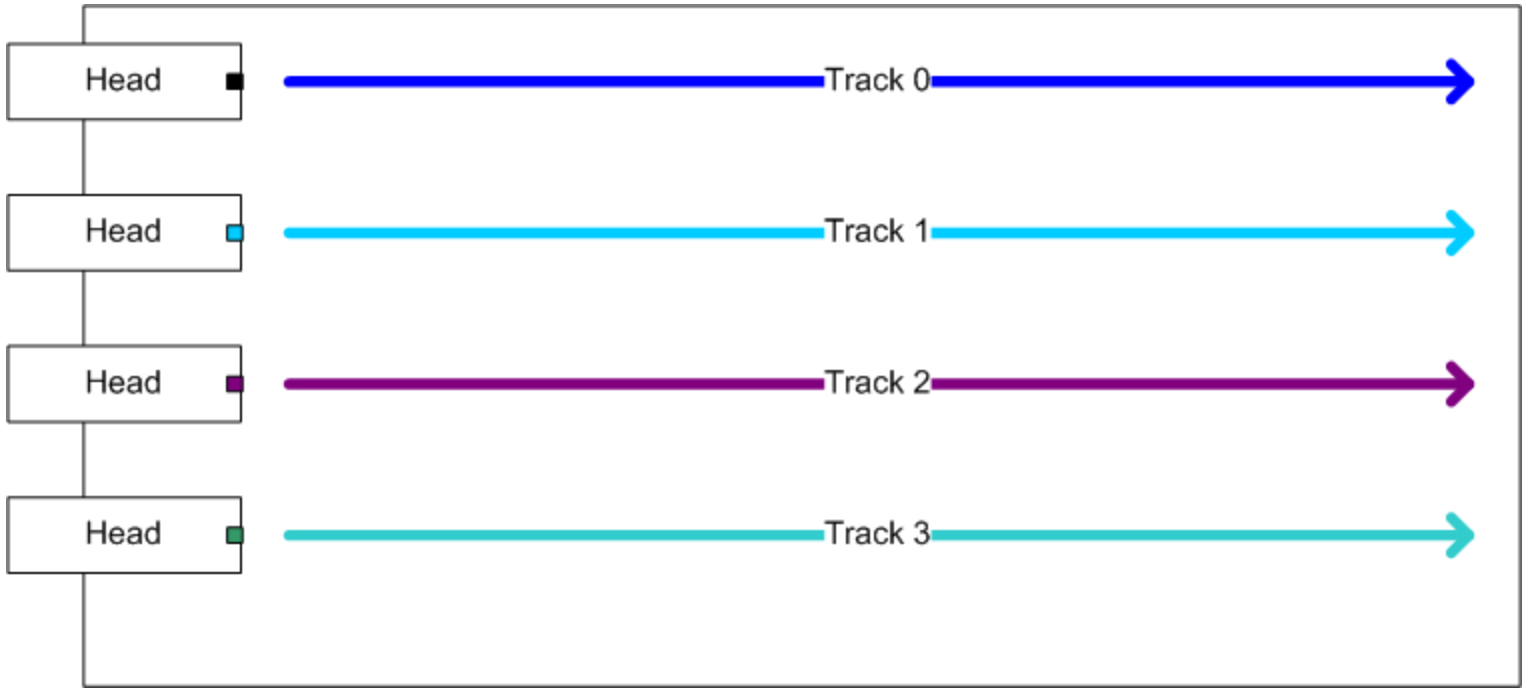
Write head

- Write
  - Transform current => magnetic field
  - Through an electromagnet (a magnet whose magnetic fields are produced by an electric current)



Linear tape

- Data recorded on parallel tracks that span the length of the tape



Tapes

- Current technology is LTO
  - Linear Tape Open
- Characteristics
  - Capacity up 6.5 TB per tape
  - Drive cost ~ $2500
  - Tape cost ~ $45 for 2.5TB tape
- Tape access time (~ minute)
  - Time to mount the tape
  - Time to wind the tape to correct position
- Data rates ~ 250Mbyte/sec

