

Introduction & Disk Drive Operation

Lecture 1
September 12, 2006

About Jiri

- Undergraduate/Master's – MIT
 - Electrical Engineering and Computer Science
- IBM, *Boblingen, Germany*
 - Software Engineer in S/390 I/O microcode group
- PhD in 2003 – Carnegie Mellon *Pittsburgh, PA*
- EMC, *Hopkinton, MA*
 - Centera System Architect
 - Content-addressable storage system
- Network Appliance
 - Research staff in Advanced Development



Course Info on CSG389

- “Information Storage Technologies”
 - Focus on the ecosystem/data centers
- Not “Enterprise Storage Systems”
 - the “nuts and bolts” of Storage Systems
- If you took csg389 before...
 - That’s OK
 - Less than 20% overlap, mostly foundations
- Emphasis on real-world applications
 - ...and big systems

Plan for today

- Class Structure
 - First some administrative stuff
- Storage hierarchy
 - Our goals and why you should care
- Disk components
 - How it's going to work
- On-disk data organization
 - what they're all about
- Disk Technology Trends

Class Structure

- Discussions
 - ... and lots of it
 - interested in hearing what YOU have to say and learned
- Reading primary & secondary sources
 - No textbook, may assign some chapters from a book on reserver in the library
 - Read materials beforehand
 - Posted on the web usually the next day after a class
- Homeworks
 - Reading materials before class
 - Writing assignments
 - Practical assignments

Collaboration

- Encouraged to discuss class materials
 - Especially in the class
 - But no collaboration allowed on HWs
 - Give credit where credit is due
- Exams
 - Two (midterm and final)
 - Want to know what you individually learned
 - Closed books
 - I am interested in your thoughts: how you put concepts together not on memorizing facts and putting them on paper

Scope and Breadth

- I expect you already know:
 - Basics of networking (Ethernet, TCP/IP)
 - Basics of OS - virtual memory, filesystems, synchronization primitives
- Will teach distributed systems concepts
 - ... to build (small, but) solid foundations for later on
 - No rigorous theory or proofs
- Build upon the foundations and apply concepts
 - Limited coding/development (no large lab assignments)
- Focus on technology, how it shapes systems research

List of Topics

- Storage technologies basics
 - Disk drives and technology trends
- Distributed and Clustered Computing
 - Basics of distributed systems - consensus
 - Fault models (Byzantine, fail-stop)
- Database systems
 - Basic architecture
 - Logging, locking, ACID properties
 - Basics of data organization (data layout)
- Applications
 - Traditional DB workloads (OLTP, DSS)
 - Amazon, Google, Yahoo, eBay...

List of Topics cont'd

- Disk arrays
 - Basic architecture and mission-criticality
- Clustered storage systems (bricks)
 - Oceanstore, FARSite, FAB, Google File System
- Data Management: Protection
 - Disaster Prevention and Recovery (Backups)
 - Long-term archival and document retention
- Data Management: Predictability
 - Performance expectations and service level agreements
- Security
 - Preventing unauthorized access

List of Topics cont'd 2

- Interconnect Technologies
 - Software and protocols (SAN and IP networks)
 - Hardware
- Abstraction and modularization
 - Building systems out of components
 - JAVA, JNI
- Virtualization
 - ...and what it buys users
 - IBM mainframes, Virtual Machines and Xen
- Leading edge
 - Object stores
 - CAS systems
- Others?

What Is a Storage System?

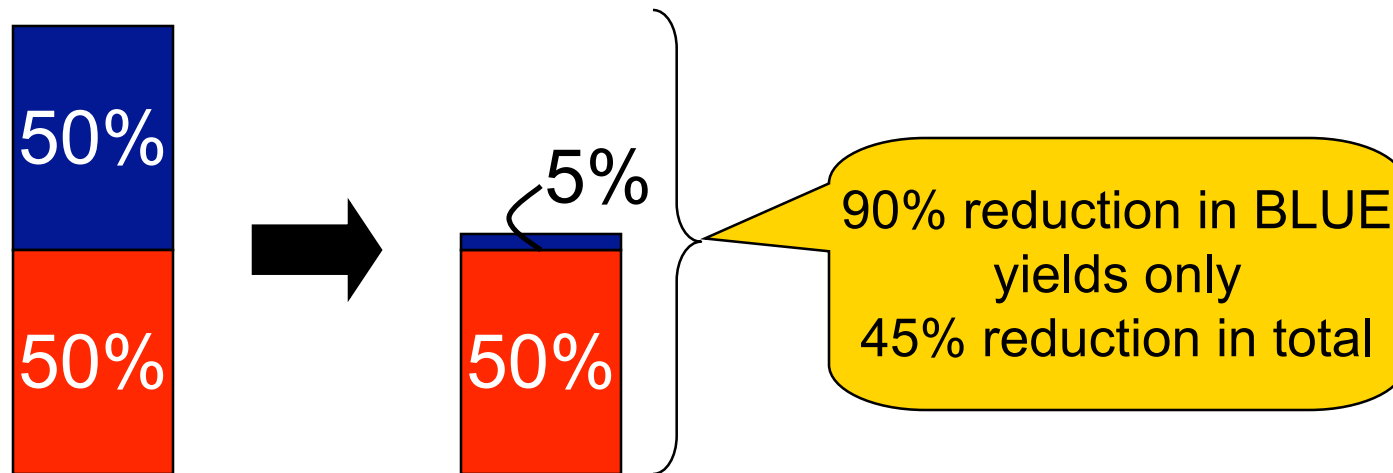
- Hardware (devices, controllers, interconnect) & software (file system, device drivers, firmware) dedicated to providing management of and access to persistent storage.
- Different views of storage systems
 - Logical: defined by functional interfaces
 - Program \leftrightarrow File System \leftrightarrow Controller Firmware
 - Physical: defined by component technologies
 - CPU \leftrightarrow I/O Controller \leftrightarrow Device ASIC

What Makes Storage Systems Unique

- Combine so many topic areas
 - hardware design, local and distributed operating systems, networking, performance analysis ...
- Still so much room to contribute
 - performance actually matters here
 - it may dominate system performance in many cases
 - simplifying/automating storage management
 - 6-10 dollars spent for every \$1 on hardware
 - dealing with heterogeneity
 - Linux/Windows clients
 - broadband/consumer networks vs. data center LAN
 - helping users find desired information
 - web & growing disk capacity cause information overload

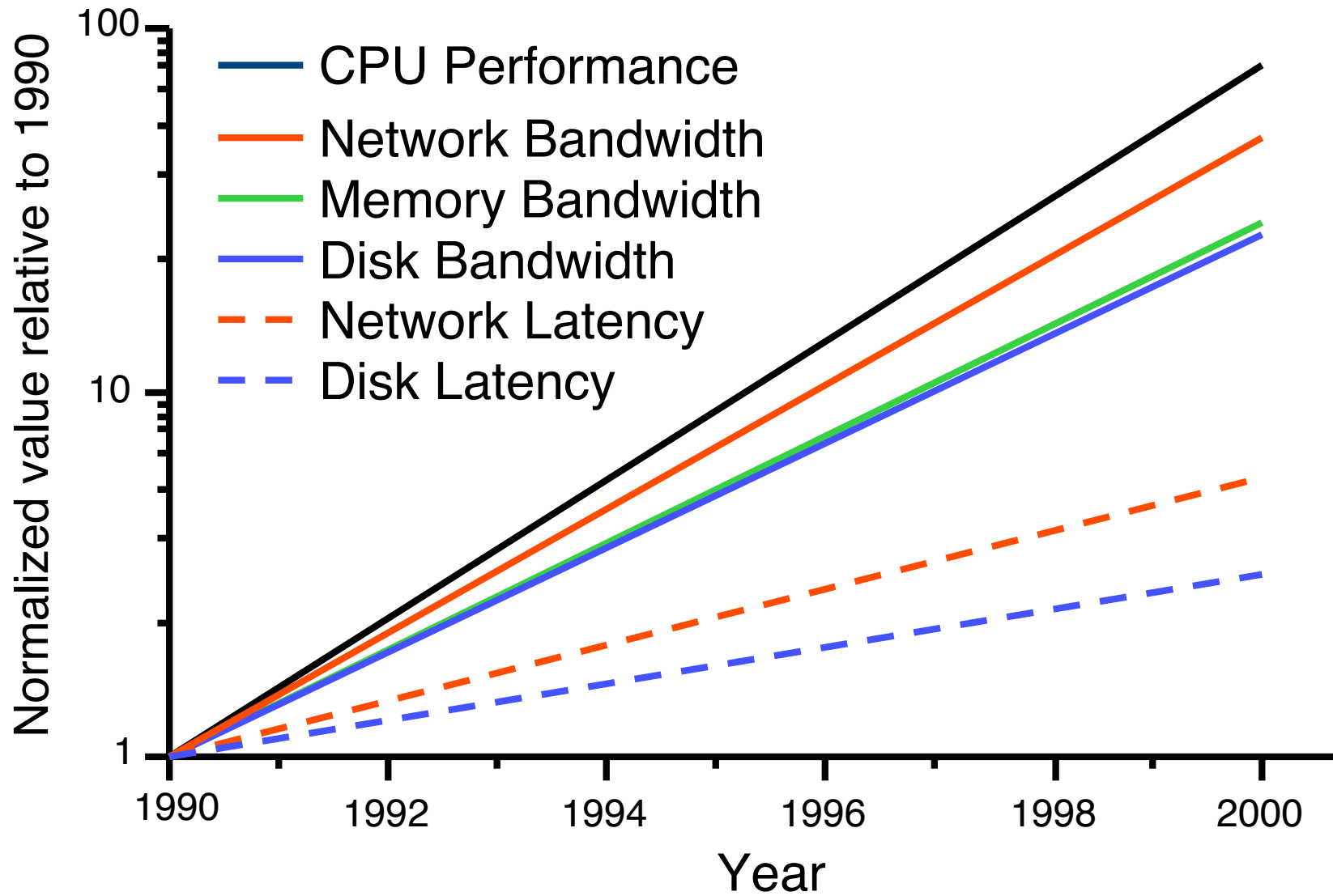
Performance: Amdahl's Law

- Speedup limited to fraction improved
 - obvious, but fundamental, observation



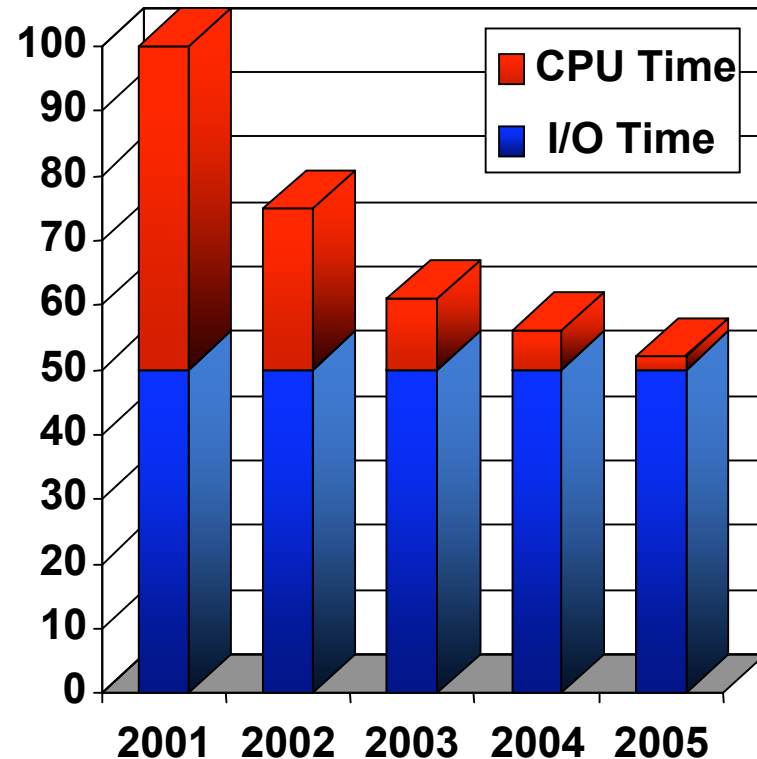
- Consequence for storage systems
 - this has been going on for years!

Technology Trends



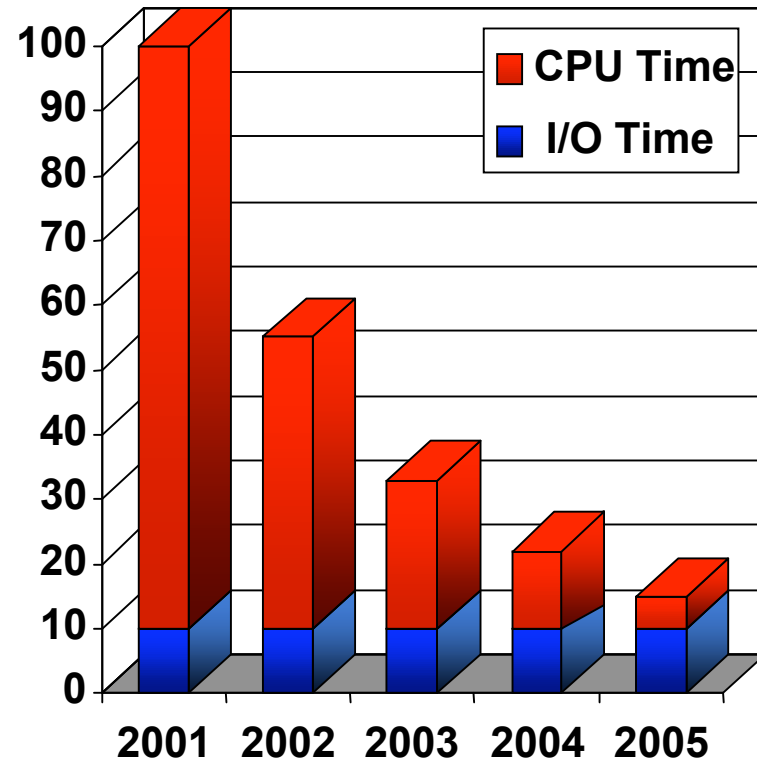
Storage Performance Dominates

- Assume 50 seconds CPU & 50 seconds I/O
- by 2002
 - CPU improves by: $N = 50/25 = 2$
 - Program performance improves by:
 $N = 100/75 = 1.33$
- by 2003
 - CPU performance - factor of 2
 - Program performance $N = 75/62.5 = 1.2$
- by 2004
 - CPU performance - factor of 2
 - Program performance $N = 62.5/56.5 = 1.1$
- Example of Amdahl's Law



Even for Once CPU-bound Workloads

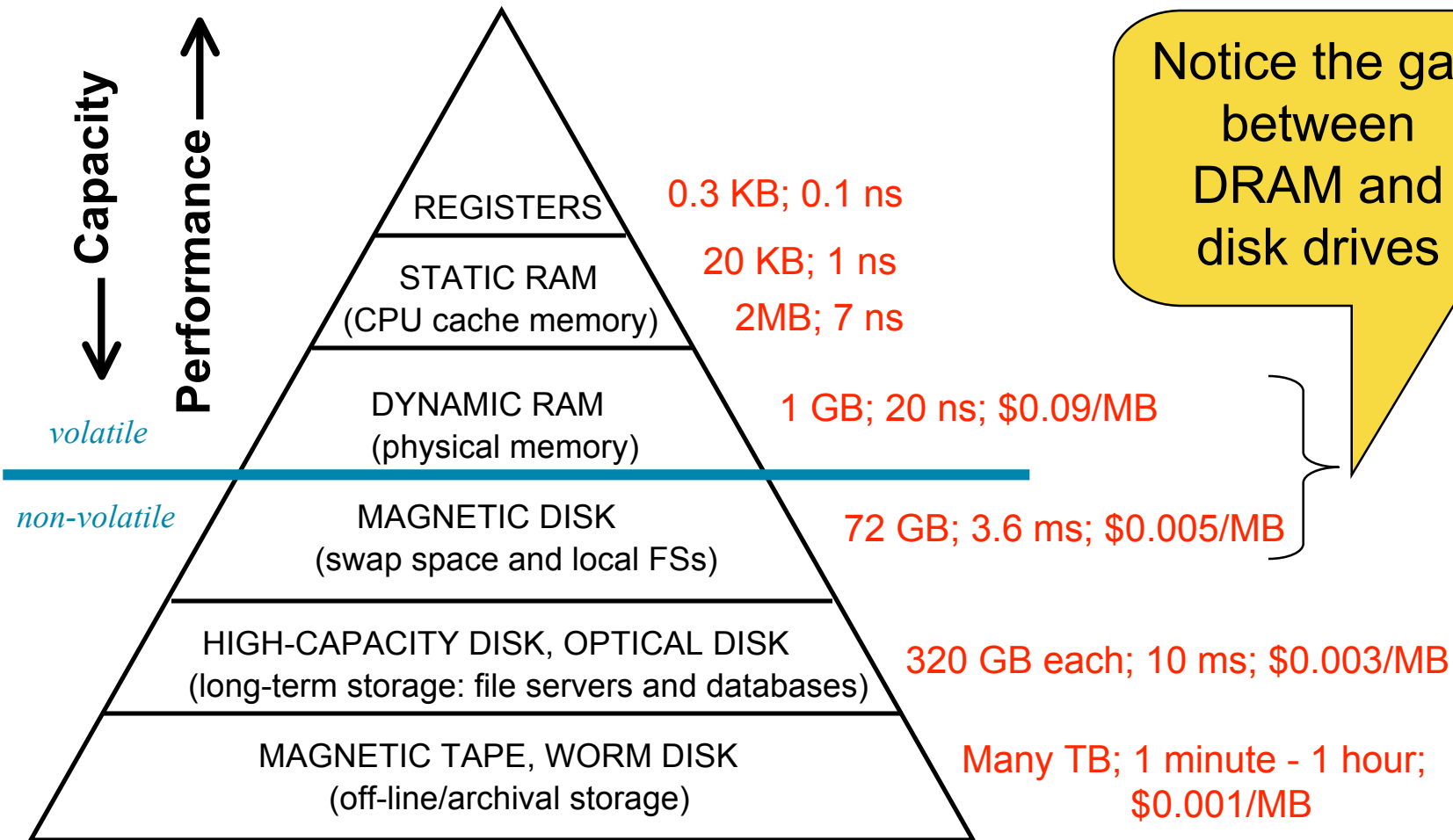
- by 2002
 - CPU improves by $N = 90/45 = 2$
 - Program performance improves by $N = 100/55 = 1.81$
- by 2003
 - CPU performance - factor of 2
 - Program performance $N = 55/32.5 = 1.7$
- by 2004
 - CPU performance - factor of 2
 - Program performance $N = 32.5 / 21.25 = 1.53$
- by 2005
 - CPU Performance - factor of 2
 - Program performance $N = 21.25 / 15.6 = 1.36$



Memory/storage Hierarchies

- Combine technologies to balance costs and performance benefits
 - small memories are **fast but expensive**
 - large memories are **slow but cheap**
- Exploit locality to get the best of both worlds
 - locality = re-use/nearness of accesses
 - allows most accesses to use small, fast memory
- Locality is a general concept
 - power, management, etc.

Example Memory Hierarchy Values



Representative data: 2004

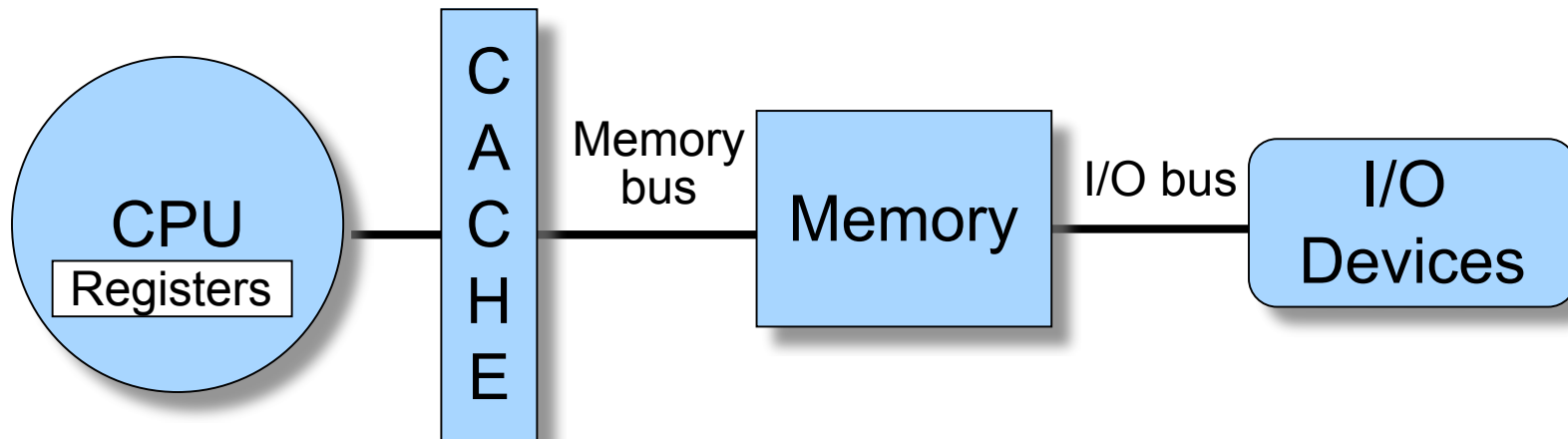
Storage Systems Change Drivers

- Technology
 - eliminates some problems and creates new ones
 - ... and enables new applications over time
 - incommensurate scaling makes things interesting
 - must be on top technology characteristics & trends
- New application requirements
 - changes rules/assumptions, often forcing redesign
 - example: home entertainment vs. database servers
 - example: mobile computing vs. file system caching
- Systems complicated & consist of many parts
 - to do top-quality job, must know about them all!
 - ... and their interactions too.

What Is a Storage System?

- Hardware (devices, controllers, interconnect) & software (file system, device drivers, firmware) dedicated to providing management of and access to persistent storage.
- **Different views of storage systems**
 - **Logical: defined by functional interfaces**
 - Program \leftrightarrow File System \leftrightarrow Controller Firmware
 - **Physical: defined by component technologies**
 - CPU \leftrightarrow I/O Controller \leftrightarrow Device ASIC

Physical View: Memory Hierarchy



Register
Reference

L1/L2 Cache
Reference

Memory
Reference

Disk
Reference

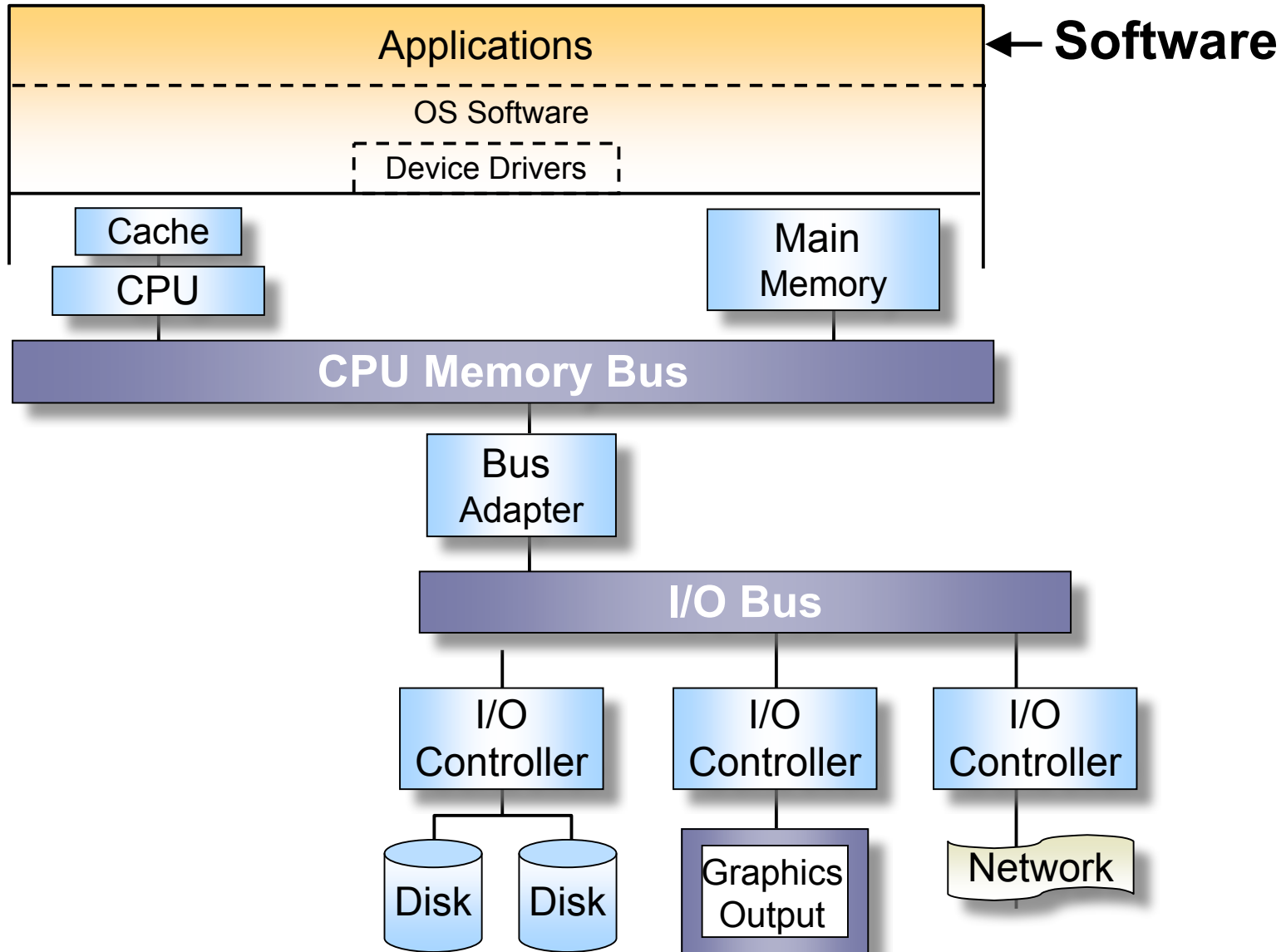
Size: 300 B
Speed: 0.2 ns

20 / 2MB
1 / 5 ns

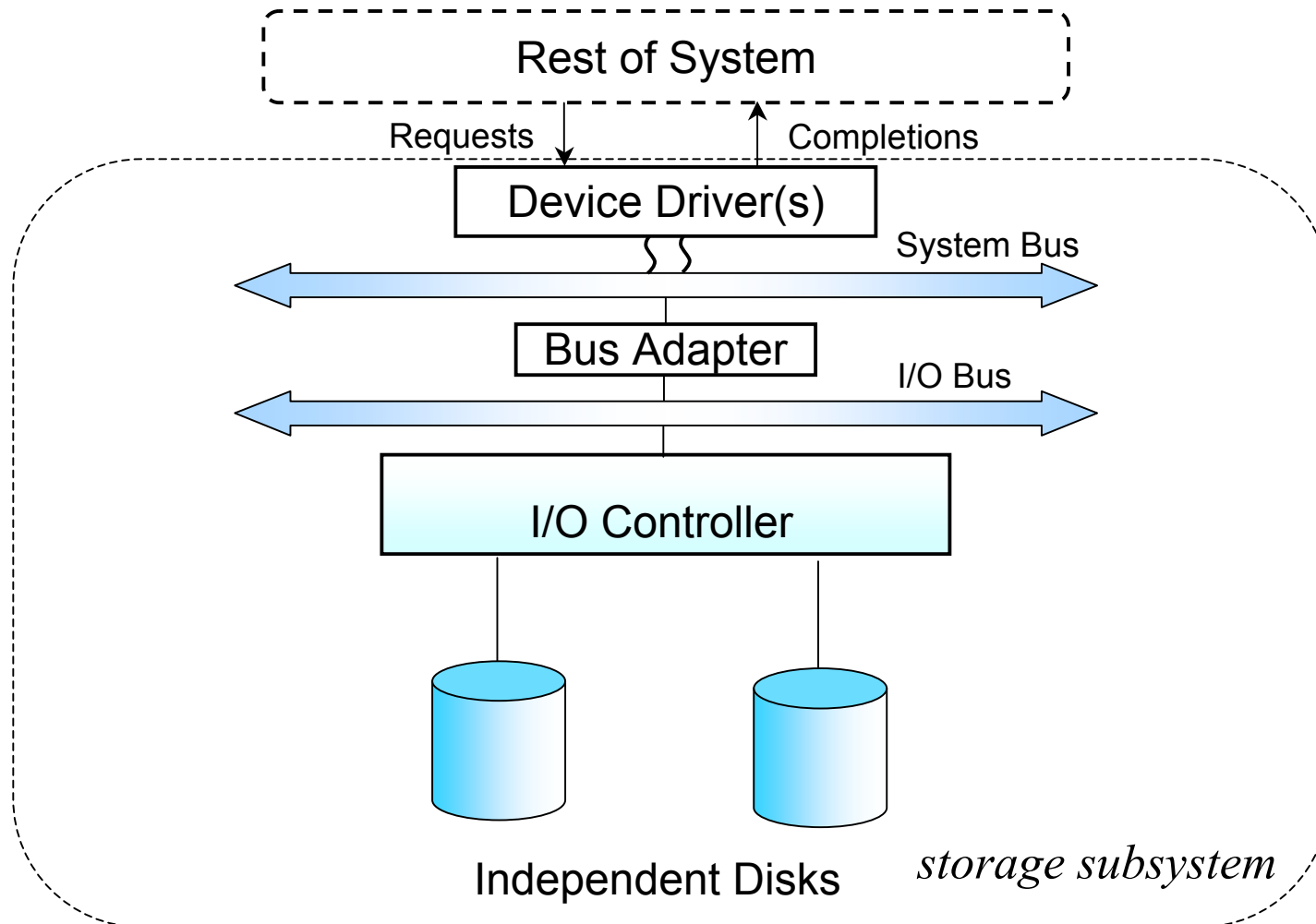
1GB
30 ns

100-500 GB
10 ms

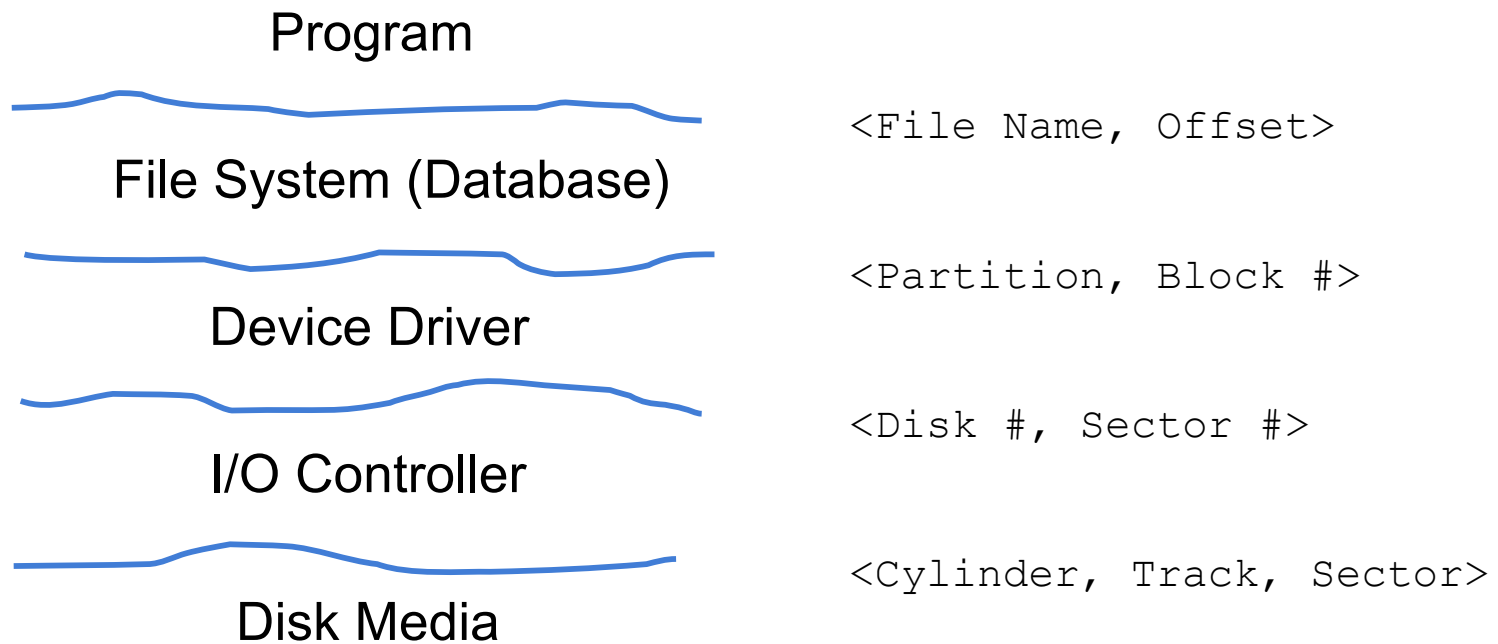
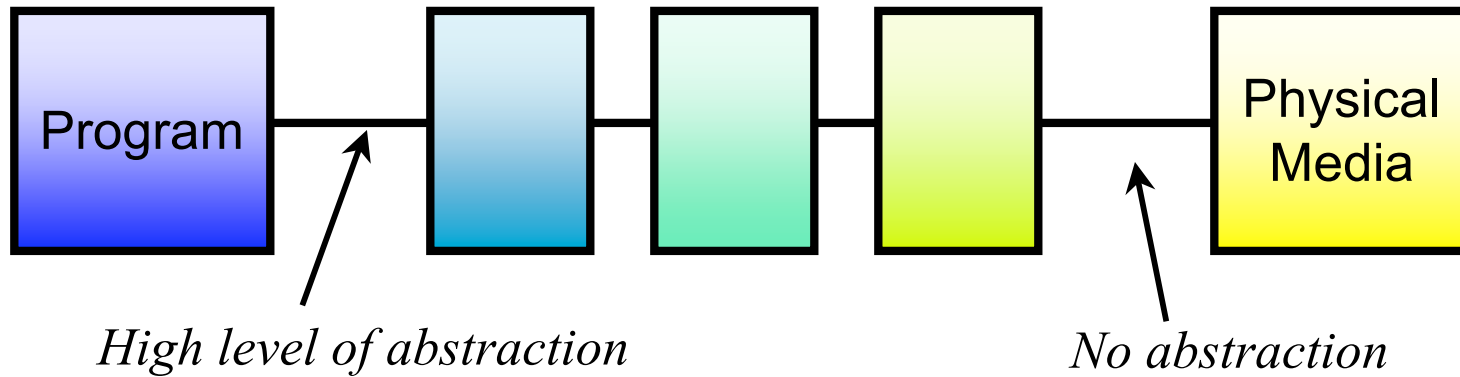
Physical View: Computer System



Physical View: Storage Subsystem

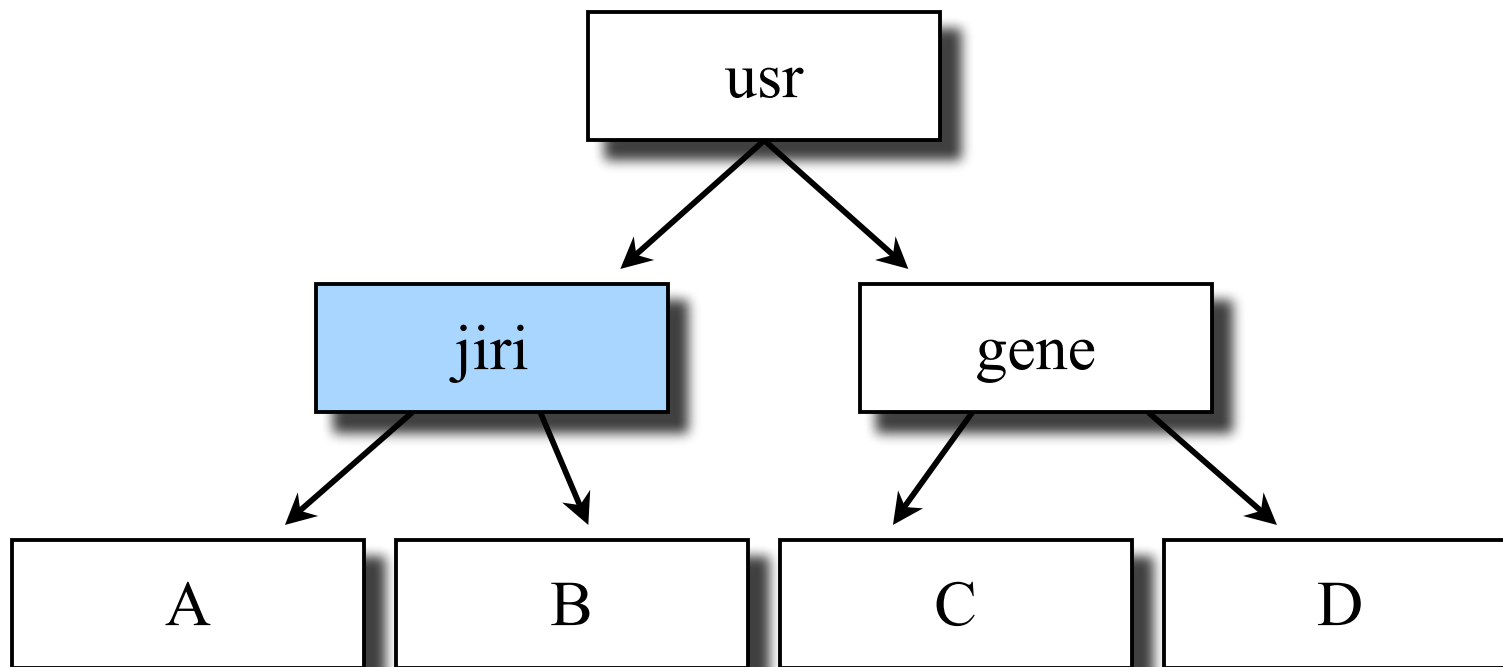


Logical View: Storage Interfaces



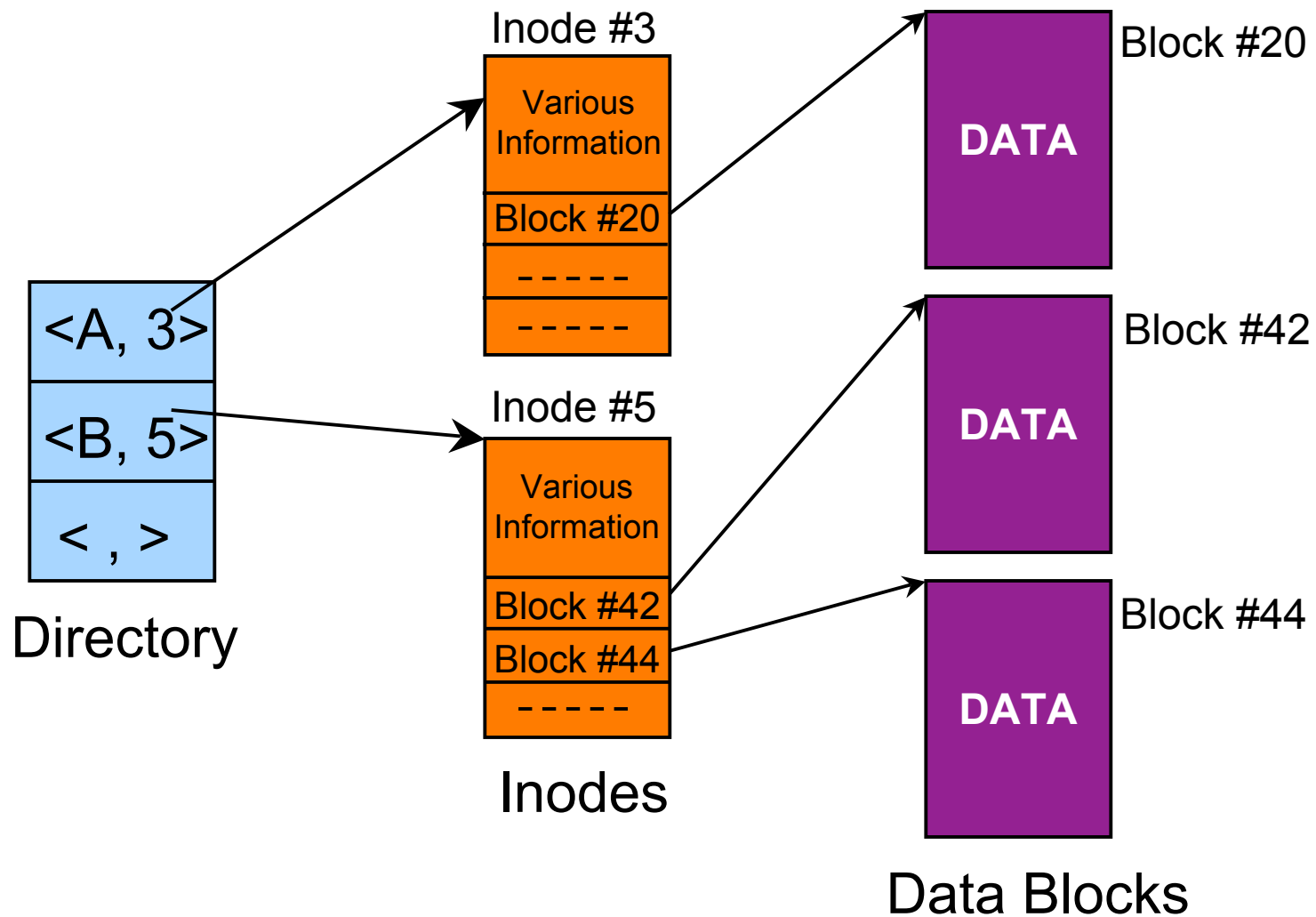
Program's View of File system

Directory Hierarchy



There are more advanced ways of organizing data too...

File system/OS Level Organization



OS's Logical View of Storage Systems

- Linear address space of equal-sized blocks
 - each identified by logical block number (LBN)
 - SCSI or ATA



- Common block size: 512 bytes
- Number of blocks: device capacity / block size
- OS-to-storage requests defined by few fields
 - R/W, block #, # of blocks, memory source/dest

Disk Drive Physical Characteristics



Outline

- Overview of disk components
- Overview of disk operation
- Components in more detail
 - magnetic recording
- Access time in more detail
 - service components

What's Inside A Disk Drive?