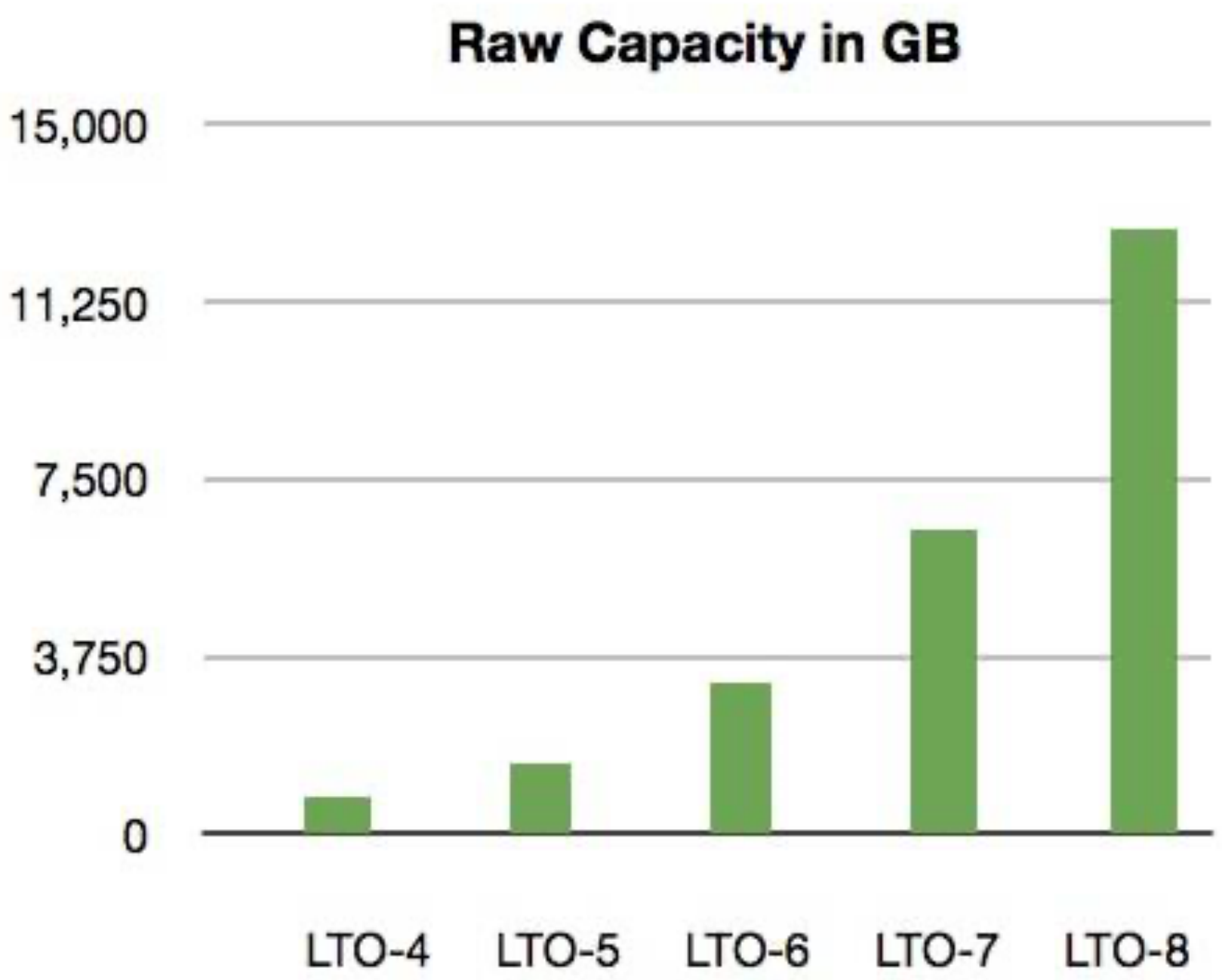
Performance Characteristics

- High latency/low cost makes tape most appropriate for “archival” storage
  - Low frequency of reads
  - Very large data objects
- Random access will be slow due to latency
  - Sequential reads will be fast

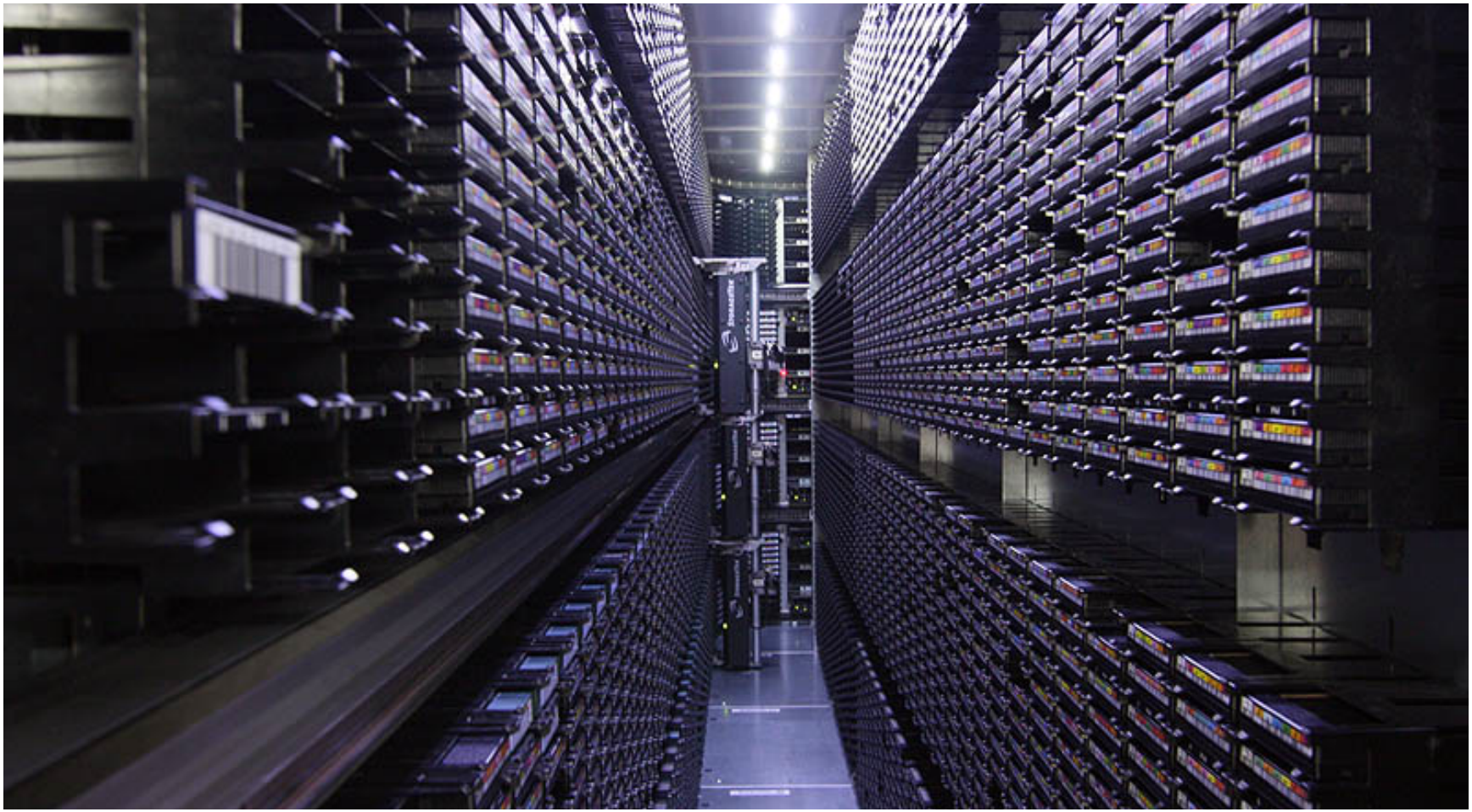
Linear Tape File System

- Two partitions on tape.
  - First contains metadata and directories. Tape reader can find and load this very quickly
  - Second contains blocks for data
- Directory structure coded in XML
  - Self describing file format…