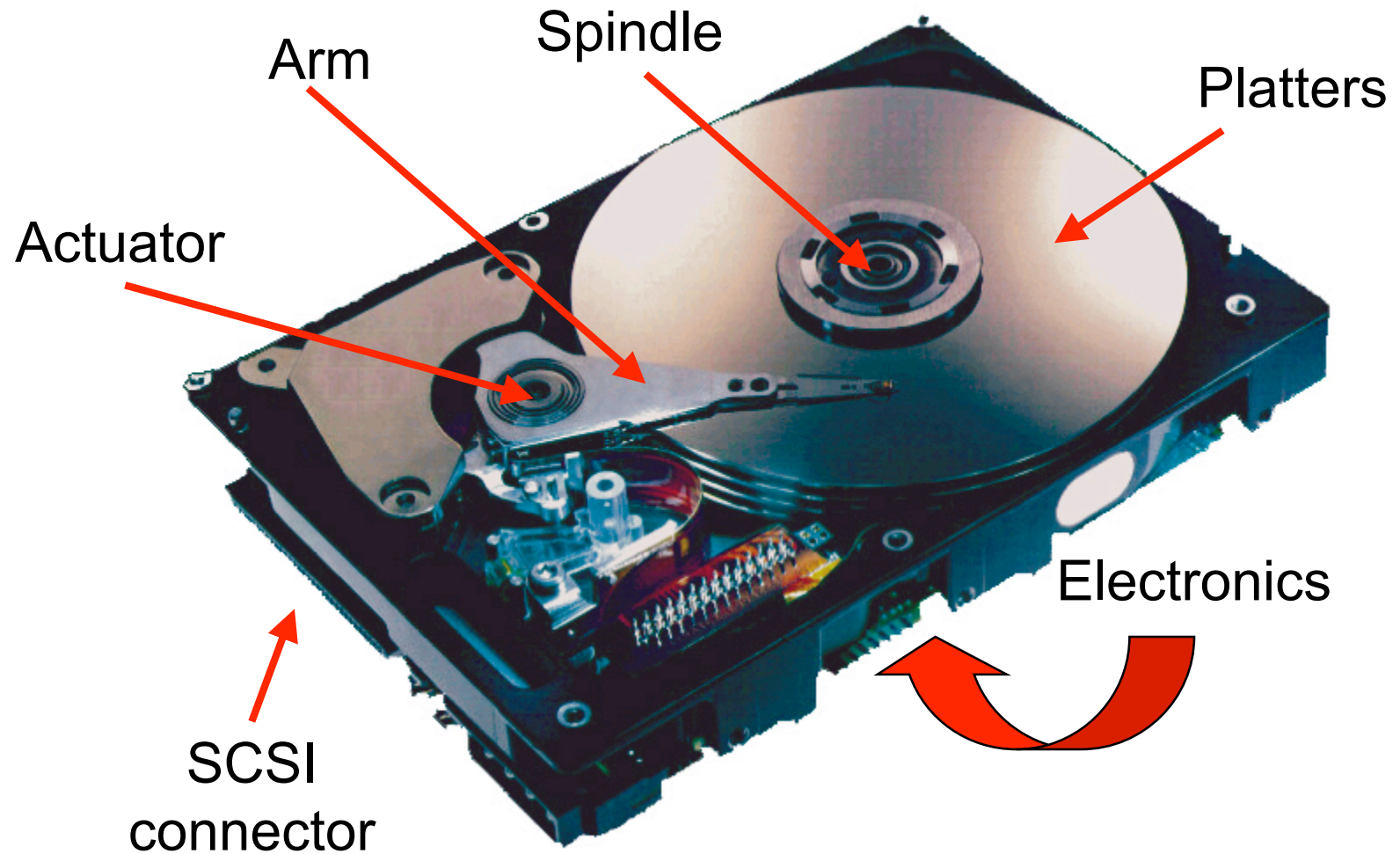
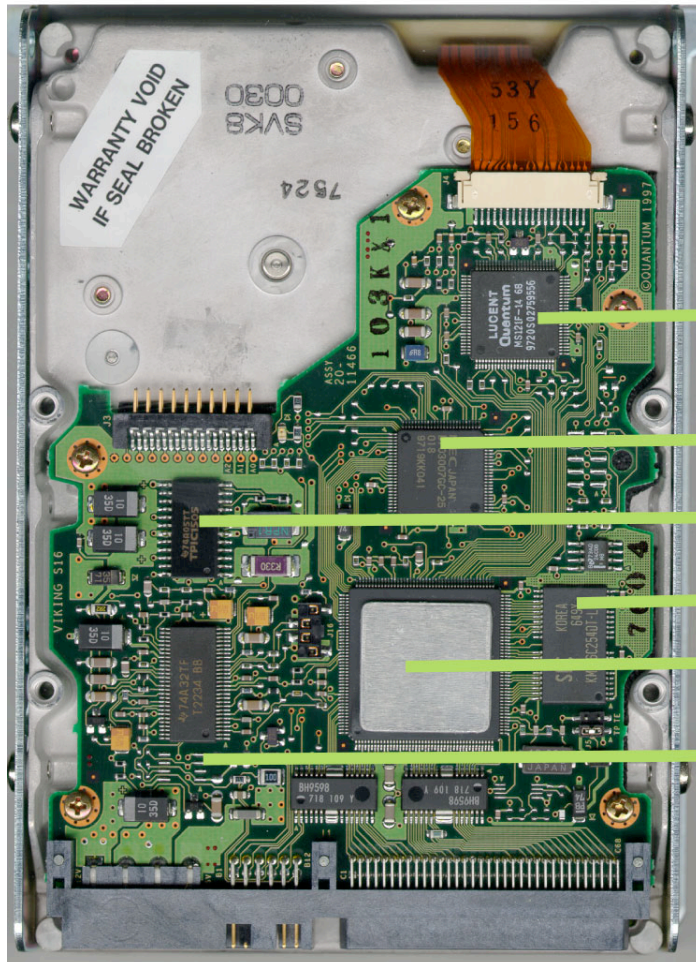


Image courtesy of Seagate Technology

Disk Electronics

Quantum Viking (circa 1997)



6 Chips

Just like a small computer – processor, memory, network iface

R/W Channel

- Connect to disk

uProcessor

32-bit, 25 MHz

- Control processor

Power Array

2 MB DRAM

- Cache memory

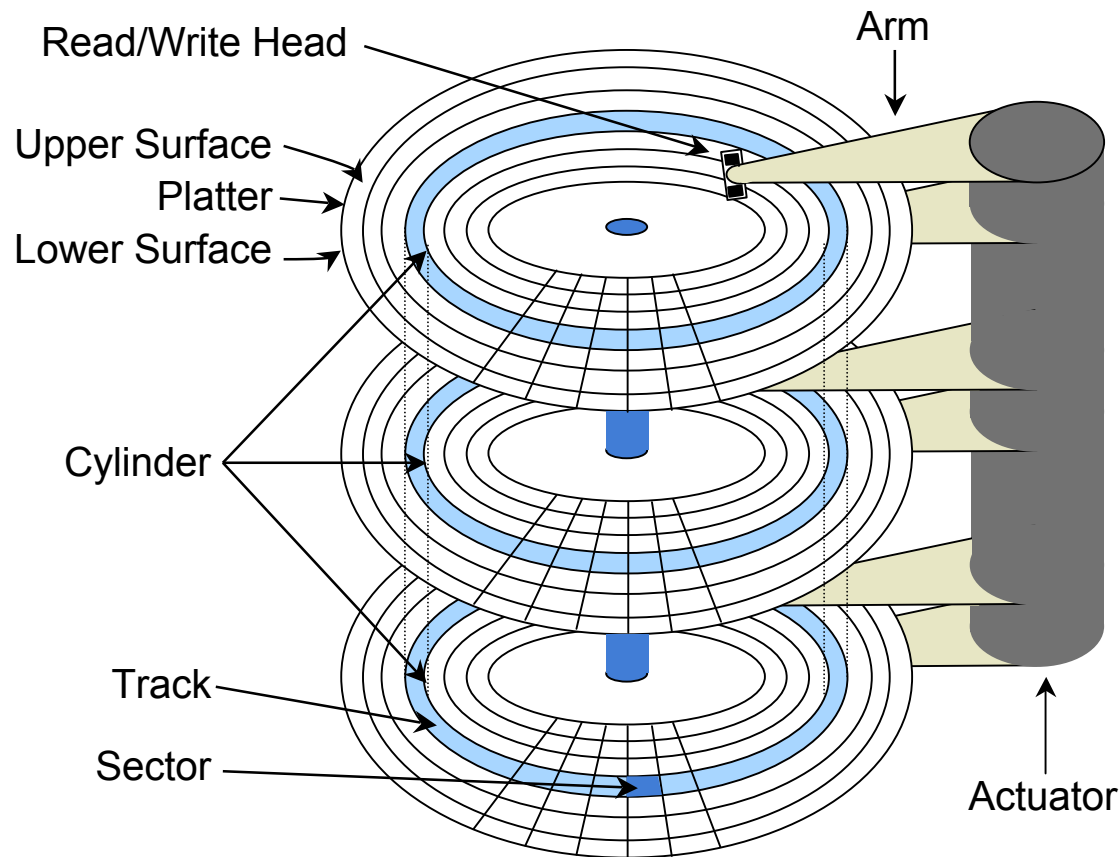
Control ASIC
SCSI, servo, ECC

- Control ASIC

Motor/Spindle

- Connect to motor

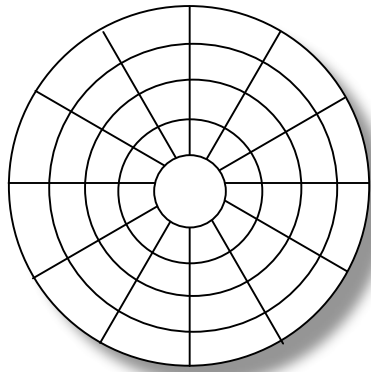
Disk Structure



Seagate Cheetah 15K.3	
model	ST373453
capacity	73.4 GB*
platters	4
heads	8
cylinders	31,310
tracks	250,480

*often GB = 1 billion bytes (10^9), not 2^{30}
(the difference is 7%!)