Tape Cartridge


Raw Capacity in GB
15,000
11,250
7,500
3,750
0
LTO-4
LTO-5
LTO-6
LTO-7
LTO-8

A Tape Library



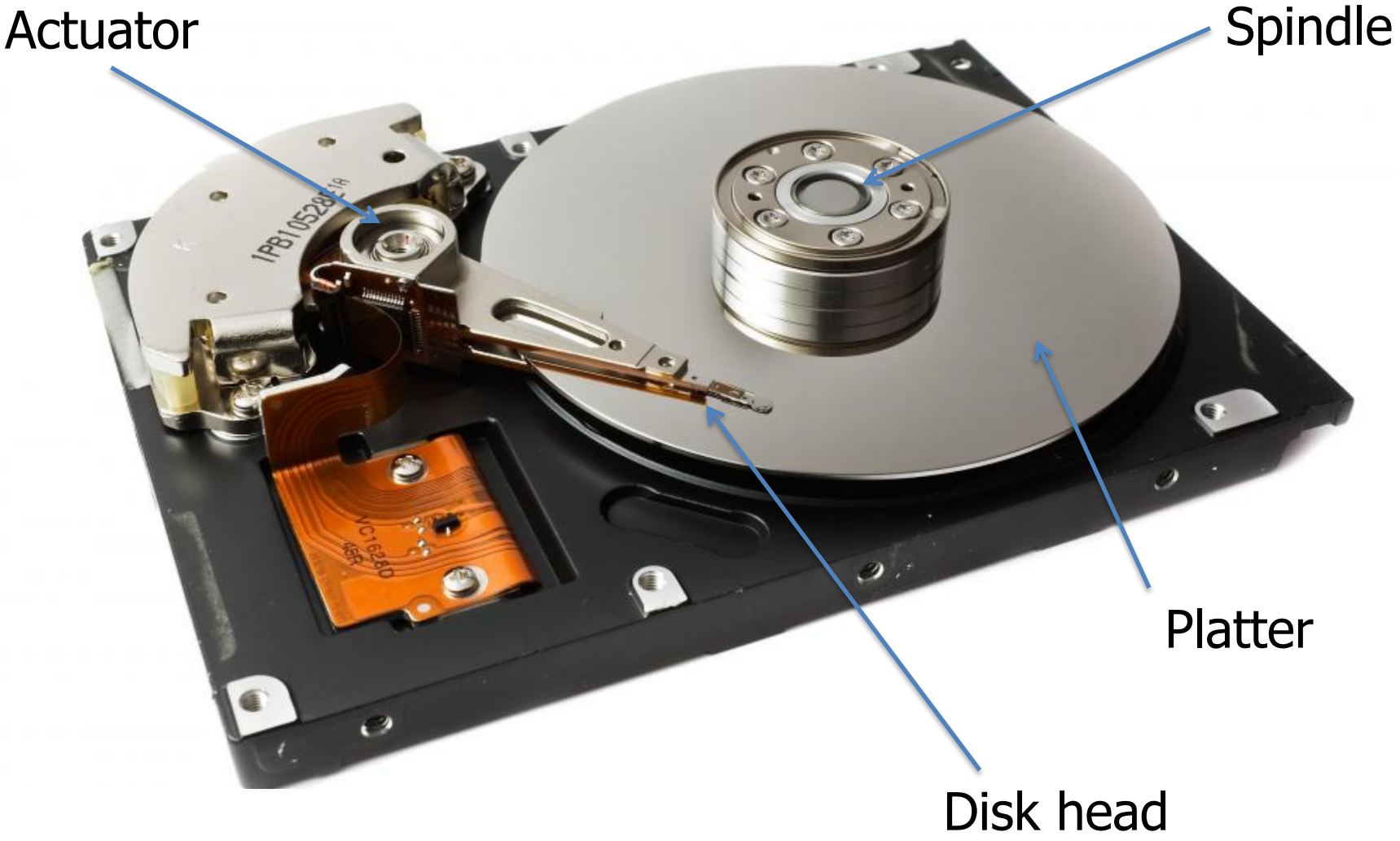
Hard Disk Drives

- Perhaps the most pervasive form of storage
- Basic Idea:
  - One or more spinning magnetic platters
  - Disk arm positions over the radial position where data is stored
  - Data is read/written by a read/write head as platter spins


Western Digital
www.westerndigital.com
3 TB
SATA / 64MB Cache
WD30EZRS
WD Caviar Green