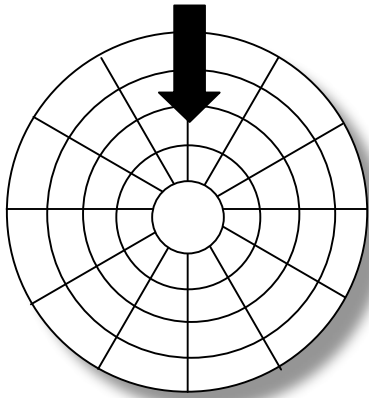
Disk: top view of single platter



Surface organized into tracks

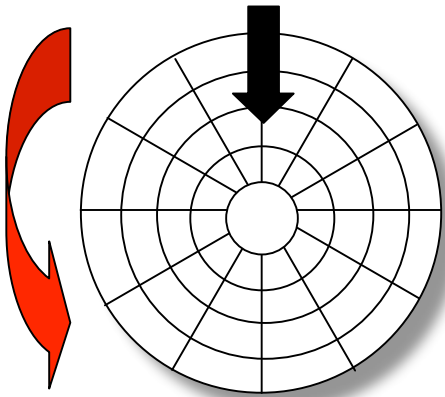
Tracks organized into sectors

Disk Access



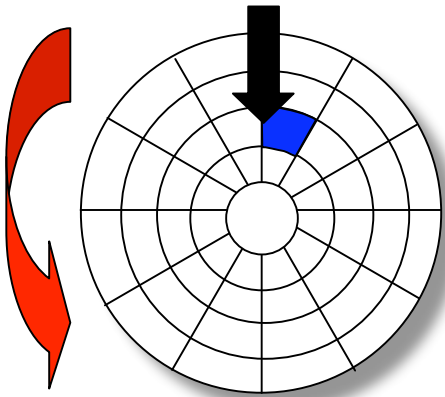
Head in position above a track

Disk Access



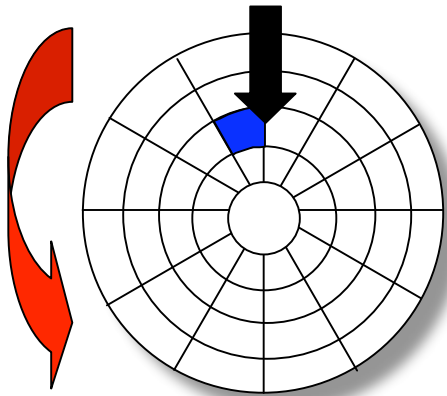
Rotation is counter-clockwise

Disk Access – Read



About to read blue sector

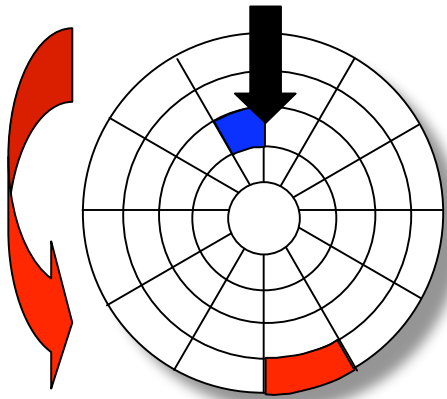
Disk Access – Read



After BLUE read

After reading blue sector

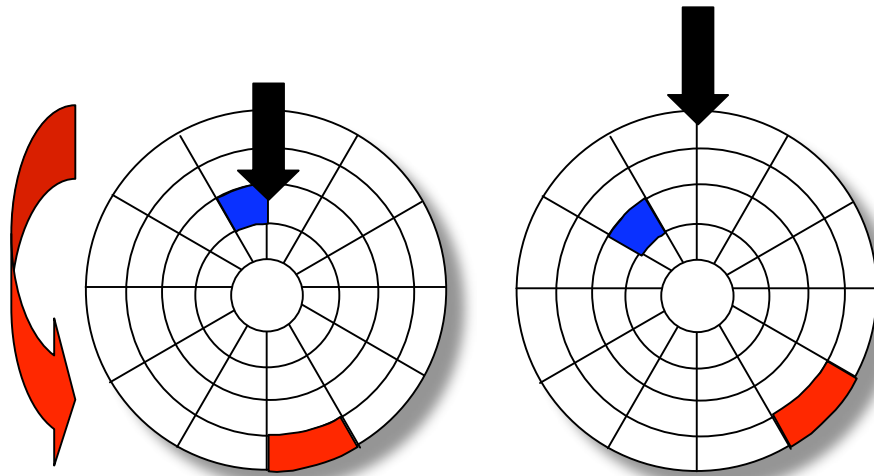
Disk Access – Read



After **BLUE** read

Red request scheduled next

Disk Access – Seek

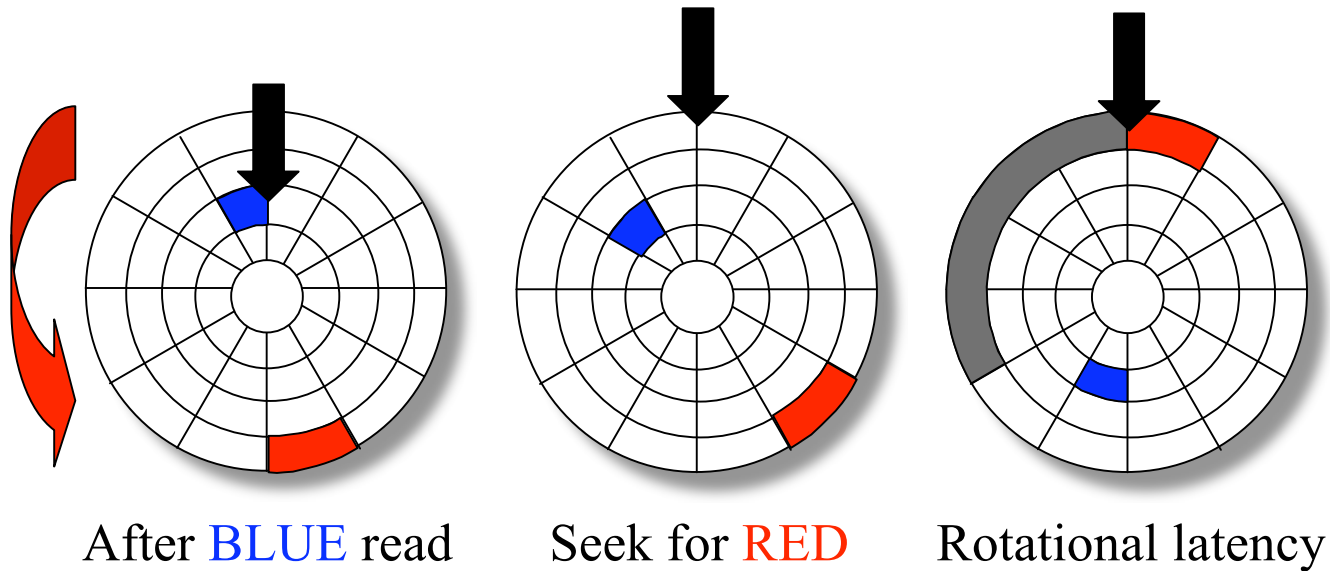


After **BLUE** read

Seek for **RED**

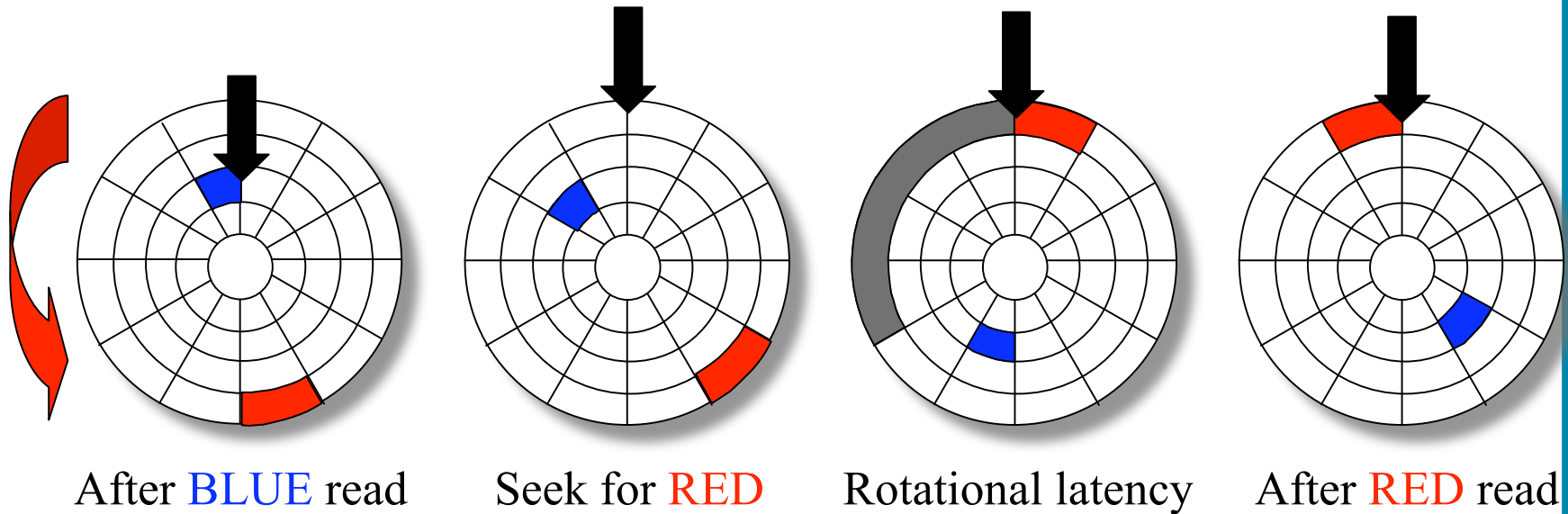
Seek to red's track

Disk Access – Rotational Latency



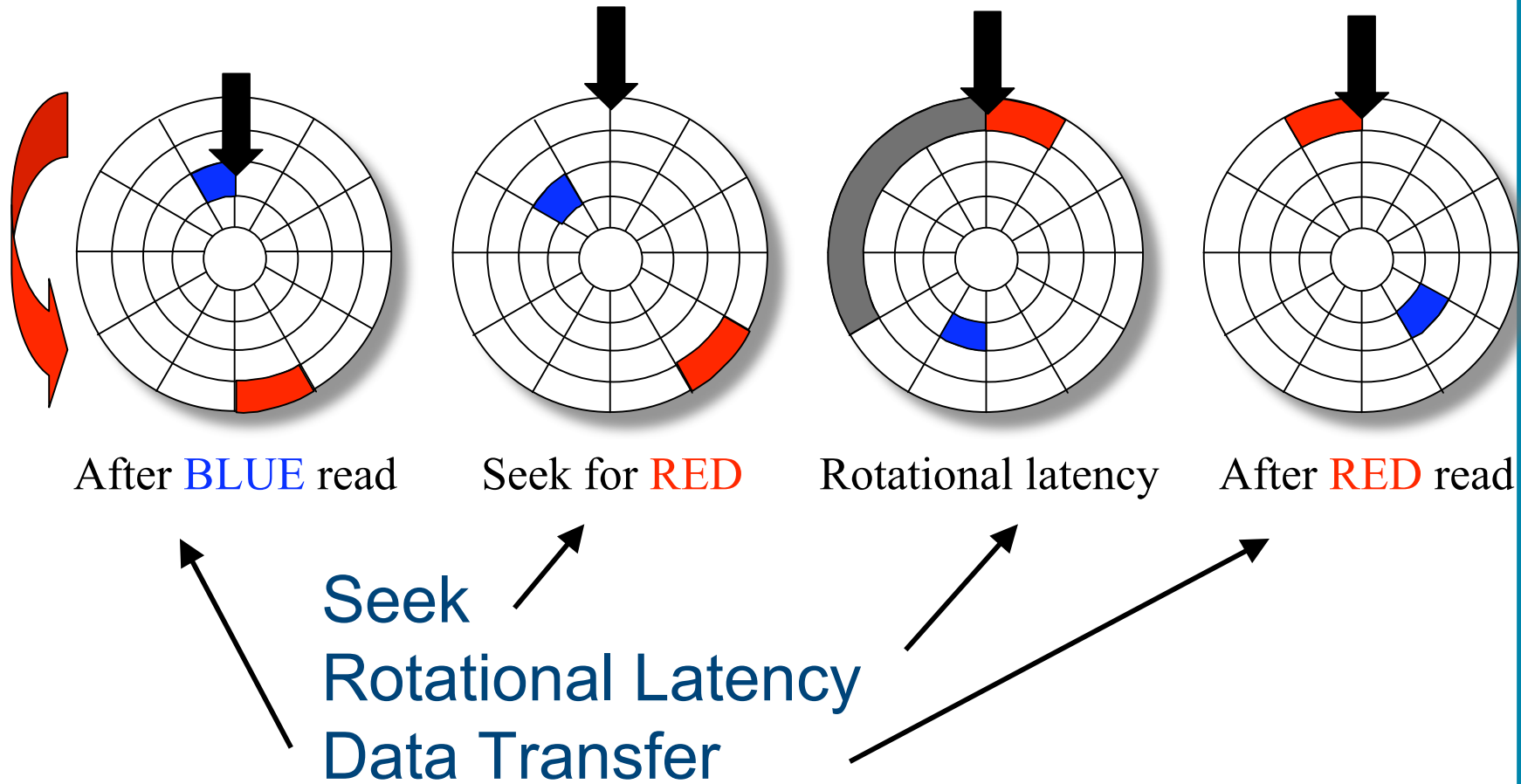
Wait for red sector to rotate around

Disk Access – Read



Complete read of red

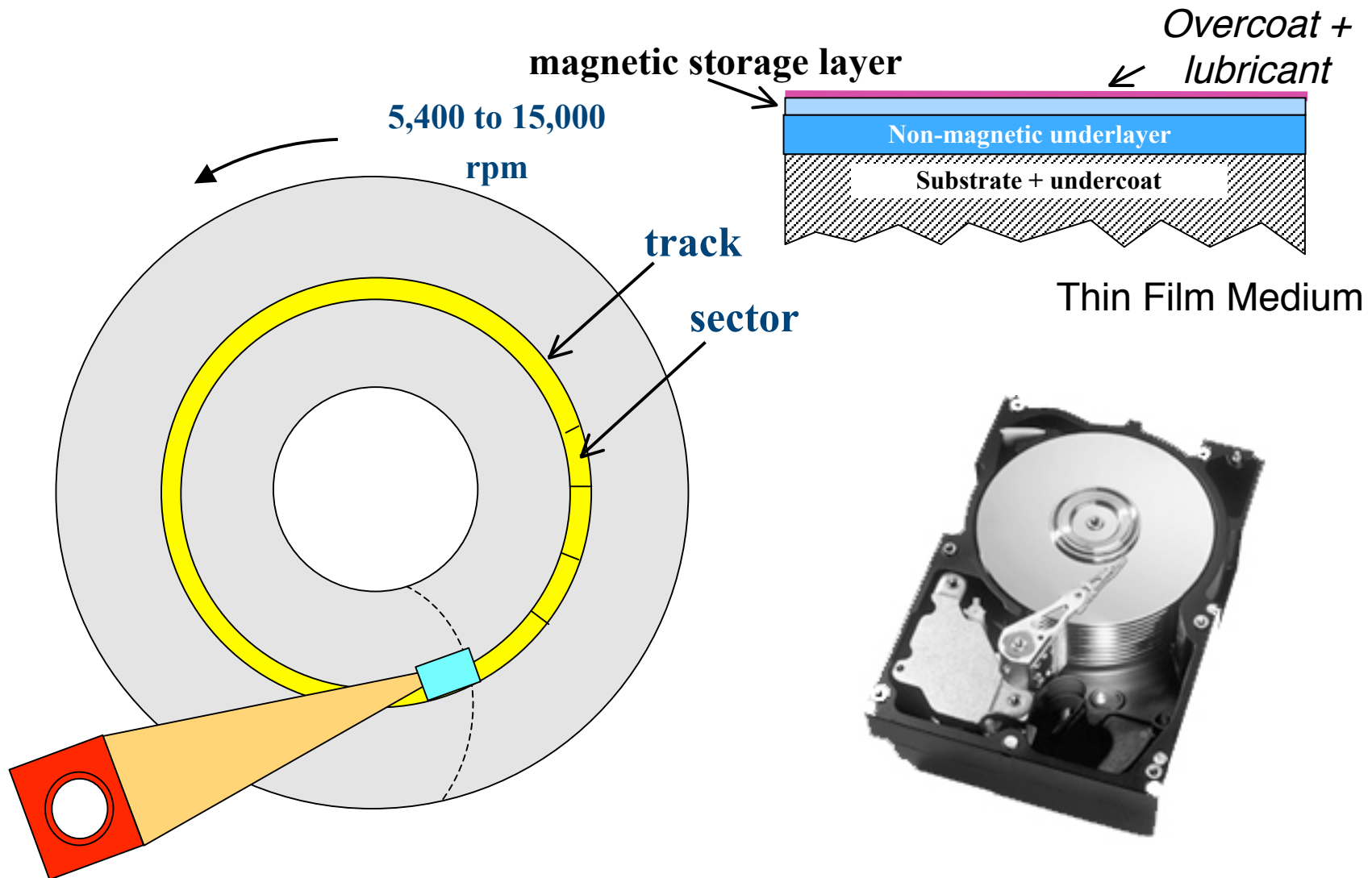
Disk Access – Service Time Components



Disk Recording Components

How the bits get/stay
there

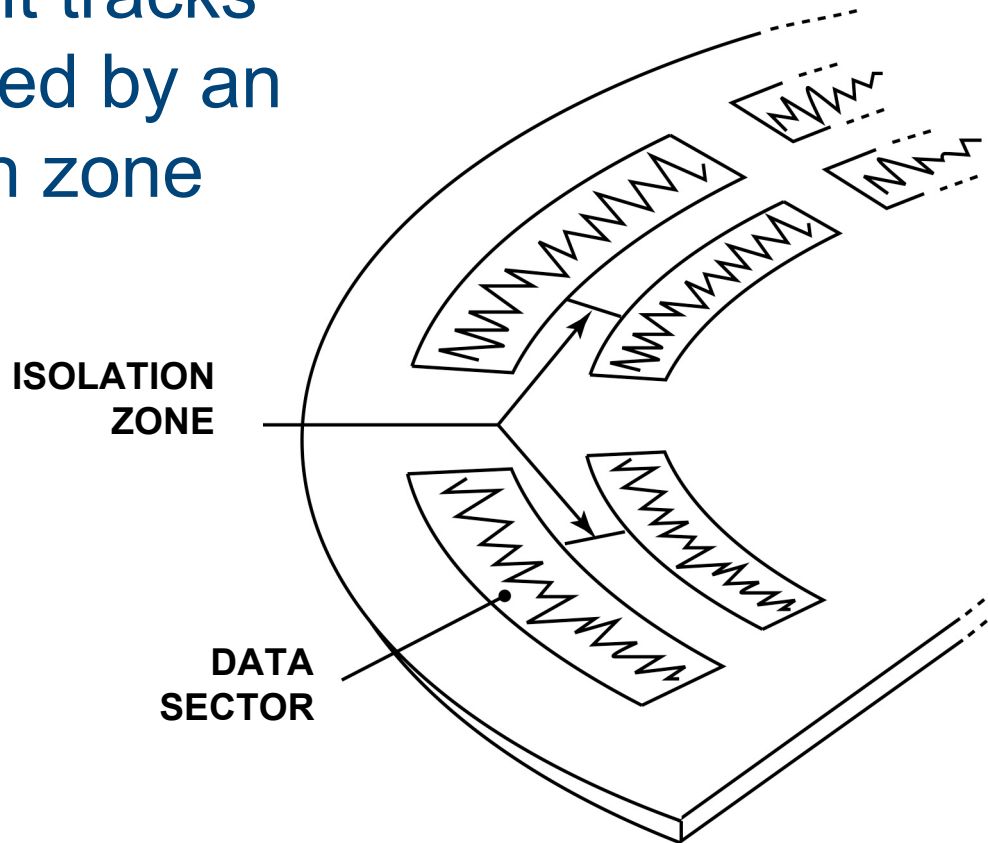
Disk Drive – Media



Courtesy of Jimmy Zhu, CMU

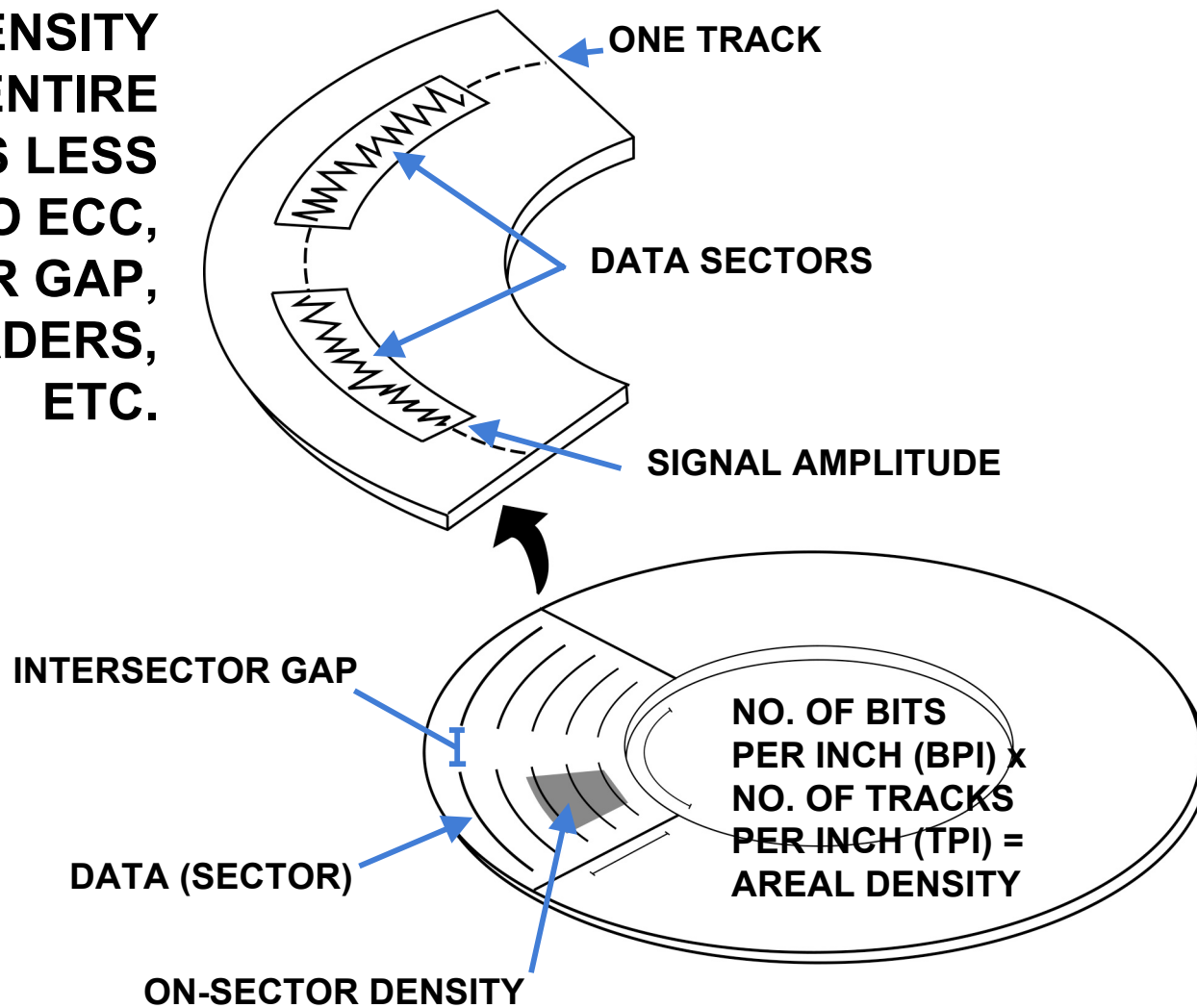
Disk Drive – Servo

Adjacent tracks
separated by an
isolation zone

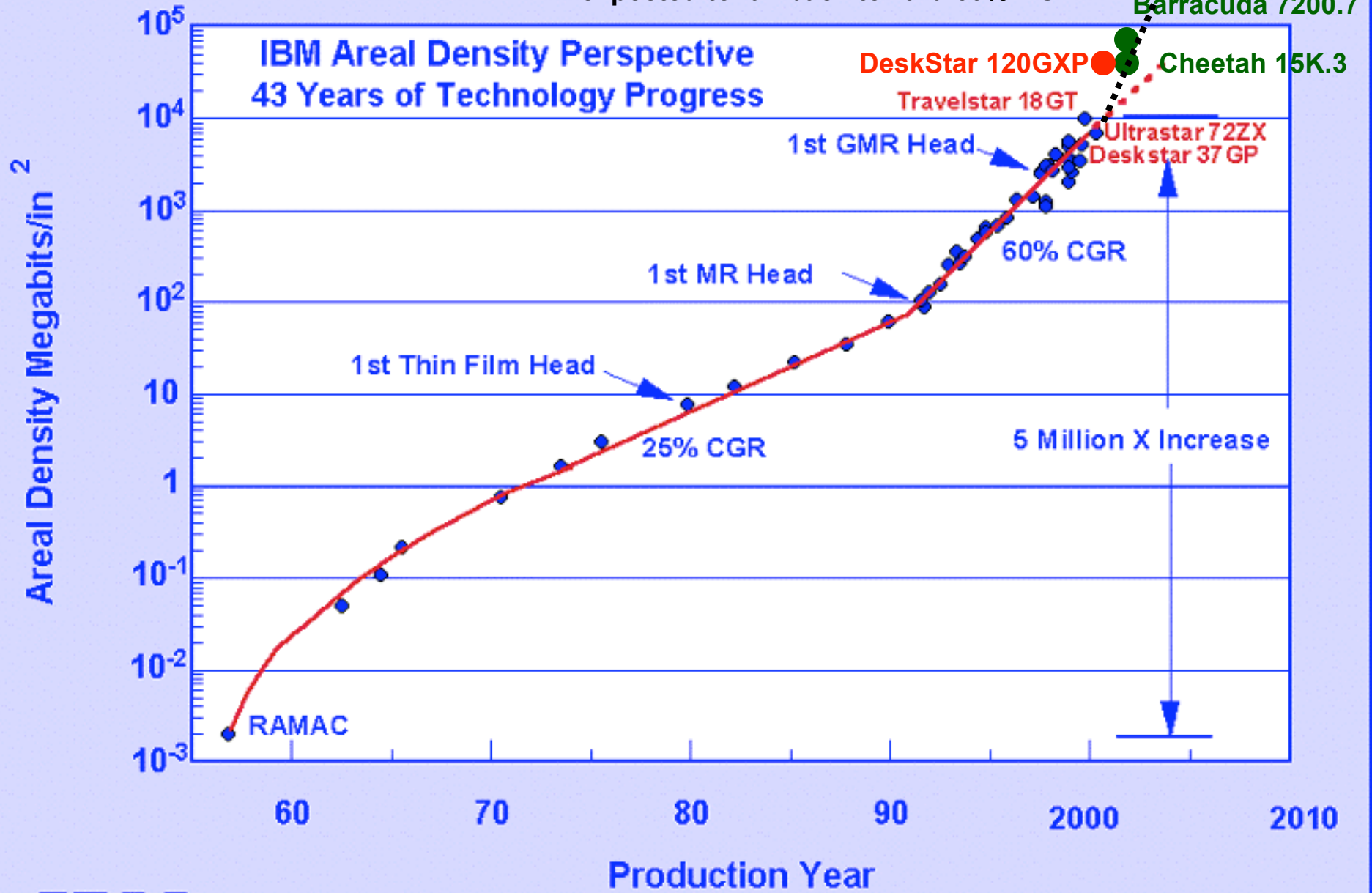


Disk Drive – Density

**AVERAGE DENSITY
ACROSS ENTIRE
SURFACE IS LESS
DUE TO ECC,
INTERSECTOR GAP,
BLOCK HEADERS,
ETC.**



Recent CGR closer to 100%, likely not sustainable, expected to fall back toward 60% CGR



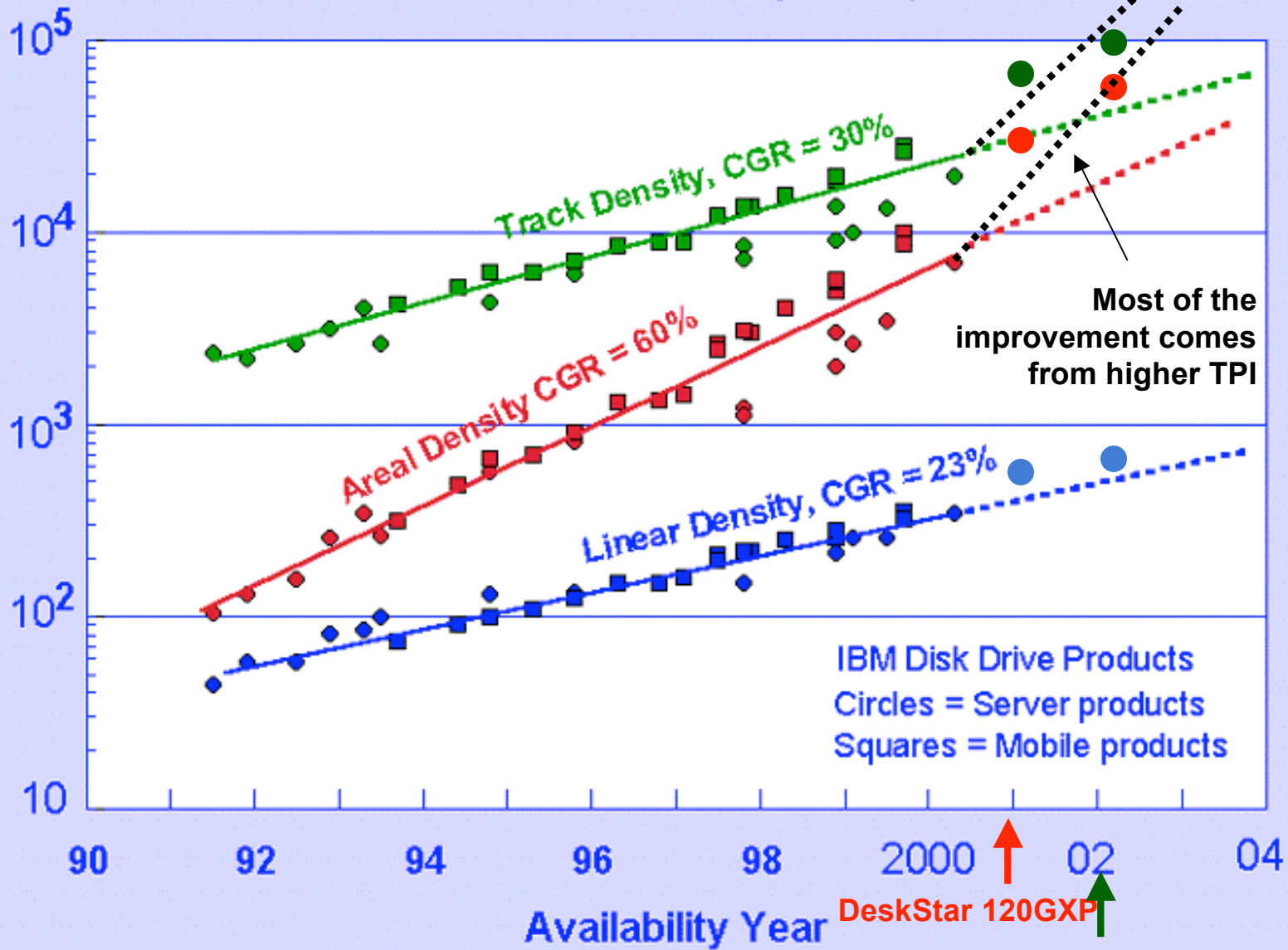
ARPERS2000.PPT



Ed Grochowski at Almaden

Track, Areal, Linear Density Perspective

Track Density, tracks/in
 Areal Density, mbits/in²
 Linear Density, kbits/in



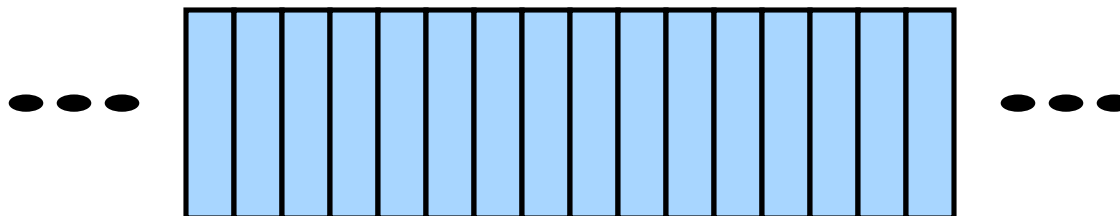
net98a1.prz



Ed Grochowski at Almaden

Disk Drive – Density

- Densities not expected to grow forever
 - superparamagnetic limit
 - when thermal forces swap bits on their own
 - particle size has limits eventually
- Aspect ratios for bits
 - linear density is 10-20X track density
 - a lot of it is the inter-track isolation zone
 - so, track density offers more room for improvement
 - track density doesn't help data transfer and can hurt track-following (it's all about better servo performance)

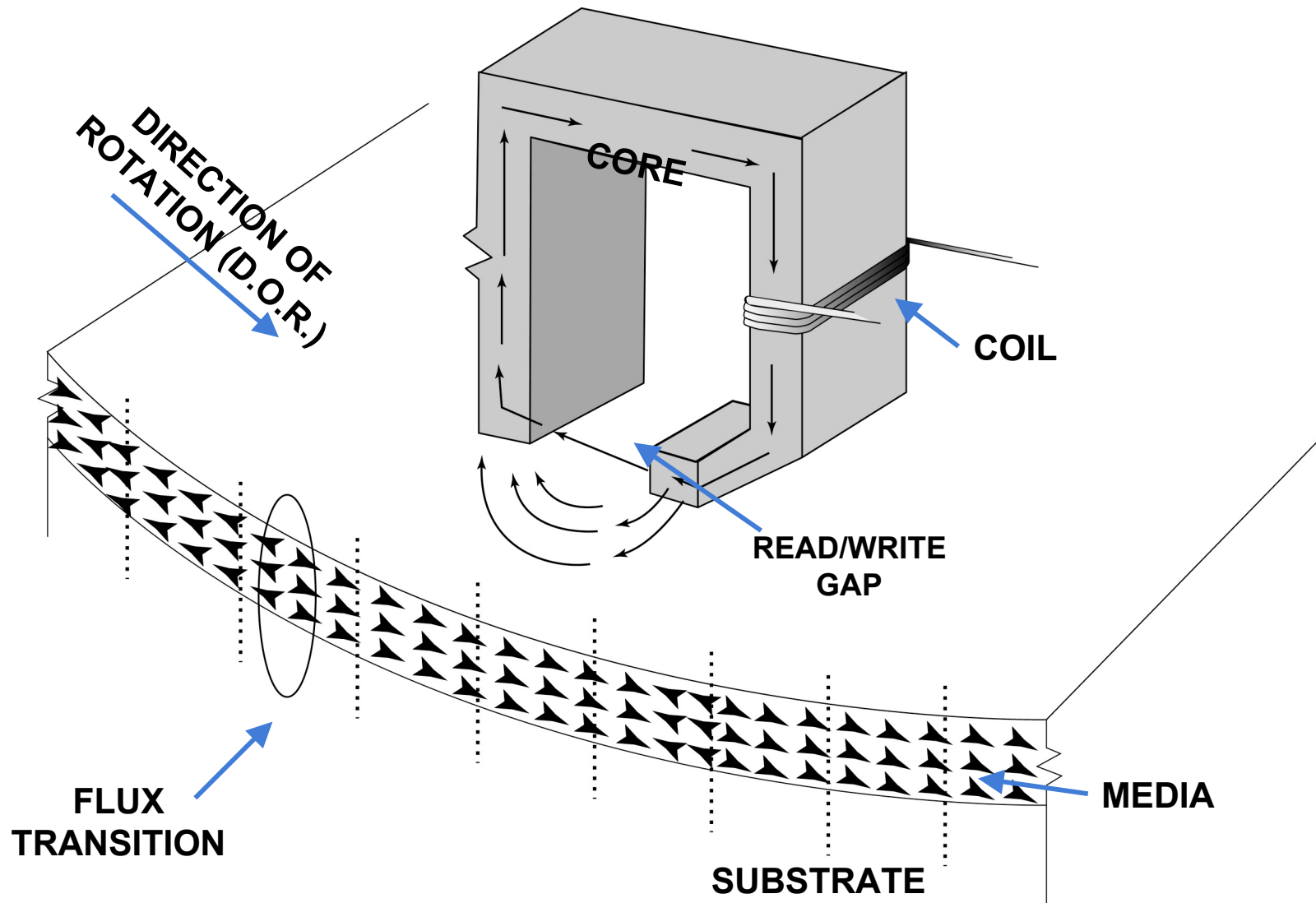


Disk Drive – Magnetic Recording

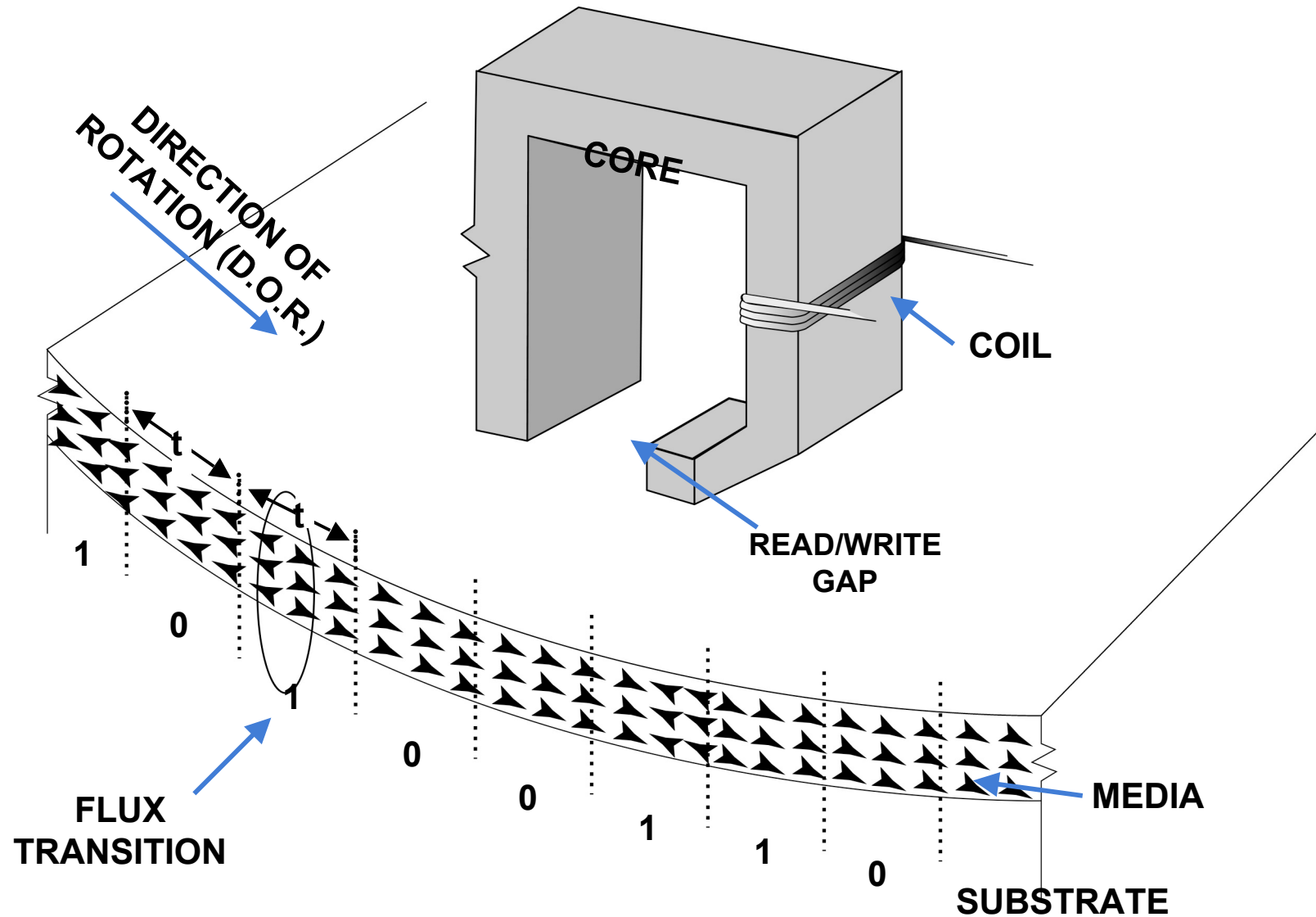
- How does a disk's magnetic recording work?
 - this will be only a high-level overview
- Important for several reasons
 - gives some insight into external properties of disks
 - wastes some of the potential capacity
 - will (eventually) impose some fundamental limits*
 - you might work for a disk company some day

*note that such limits have been talked about for many years, and have always been overcome so far

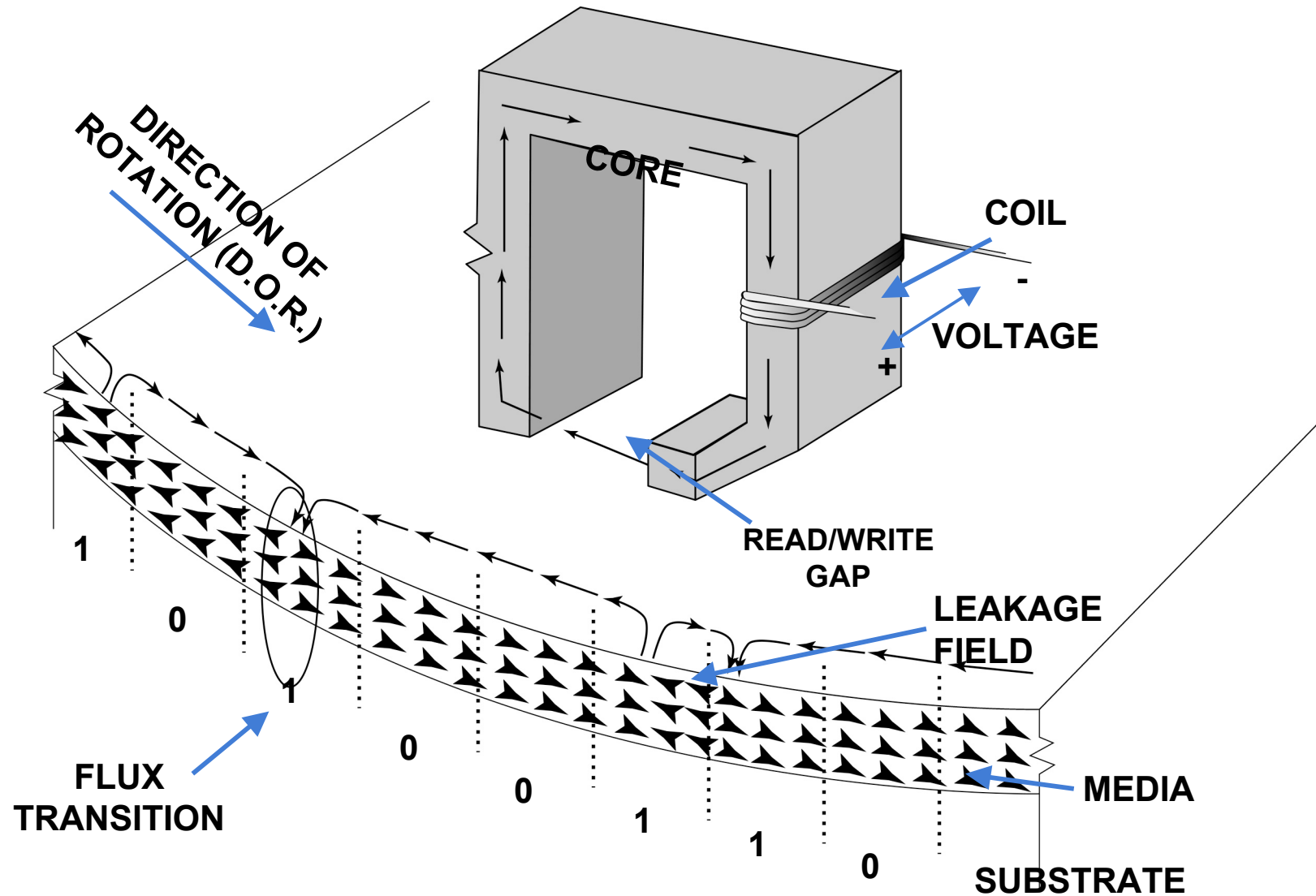
The magnetic recording process



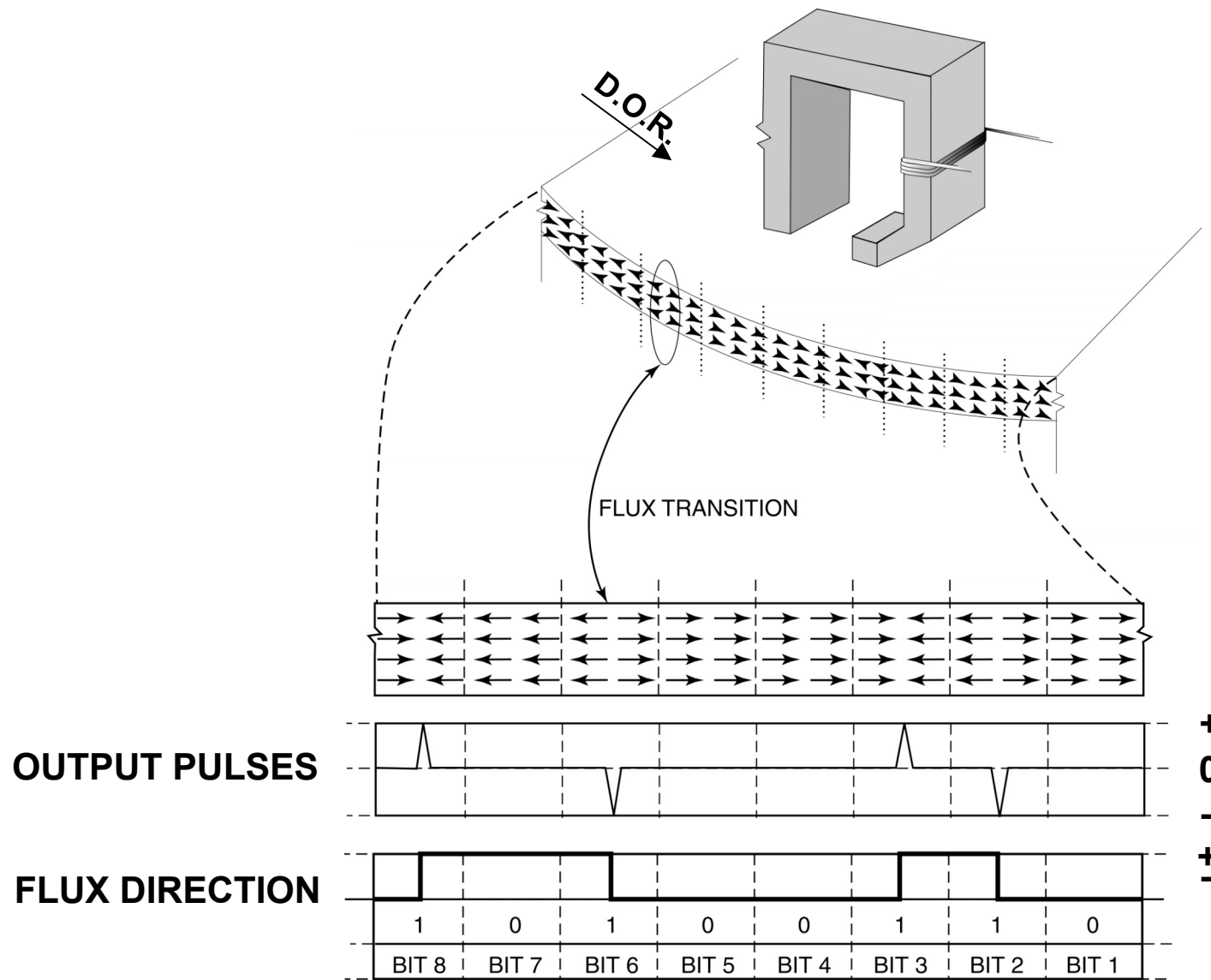
Using magnetic recording to store data



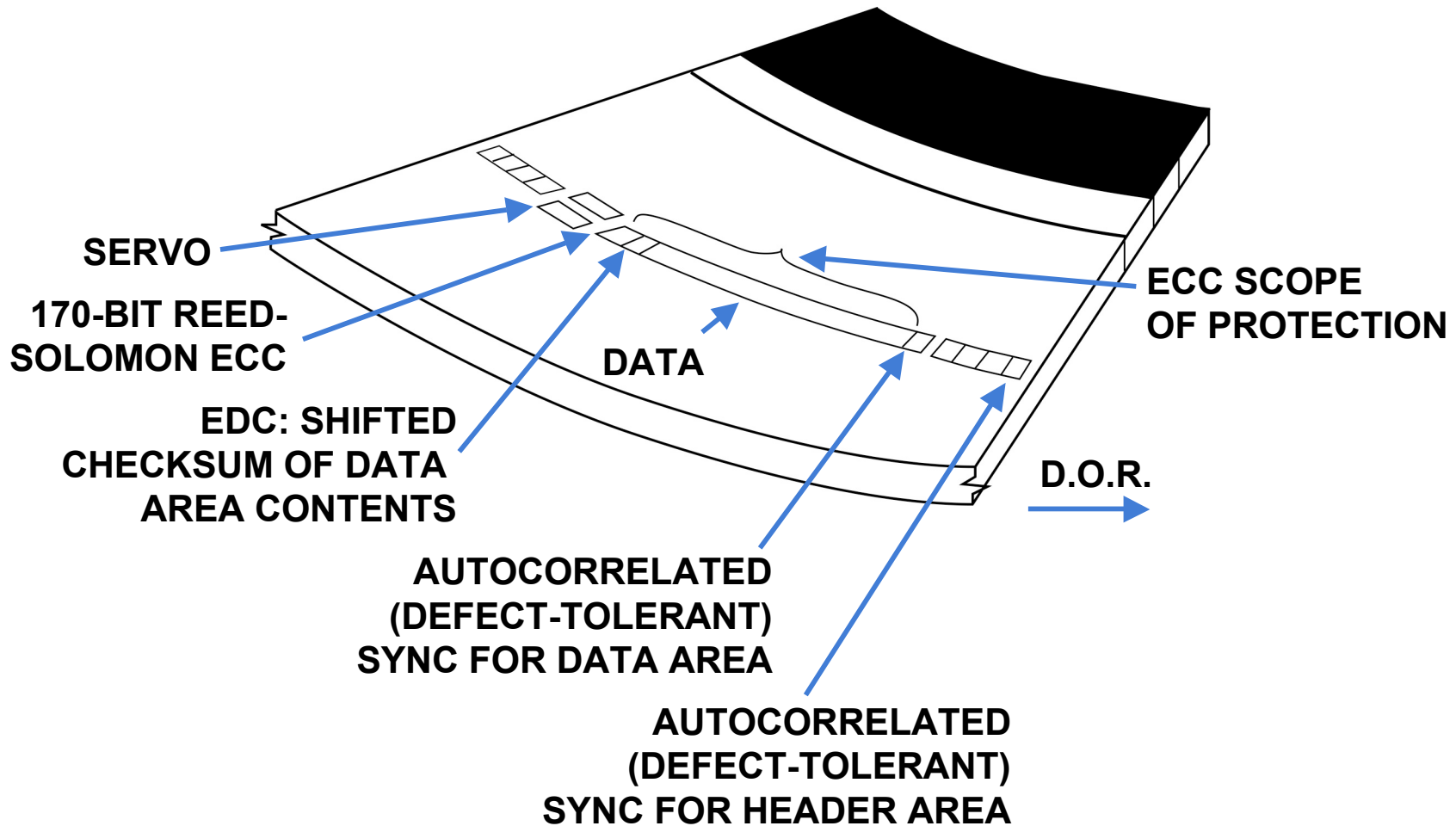
Recovering magnetically recorded data



Recording and its output (simplified)