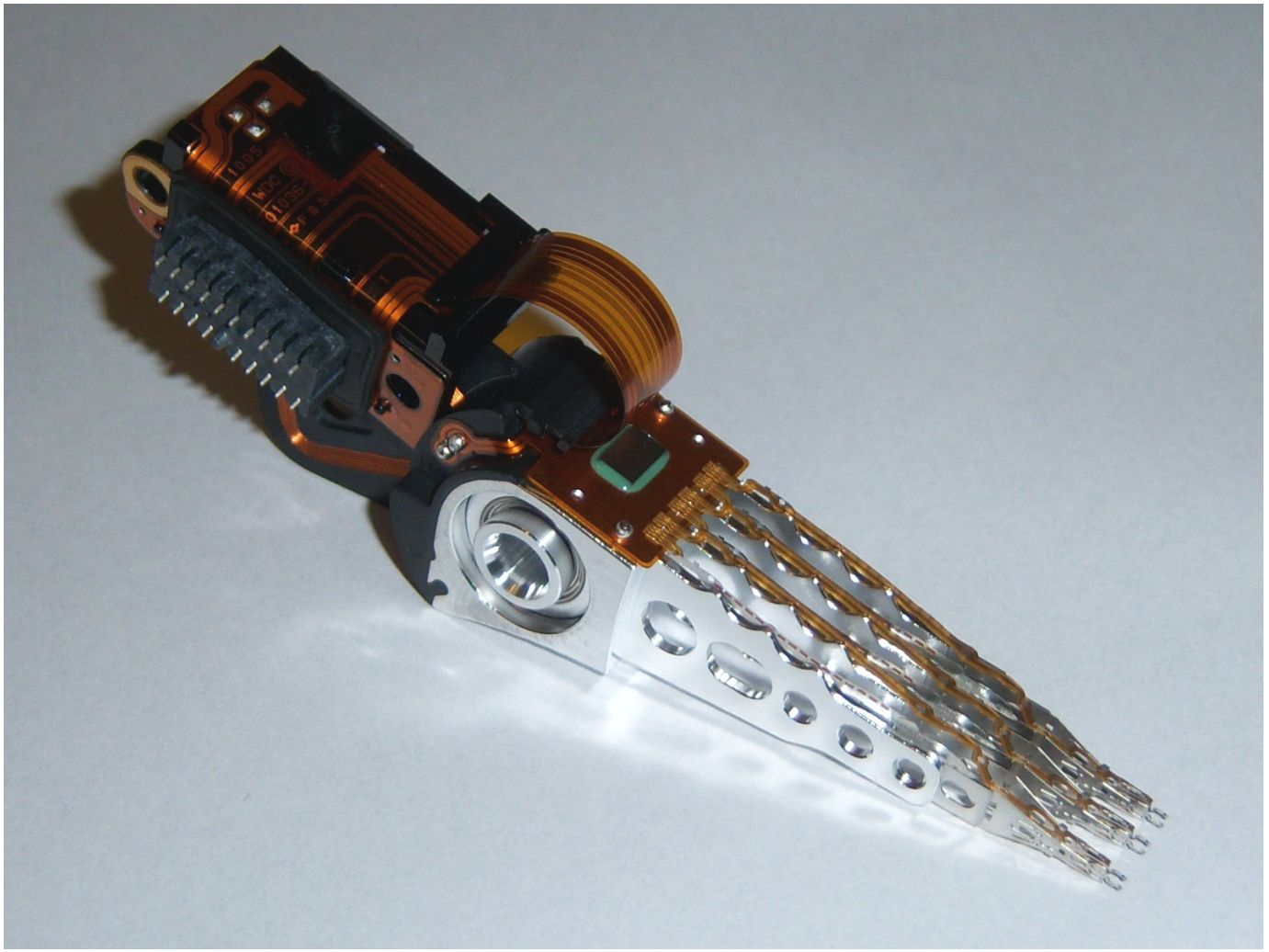
2GB of 1980s Storage ($250,000)

Internal of hard disk



Disk arm and platter

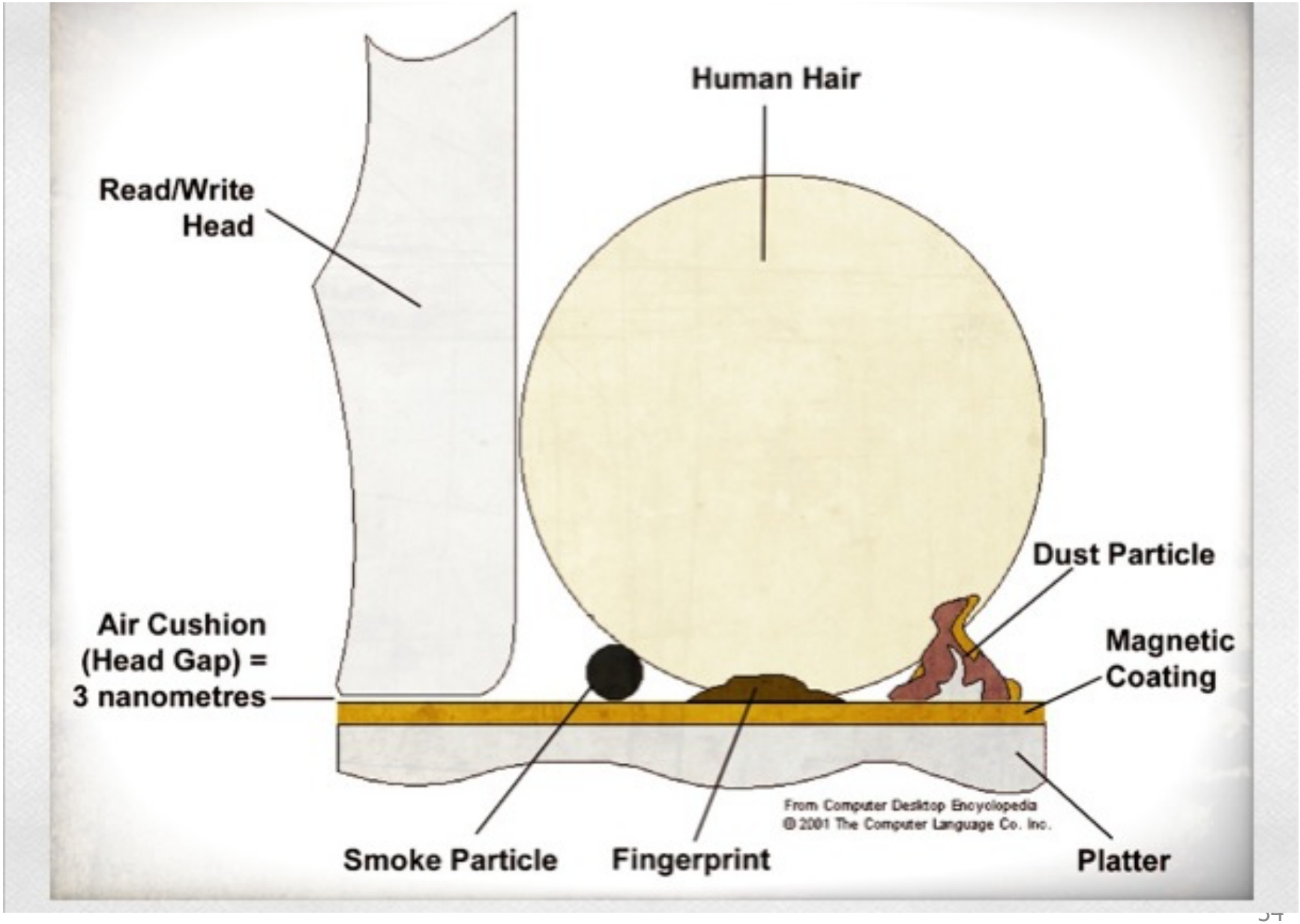
Disk head close-ups



Disk head close-ups

Disk head movement

- Hard disk head movement while copying files between two folders (e.g., partition c to d)
  - https://www.youtube.com/watch?v=BlB49F6ExkQ


Human Hair
Read/Write Head
Dust Particle
Air Cushion (Head Gap) = 3 nanometres
Magnetic Coating
From Computer Desktop Encyclopedia
© 2001 The Computer Language Co. Inc.
Smoke Particle
Fingerprint
Platter

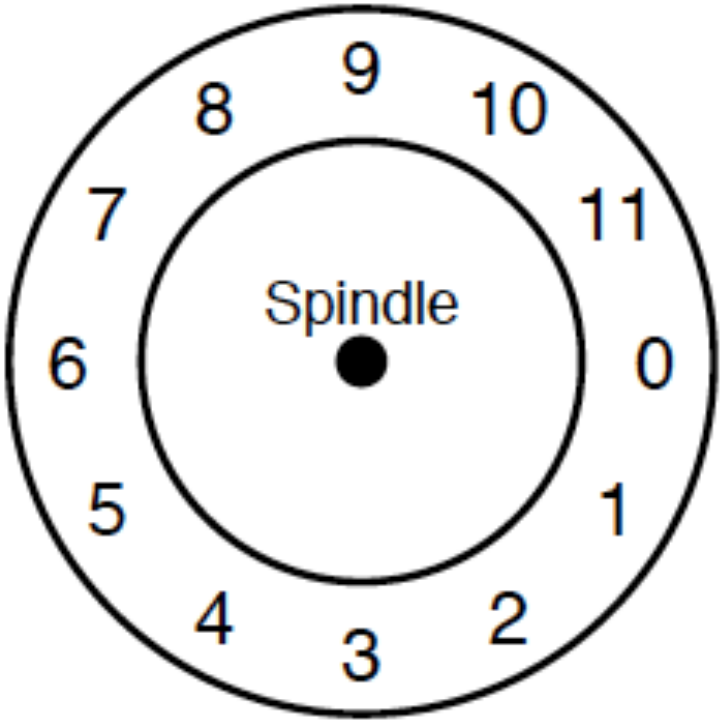
Physical Characteristics

- 5.25 in (common in servers),
- 3.5 in (common in laptops)
- Rotational speed
  - 5,400 RPM
  - 7200 RPM
  - 4800 RPM
  - 10000 RPM (6 msec rotation)
- Between 5-7 platters
- Current capacity up to 15 TB

Disk Organization

- Disk split into N fixed size sectors
  - Typical sector size is 512 bytes
  - Sectors can be numbered from 0 to N-1
  - Entire sector is written “atomically”
    - All or nothing

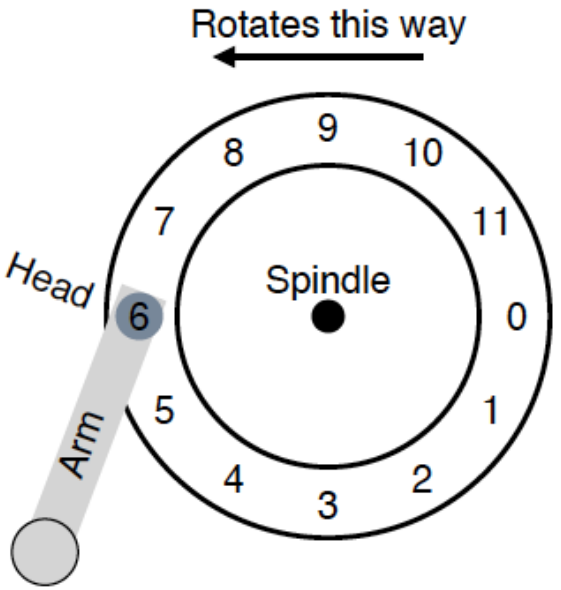
A Simple Disk Drive



Arpaci-Dusseau

Rotational latency

- Waiting for the right sector to rotate under the head
  - On average: ½ of time for a full rotation
  - Worst case?
  - Best case?