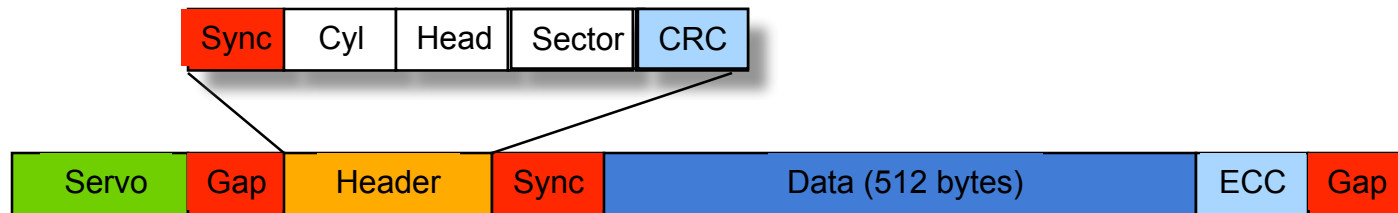


Disk Drive – sector format



Disk Drive – detailed sector format

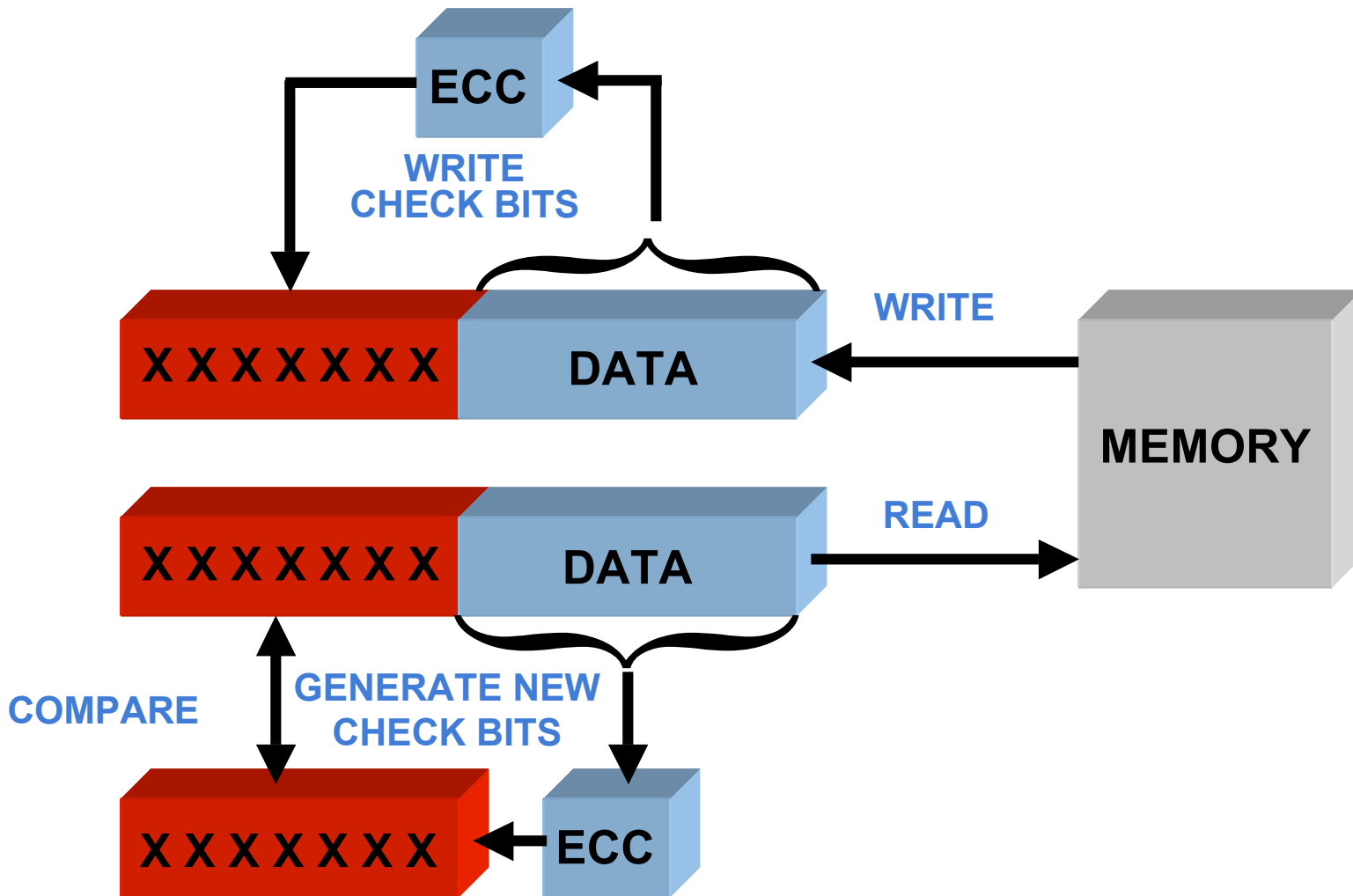
- Addressable unit is a sector



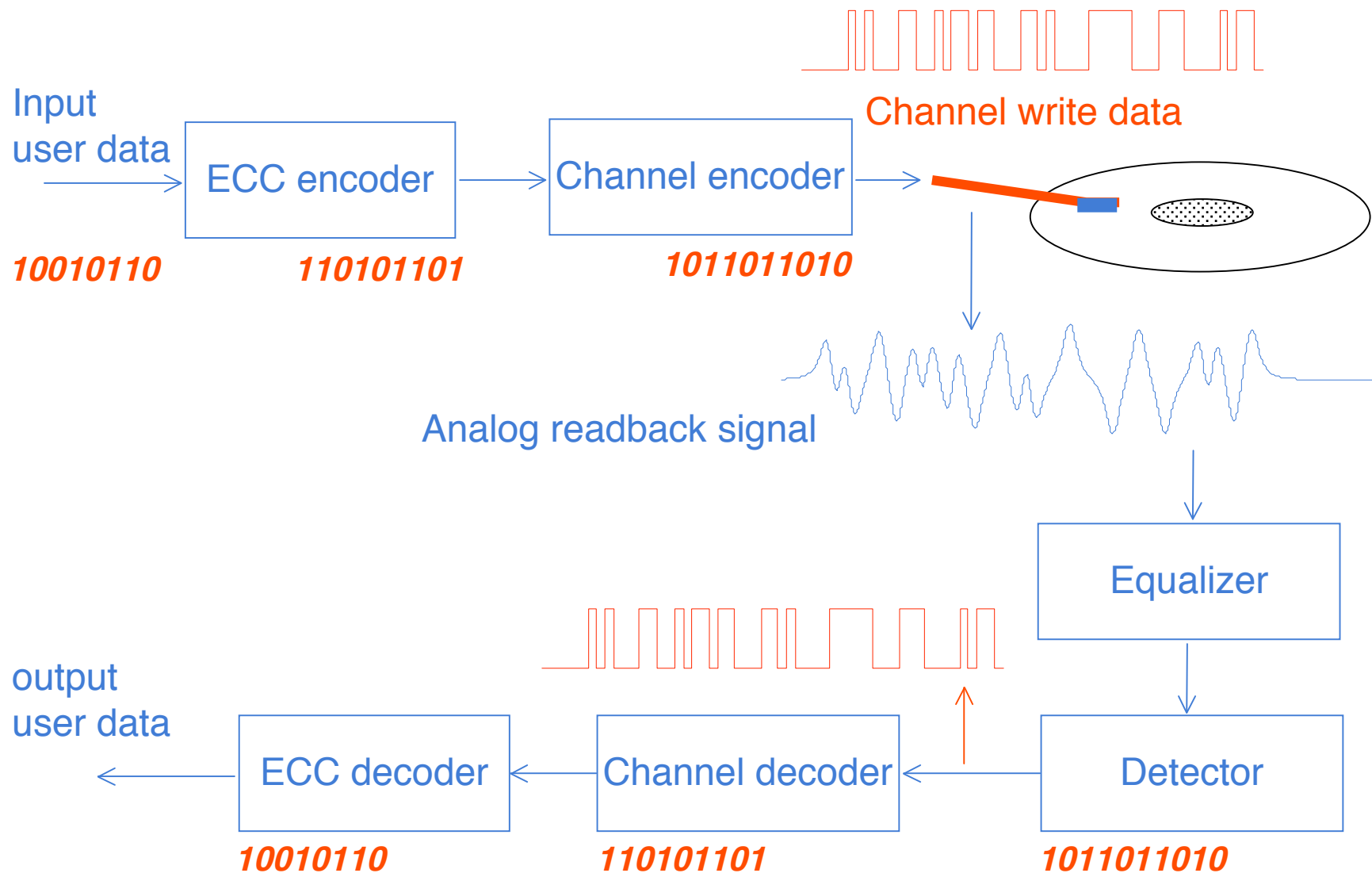
- Sector breaks down into several different fields
 - Typical data portion size - 512 bytes*
 - Typical format
 - sync followed by address field (cyl, head, sector, crc)
 - crc used to verify cyl, head, sector info
 - gap followed by the data
 - ecc over the data
 - verify data and correct bit errors
 - header, ECC and gaps typically use between 40 and 100 bytes

*520 and 528 also possible, larger Real Soon Now

Disk Drive – error-correction code (ECC)



Recording Channel Data Flow



Encoding and ECC

- Why encoding?
 - Signal processing channels can only detect changes so rapidly
 - so, only so many 1s in a row
 - Timing can only stay in sync for so long
 - so, only so many 0s in a row
- Why ECC?
 - At such high densities, problems occur frequently
 - ECC detects and can allow on-the-fly correction
 - Important consequence
 - when writing a sector, one ends up with one of three states
 - all written
 - all not written
 - sector destroyed
 - NEVER just partially modified (and able to pass ECC check)

Disk Access Time Components

Moving the bits on and off

Response Time for Disks

- Response time: user-visible service time
 - Queue time + Access time
- Access time: service time for a disk access
 - Command + Seek + Rotation + Transfer
 - we'll focus on the last three today

Seek Time

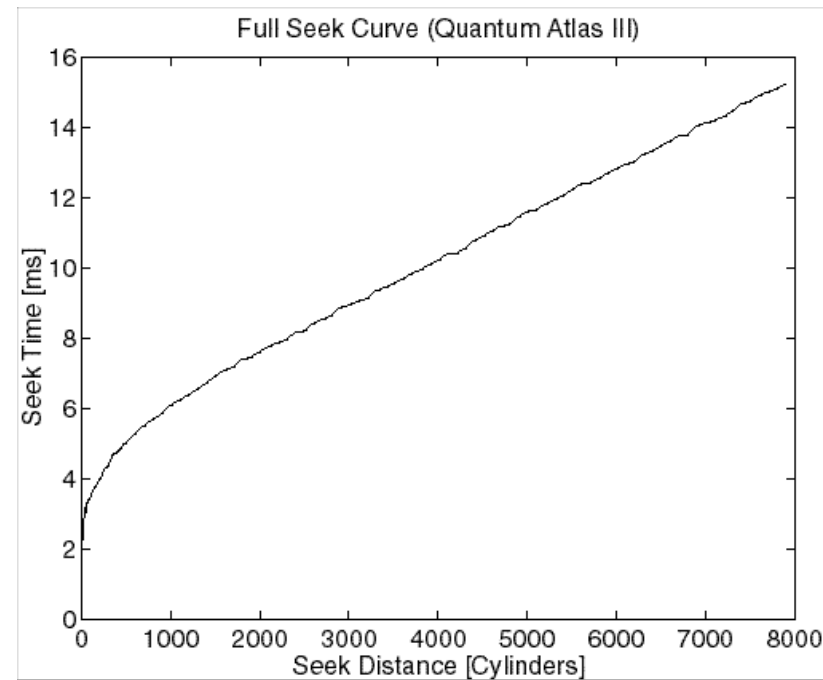
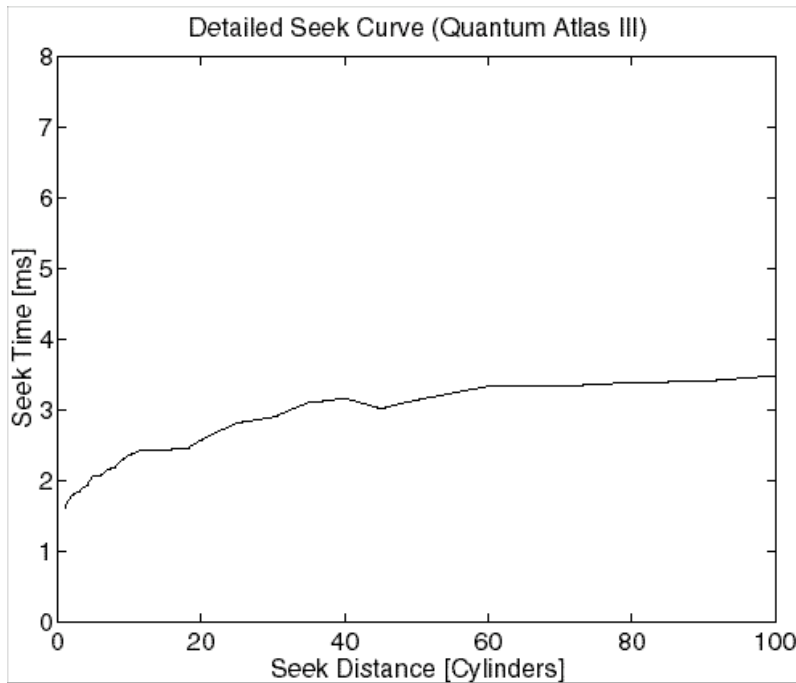
- Time required to move head to desired track
- A seek has up to four components
 - accelerate
 - coast at max velocity
 - only if going far enough to reach max velocity
 - decelerate
 - settle onto correct track
 - even required for switching tracks
 - remember thermal expansion and runout?
 - takes extra time to settle before writing
 - need to make extra certain to avoid destroying adjacent data
 - reads, on the other hand, can take chances

 - short seeks today just rely on track settling mechanism

“Average Seek Time”

- Watch out for misrepresentations
 - purposeful or accidental
- What it is
 - depends on workload
 - for random workloads, average of seeks for all possible reqs
- What it is not
 - seek time for average of possible distances
 - this would be $0.5 * \text{number of cylinders}$
 - seek time for distance from any LBN to any other
 - avg. distance for random workloads is about $1/3$ of total

A Real Seek Profile



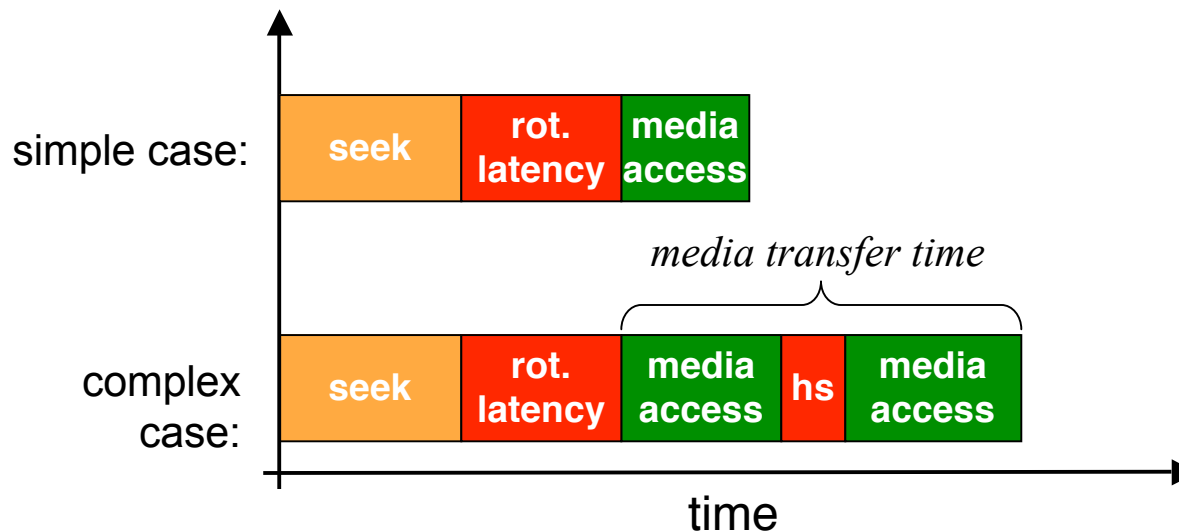
Note that it isn't linear

Rotational Latency

- Time required for first desired sector to reach head
- Depends on rotation speed
 - measured in Rotations Per Minute (RPMs)
- Computing average rotational latency
 - for almost all workloads, we can safely assume that there is an equal likelihood of landing on any sector of the track
 - this gives equal probability of each rotational latency
 - from 0 sectors to N-1 sectors
 - thus, average rotational latency is time for 1/2 revolution
 - e.g., for 7200 RPM
 - one rotation = $60\text{s} / 7200 = 8.33\text{ ms}$
 - average rotational latency = 4.16 ms

Media Transfer Time

- Time for needed sectors to rotate under head
- Computing transfer time
 - simple case: all sectors on one track
 - sectors desired * time for revolution / sectors per track
 - more complex case: spread across two or more tracks
 - add a head switch time, as needed



Disk Drive Typical Characteristics

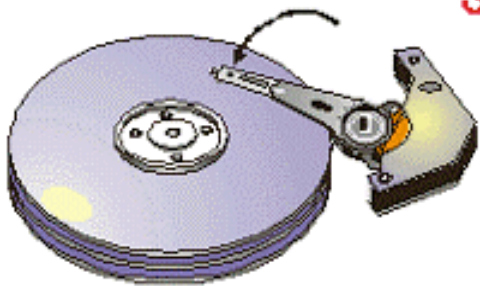
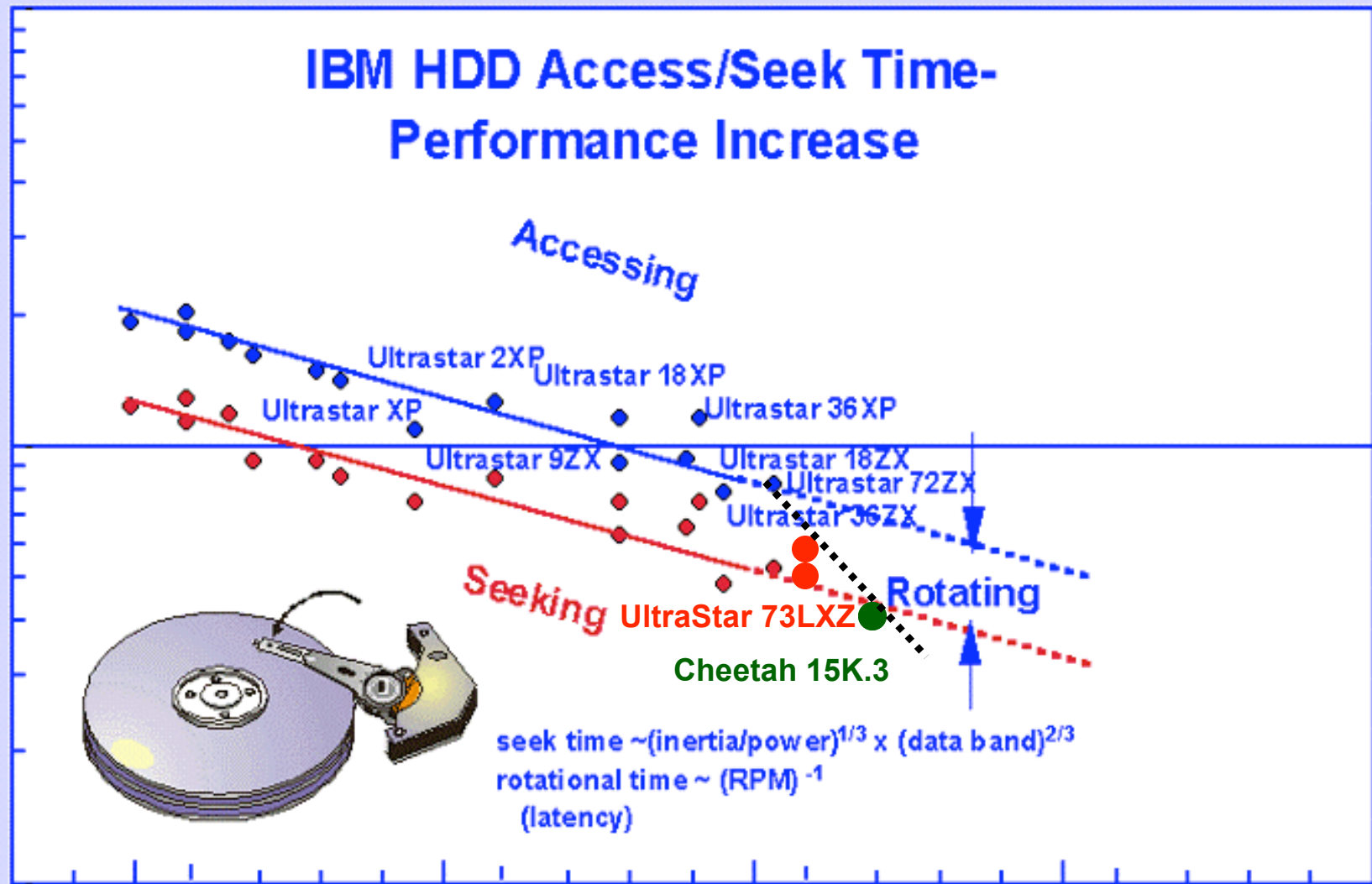
- Seek times: 0.6 – 7 ms, depends on distance
 - average 3 – 4 ms
 - improving at 7-10% per year
- Head switch time: 0.6 ms
- Rotation speeds: 15,000 RPM
 - average latency of 2 ms
 - improving at 7-10% per year
- Data rates: 50-75 MB/s*, depending on zone
 - avg sector xfer time of 25 us
 - improving at 40+% per year

Numbers for
Seagate
Cheetah 15K.3
(late 2002)

*note these are external data rates, internal are higher
(without all the ECC, sector gaps, etc.)

IBM HDD Access/Seek Time-Performance Increase

Time, milliseconds



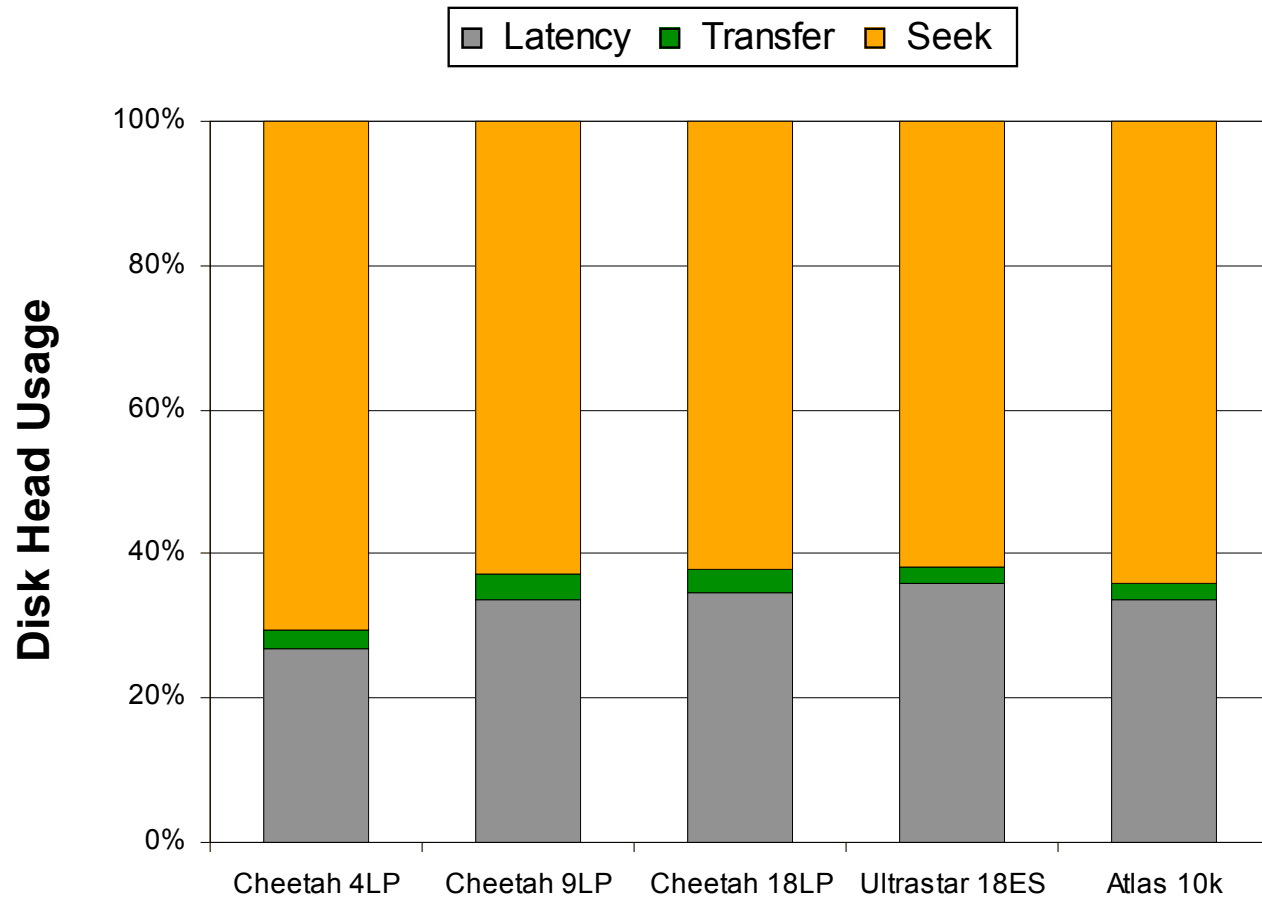
seek time $\sim (\text{inertia/power})^{1/3} \times (\text{data band})^{2/3}$
 rotational time $\sim (\text{RPM})^{-1}$
 (latency)

SEEK1999A.PRZ



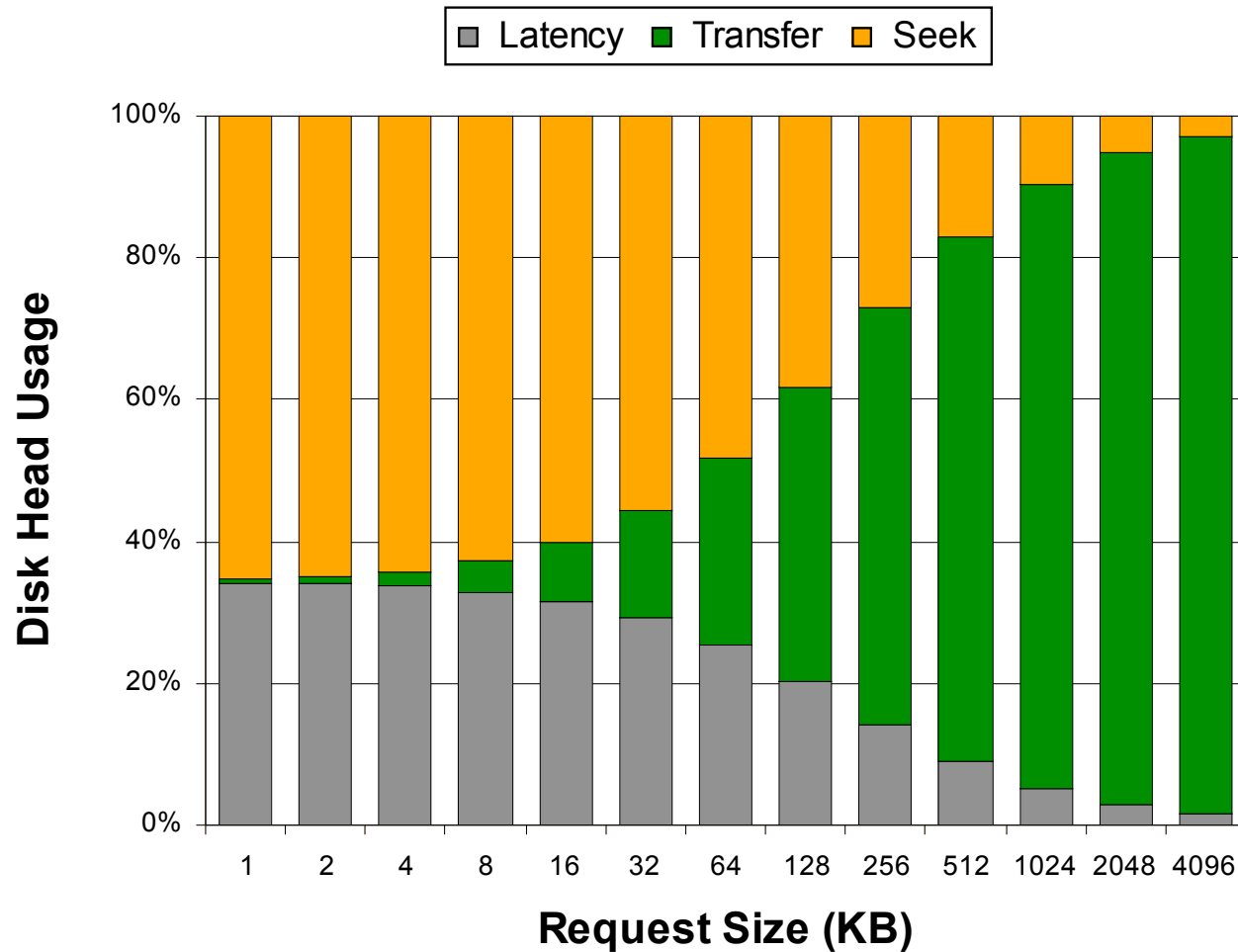
Ed Grochowski at Almaden

Where Does Disk Head's Time Go?