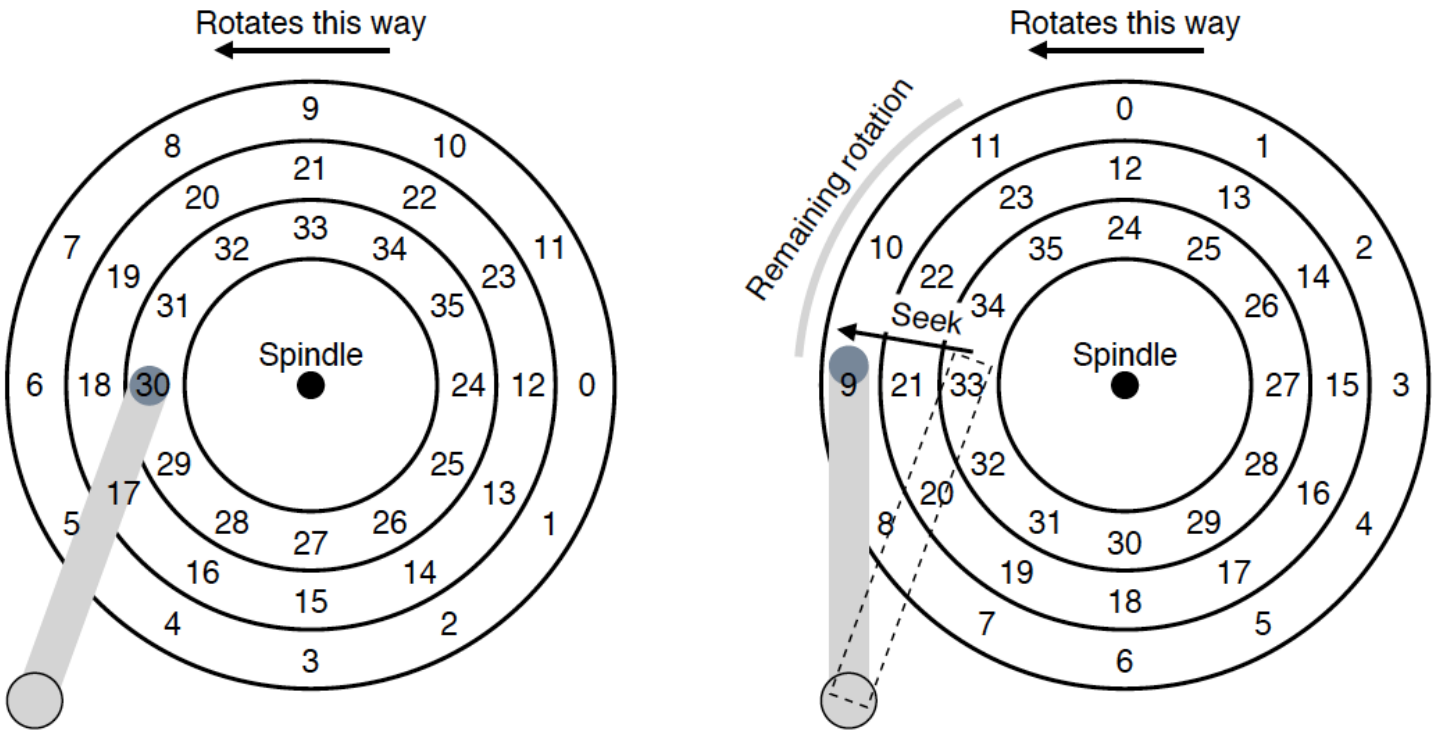
What is the time for 1 Rotation?

- Assume 10,000 RPM (rotations per minute)

$$\frac{Time\ (ms)}{1\ rotation} =$$

$$\frac{1\ \text{min}}{10000\ rot} * \frac{60\ \text{sec}}{1\ \text{min}} * \frac{1000\ msec}{1\text{sec}} = \frac{60000 msec}{1000\ rot} = \frac{6\ msec}{rotation}$$

Muliple Tracks/Platter



Average seek time is aprox. 1/3 max seek time

How long does it take to transfer one sector

- Assume 100 MB/sec transfer bandwidth

$$\frac{Time}{request}$$

$$= \frac{512\ KB}{1\ request} * \frac{1\ MB}{1024\ KB} * \frac{1\ second}{100\ MB}$$

$$* \frac{1000\ msec}{1 second} = \frac{5\ msec}{1\ request}$$

Performance for a single request

- $T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$
  - Time to get the disk head in place
  - Time to wait for the right sector to rotate under the head
  - Time to actually transfer the data

Data Access Times:

$$T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$$

- Typical values: **SEAGATE ST4000NC000**
  - Capacity 4 TB
  - Number Of Disks 4
  - Number Of Heads 8
  - Form Factor: 3.5”
  - Buffer Size: 64 MB
  - Bytes per Sector: 4096
  - Drive Transfer Rate: 600 MBps (external)
  - Internal Data Rate: 160 MBps
  - Seek Time: 12 ms (average)
  - Average Latency: 5.1 ms
  - Spindle Speed: 5900 rpm

Time to Read a 4K Block

- 4 msec average seek time, 125 MB/sec transfer rate, 15,000 RPM
- T_{seek} = 4 msec
- Time for one rotation is 1/15000 = 4 msec. so $T_{rotation}$ = 2 msec
- Time for data transfer is

$$\frac{Time}{4k} = \frac{1\ second}{125\ MB} * \frac{1000\ msec}{1\ sec} * \frac{1\ MB}{1024KB} * \frac{4\ KB}{Transfer} = .031\ msec = 31\ microsec.$$

So we have: 4ms + 2ms + .031 = 6.03 msec.

- Note that we are dominated by latency.
  - What would happen if the transfer rate was 10 times slower?
- If we were to read N blocks, it would take N times as long
- The effective I/O transfer rate would be 1/6 msec, or .66 MB/sec

Sequential Operations

- What would happen if we did N operations and they were all to the same track?
  - We wouldn't have to reposition the disk head
  - We wouldn't have to wait for the disk to spin around?
- Time to transfer N blocks would be:
  - S = 4ms + 2ms + .031*N
- As N gets big, 1/Transfer → 125MB/sec

Sequential operations

- Assume all sectors to be read are on the same track
  - We may need to seek the track
  - And rotate to the first sector
- But no rotation/seeking is needed afterward

Sequential vs. random

- Consider disk with 7ms avg seek, 10,000 RPM platter speed and 50 MB/sec transfer rate, 4KB/block
- Sequential access of 10 MB
  - Completion time = 7 + 3 + 10/50*1000 = 210ms
  - Actual bandwidth = 10MB/210ms = 47.62 MB/s
- Random access of 10 MB (2,500 blocks)
  - Completion time = 2500 * (7 + 3 + 4/50) = 25.2s
  - Actual bandwidth = 10MB / 25.2s = .397 MB/s