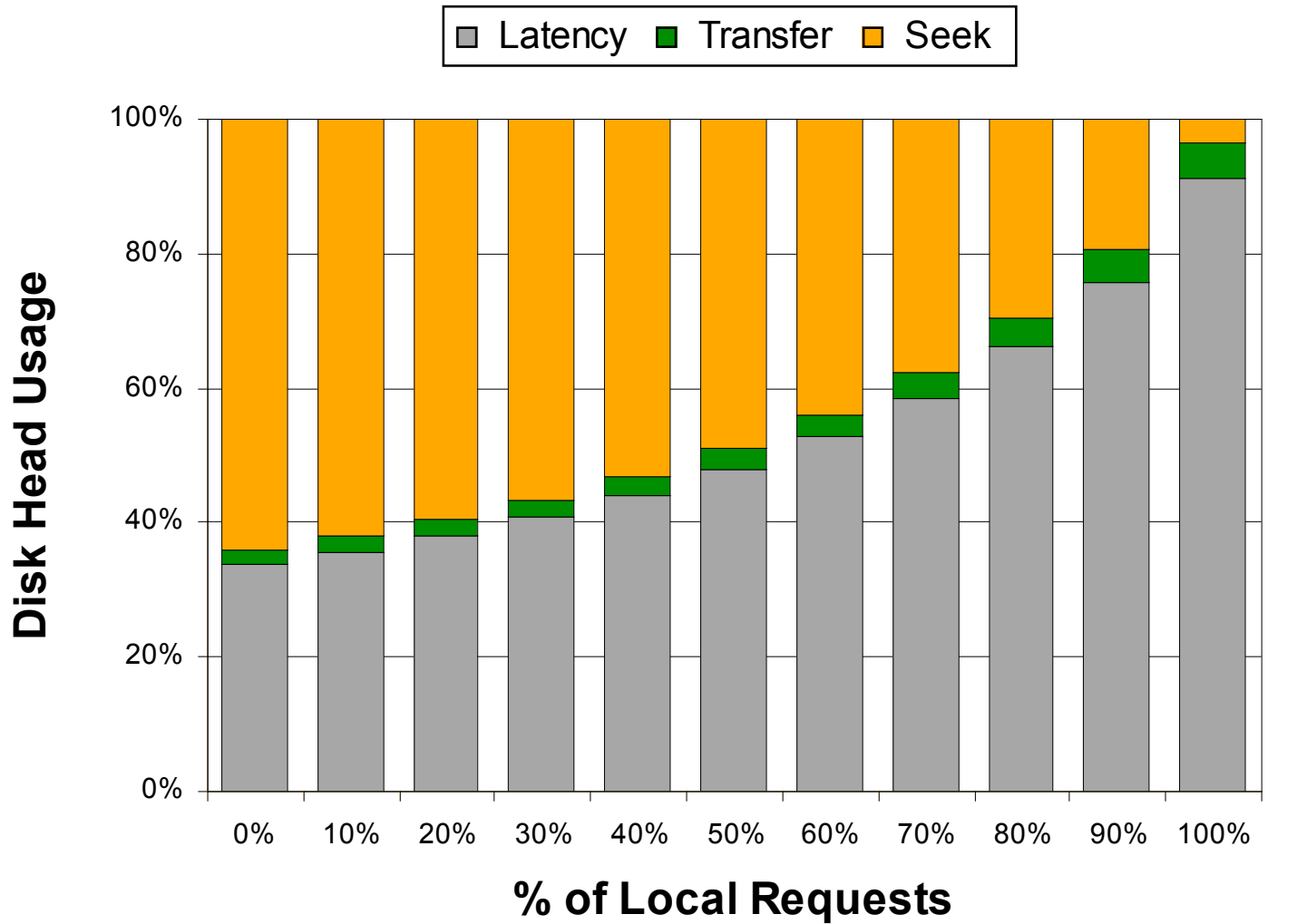


Random 4KB requests

Impact of Request Sizes



Impact of Locality



4KB requests

Microdrive: Compact FLASH form factor

Capacity: 4 gigabytes (January 2003)



Disk Drive Firmware Algorithms



Outline

- Mapping LBNs to physical sectors
 - zones
 - defect management
 - track and cylinder skew
- Bus and buffer management
 - optimizing storage subsystem resources
- Advanced buffer space usage
 - prefetching and caching
 - read/write-on-arrival

How functionality is implemented

- Some of it is in ASIC logic
 - error detection and correction
 - signal/servo processing
 - motor/seek control
 - cache hits (often)
- Some of it is in firmware running on control processor
 - request processing
 - request queueing and scheduling
 - LBN-to-PBN mapping
- Key considerations: cost and performance and cost
 - optimize common cases
 - keep things simple and space-conscious

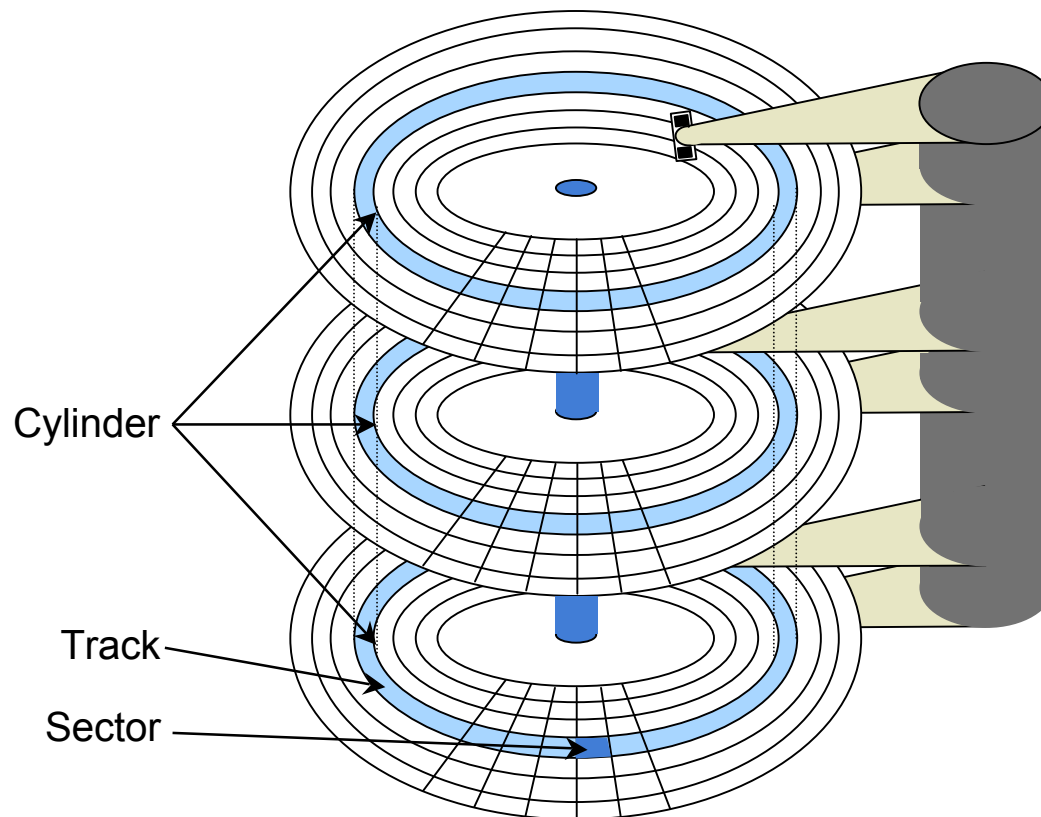
Recall the storage device interface

- Linear address space of equal-sized blocks
 - each identified by logical block number (LBN)

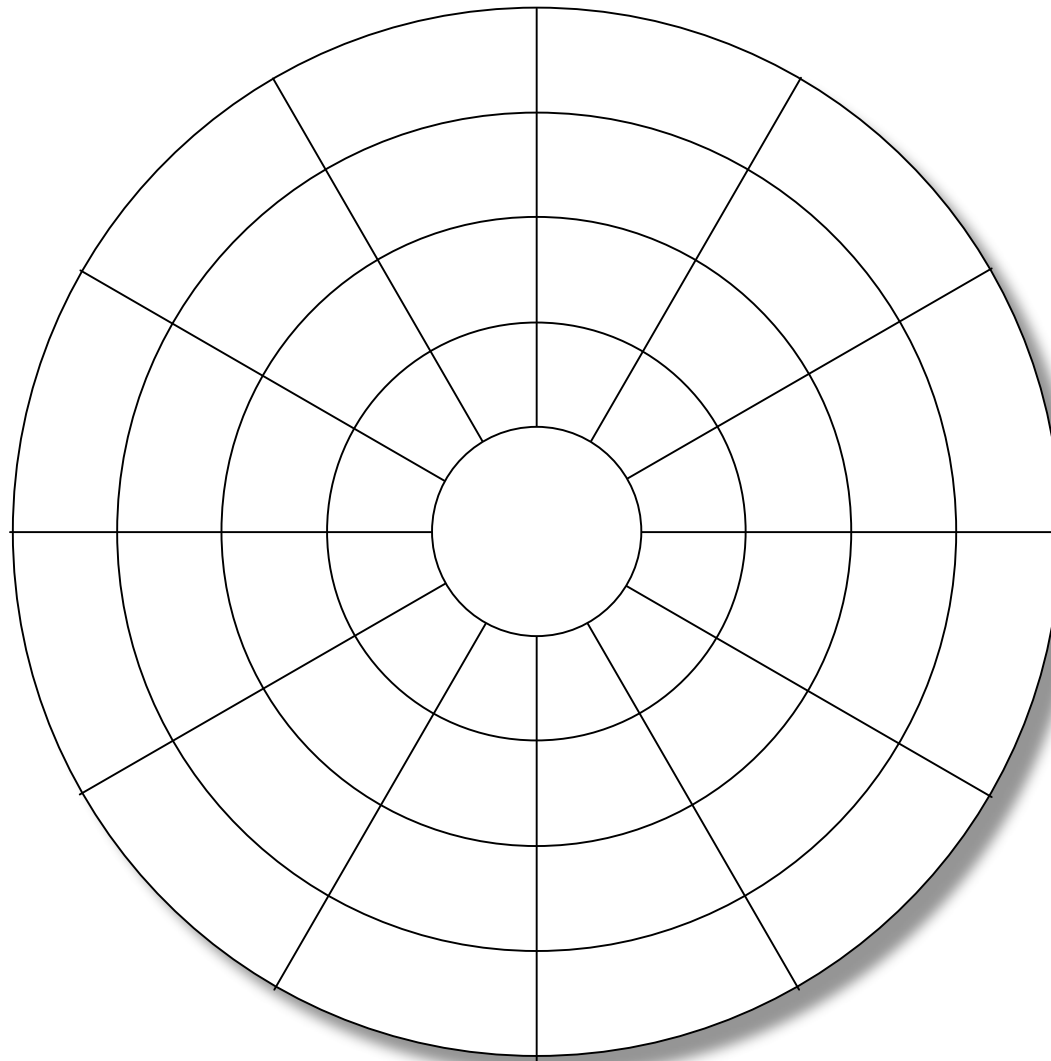


- Common block size: 512 bytes
- Number of blocks: device capacity / block size

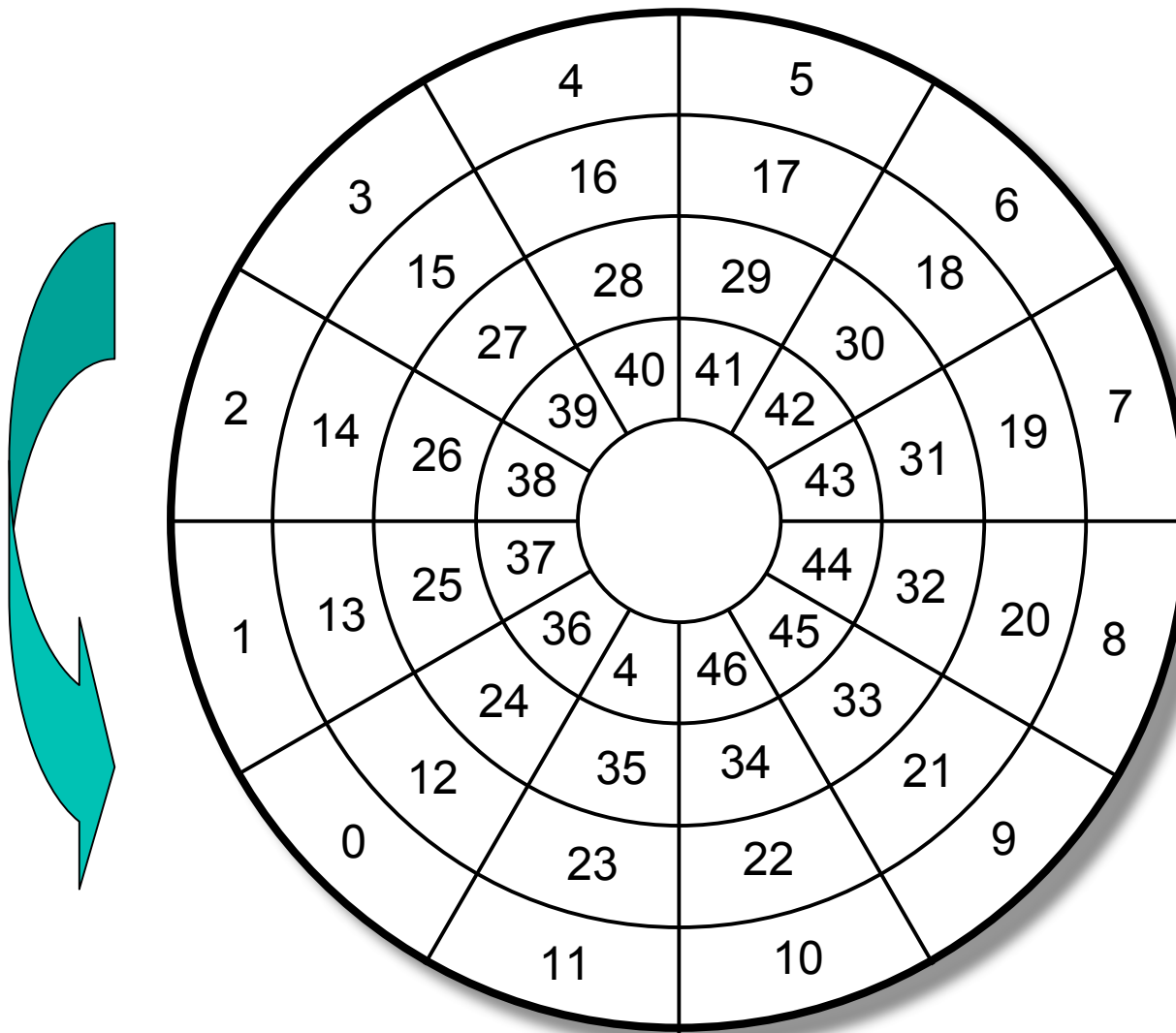
Recall the physical disk storage reality



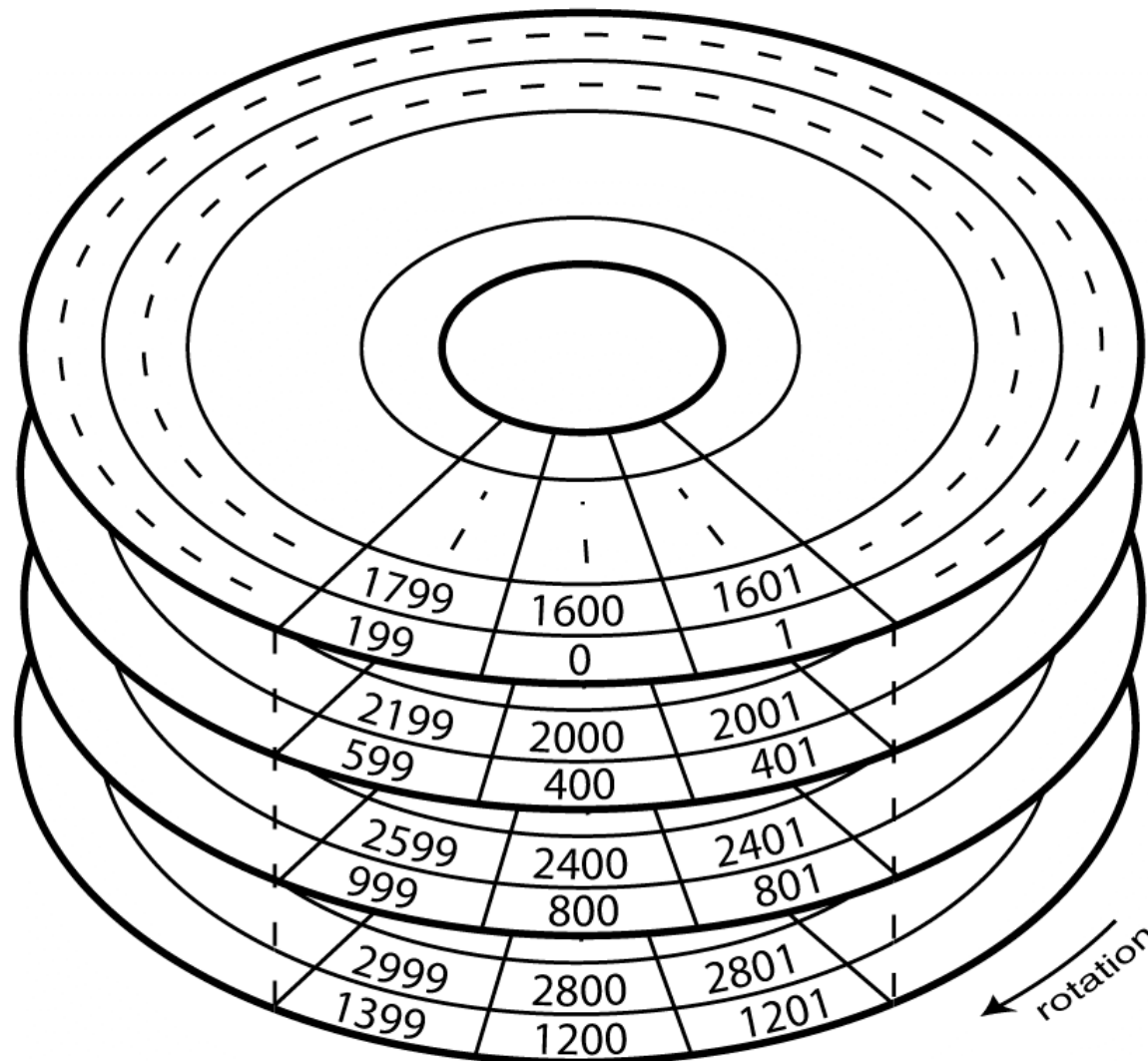
Physical sectors of a single-surface disk