Scheduling

- From sequential/random, we can see that order of reads/writes makes a difference
- In practice, we want to get more data then is in a block, so….
  - Smart disk controllers can reorder requests to get better performance
  - Smart disk controller can reorder concurrent requests to get better performance
- Operating system can help

More calculations

- Consider Consider disk with 7ms seek, 10,000 RPM platter speed and 50 MB/sec transfer rate
- Sequential access of 10 MB
  - $S = \frac{amount\ of\ data}{time\ to\ access} = \frac{10\ MB}{(7ms + 3\ ms + \frac{10MB}{50\ MB/sec})} =$

    $\frac{10MB}{210} ms = 47.62\ MB/Sec$
- Random access of 10 Kb
  - R= $\frac{10Kb}{7ms + 3ms + \frac{.10MB}{50MB/sec}} = \frac{10KB}{10.195ms} = .981\ M/Sec$

Solid State Disks

- All electronic, made from persistent memory
- Lower energy consumption
- Significantly more expensive, less capacity
  - About a factor of 10 more expensive
- Limited lifetime, can only write a limited number of times.
  - ~2000-3000 times

Solid State Drive

Chips

M.2 Form Factor

Solid State Disks

- All electronic, made from persistent memory
- Lower energy consumption
- Significantly more expensive, less capacity
  - About a factor of 10 more expensive
- Limited lifetime, can only write a limited number of times.
  - E.g., 100, 000 write cycles for SLC memory

Solid State Disks

- Same form-fact and control interface as magnetic disks
- Significantly better latency
  - No seek or rotational delay
- Consistent bandwidth
  - Benefits from improved latency
  - Writes take longer then reads

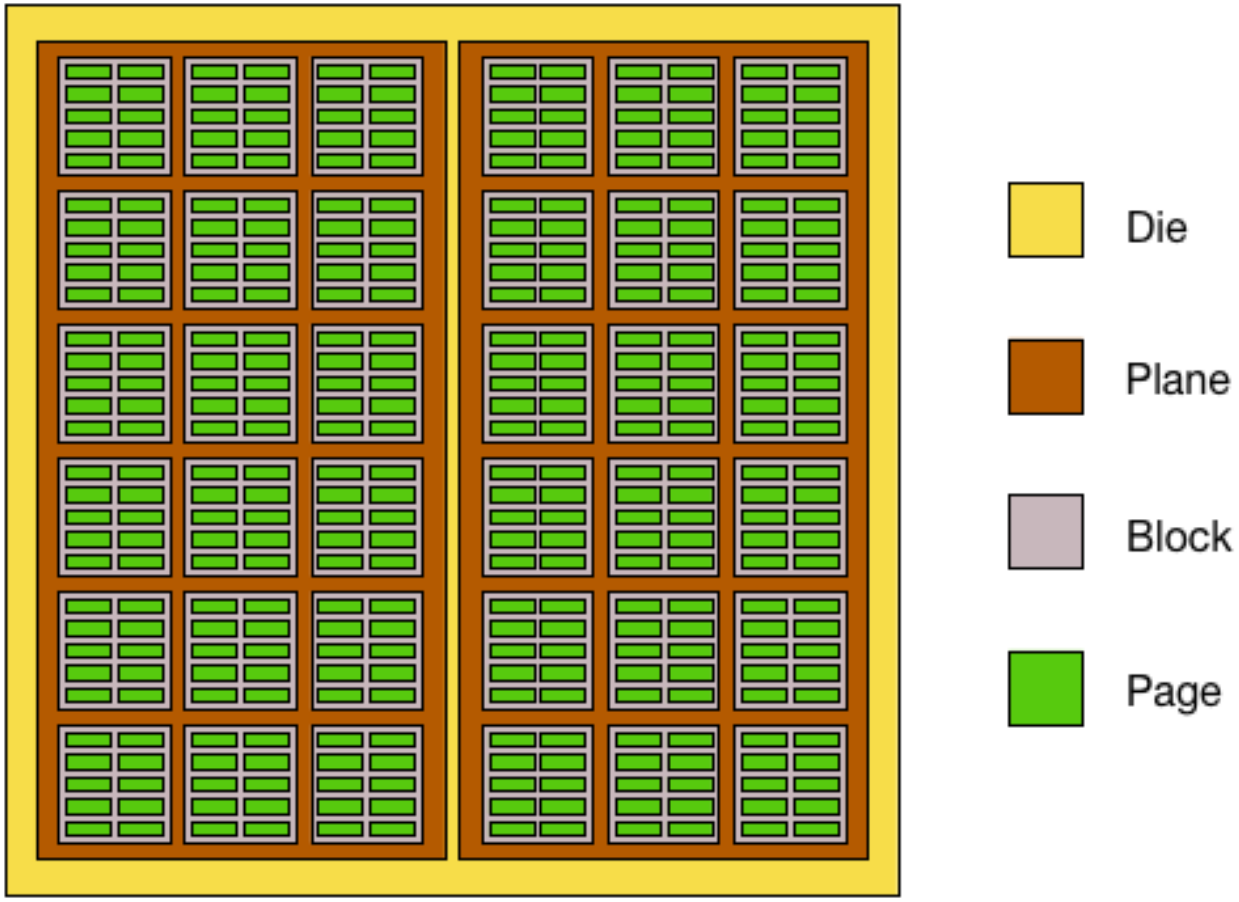
Writing to SSD is complicated

- Cannot overwrite a block, must be erased first.
- SSD organized into 8KB pages, grouped into 256*8 KB = 2MB blocks.
- SSD controllers take care of all these details

SSD

- Contains a number of flash memory chips
  - Chip -> dies -> planes -> blocks -> pages (rows) -> cells
  - Cells are NAND/NOR gates made of transistors
- Page is the smallest unit of data transfer between SSD and main memory
  - Much like a block in hard disk

NAND Flash Die Layout



Understanding Flash: Blocks, Pages and Program / Erases

Dies, planes, block, and pages

- Typically, a chip may have 1, 2, or 4 dies
- A die may have 1 or 2 planes
- A plane has a number of blocks
  - Block is the smallest unit that can be erased
- A block has a number of pages
  - Page is the smallest unit that can be programmed/written to

Typical page and block sizes

- Common page sizes: 2K, 4K, 8K, and 16K

- A block typically has 128 to 256 pages

=> Block size: 256KB to 4MB

Write vs. erase

- Page is the smallest unit that can be read or written (also called programmed)
- Block is the smallest unit that can be erased
  - i.e., make cells "empty" (storing default values)

Operation	Area
Read	Page
Program (Write)	Page
Erase	Block

Write vs. overwrite

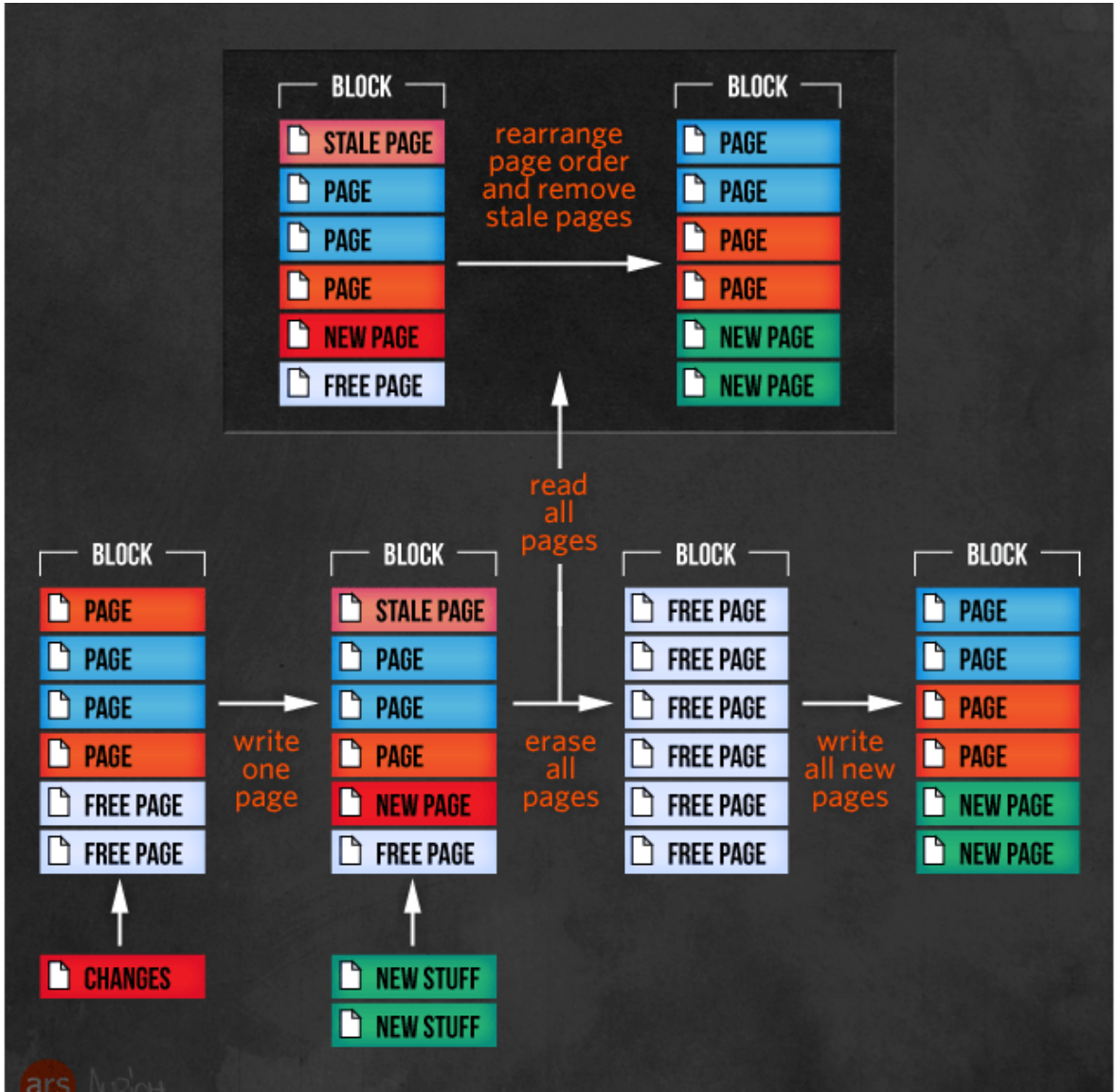
- Write = change 1 to 0
  - Need to apply voltage to the gate
- Overwrite/update = change 0 back to 1
  - Need to apply much higher voltage
  - May stress surrounding cells
  - So dangerous to do on individual pages

P/E cycle

- P: program/write
- E: erase

- P/E cycle:
  - Data are written to cells (P): cell value from 1 -> 0
  - Then erased (E): 0 -> 1

Writing to an SSD



Latencies: read, write, and erase

	SLC	MLC	TLC	HDD	RAM
P/E cycles	100k	10k	5k	*	*
Bits per cell	1	2	3	*	*
Seek latency (µs)	*	*	*	9000	*
Read latency (µs)	25	50	100	2000-7000	0.04-0.1
Write latency (µs)	250	900	1500	2000-7000	0.04-0.1
Erase latency (µs)	1500	3000	5000	*	*
Notes	* metric is not applicable for that type of memory				
Sources	P/E cycles [20] SLC/MLC latencies [1] TLC latencies [23] Hard disk drive latencies [18, 19, 25] RAM latencies [30, 52] L1 and L2 cache latencies [52]				

Next Up?

- Improving the performance over a single disk…RAID!