LBN-to-physical for a single-surface disk



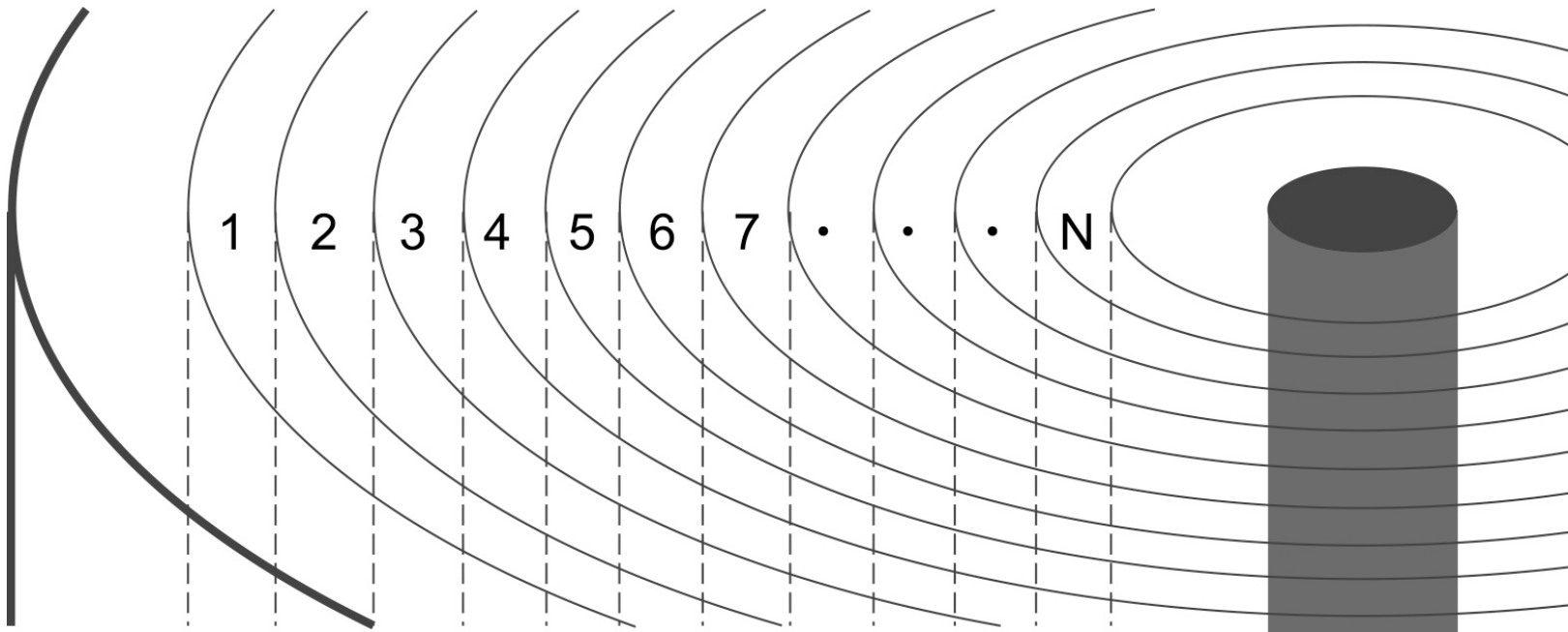
Extending mapping to a multi-surface disk



Some real numbers for modern disks

- # of platters: 1-4
 - 2-8 surfaces for data
- # of tracks per surface: 10s of 1000s
 - same thing as # of cylinders
- # sectors per track: 500-900
 - so, 250-450KB
- # of bytes per sector: usually 512
 - can be chosen by OS for some disks
 - disk manufactures want to make it bigger

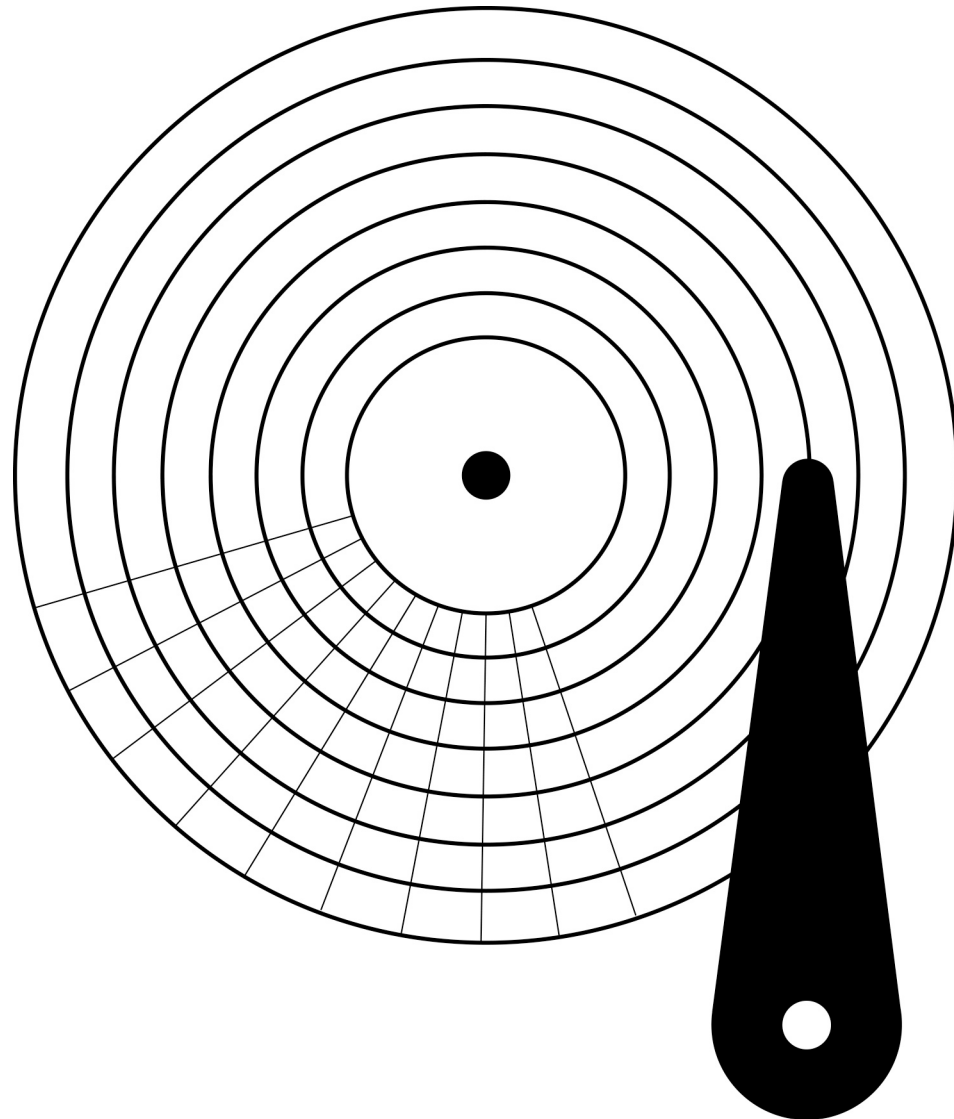
Clarification of Cylinder Numbering



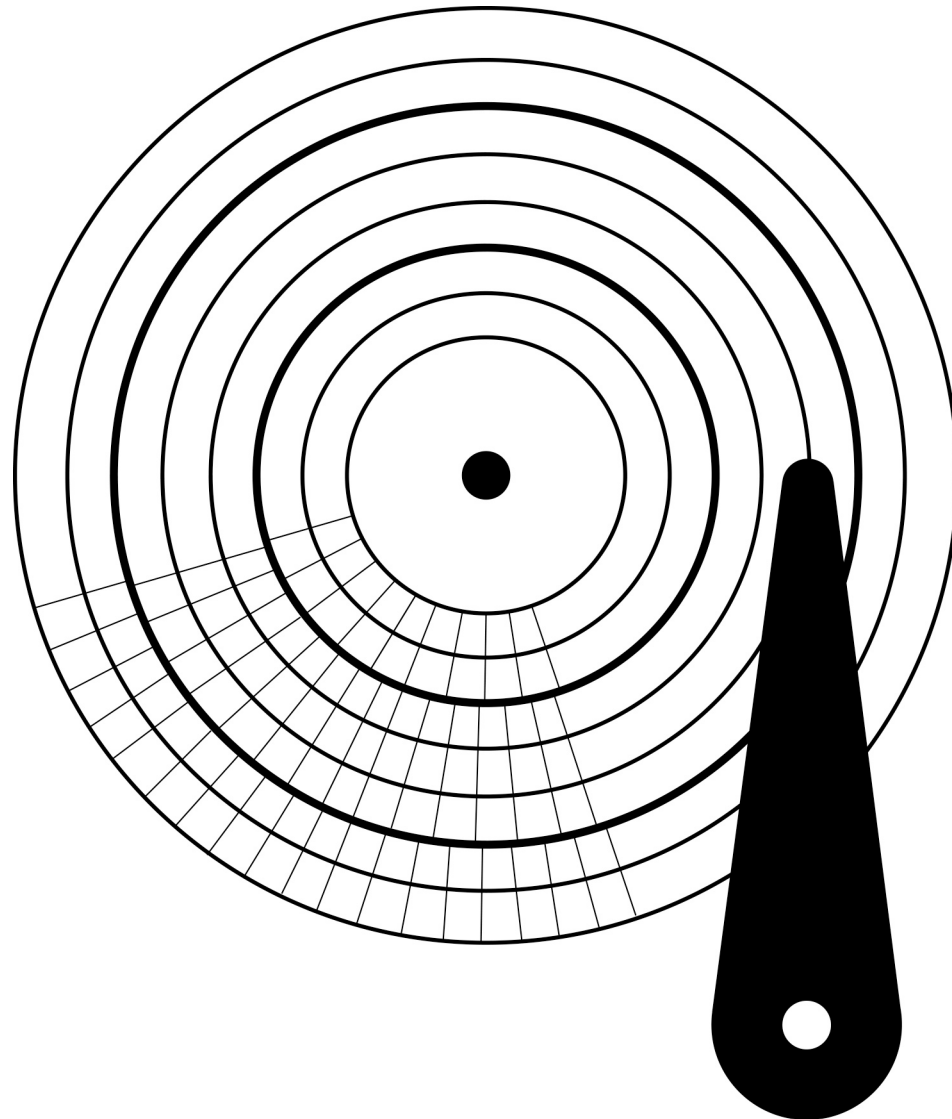
First Complication: Zones

- Outer tracks are longer than inner ones
 - so, they can hold more data
 - benefits: increased capacity and higher bandwidth
- **Issues**
 - increased bookkeeping for LBN-to-physical mapping
 - more complex signal processing logic
 - because of variable bit rate timing
- **Compromise: zones**
 - all tracks in each zone hold same number of sectors

Constant number of sectors per track



Multiple “zones”



A real zone breakdown

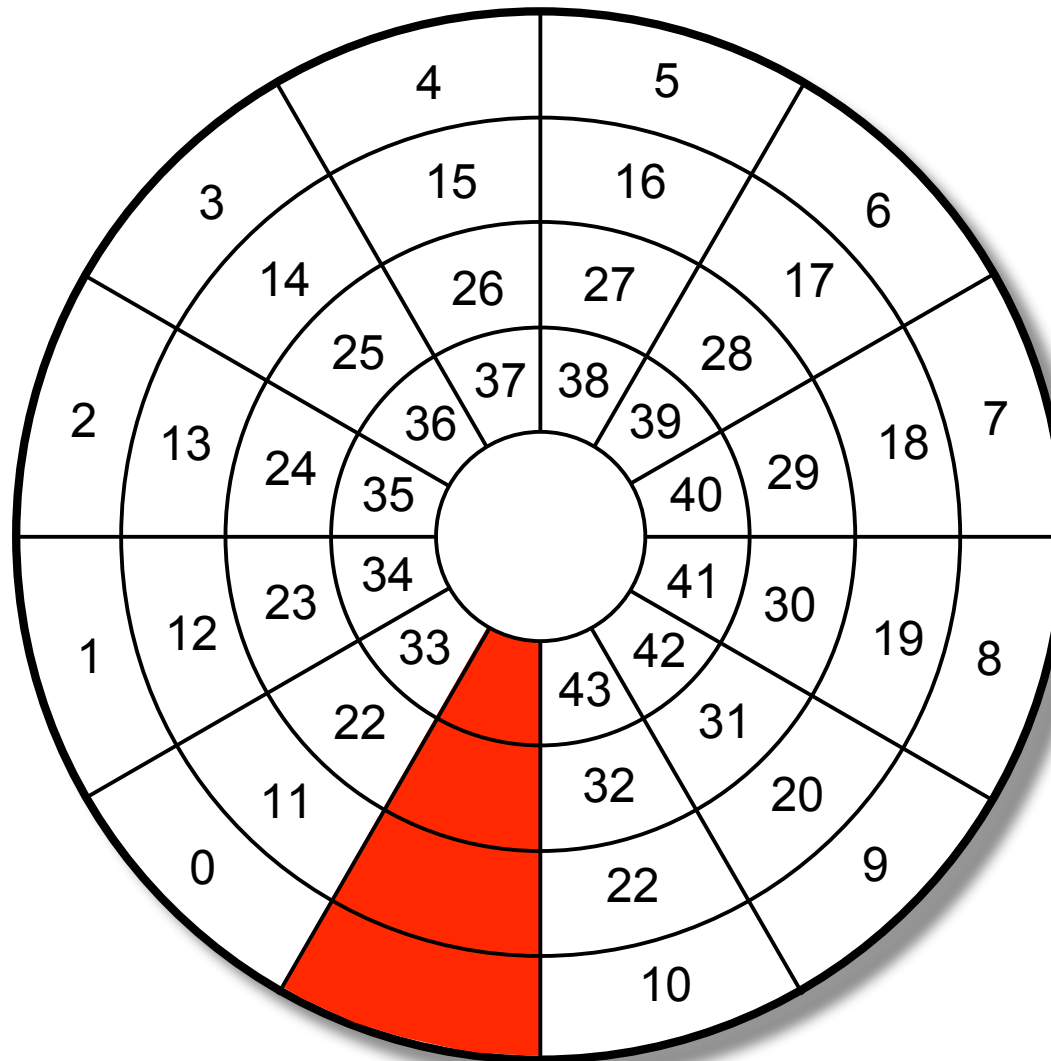
- IBM Ultrastar 18ES (1998)

Zone	Start cylinder	End cylinder	SPT
0	0	377	390
1	378	1263	374
2	1264	2247	364
3	2248	3466	351
4	3466	4504	338
5	4505	5526	325
6	5527	7044	312
7	7045	8761	286
8	8762	9815	273
9	9816	10682	260
10	10683	11473	247

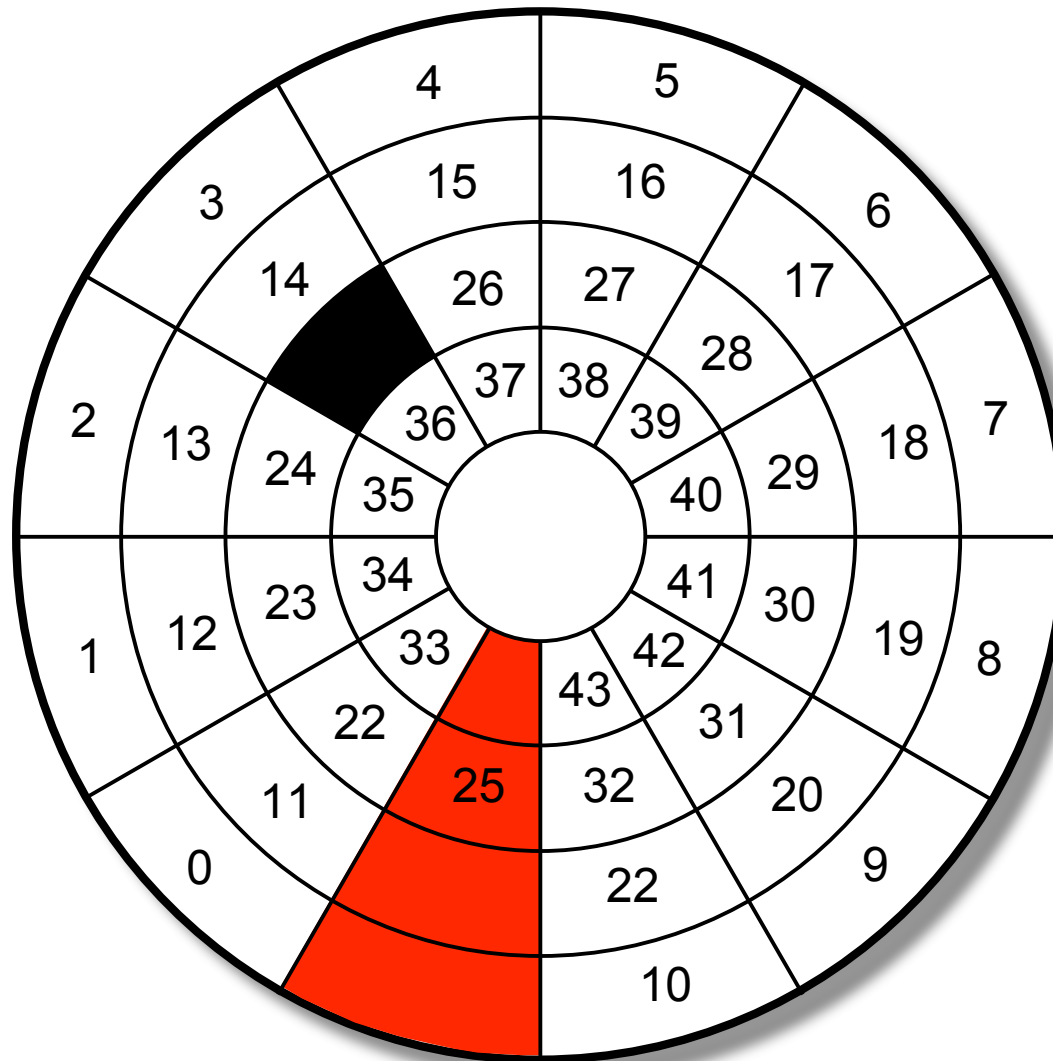
Second Complication: Defects

- Portions of the media can become unusable
 - both before installation and during use
 - former is MUCH more common than latter
- Need to set aside physical space as spares
 - simplicity dictates having no holes in LBN space
 - many different organizations of spare space
 - e.g., sectors per track, cylinder, group of cylinders, zone
- Two schemes for using spare space to handle defects
 - remapping
 - leave everything else alone and just remap the disturbed LBNs
 - slipping
 - change mapping to skip over defective regions

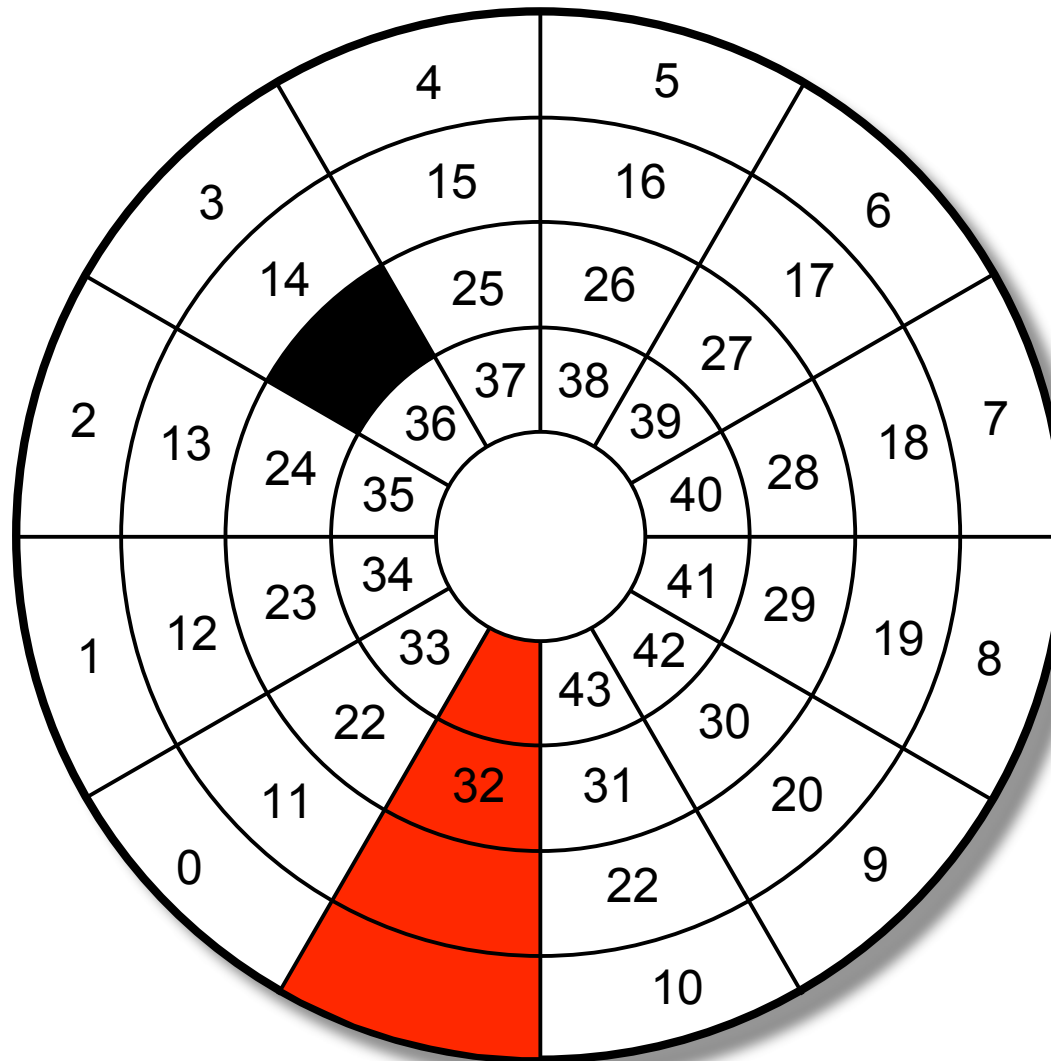
One spare sector per track



Remapping from defective sector to spare



LBN mapping slipped past defective sector



Some Real Defect Management Schemes

- High level facts
 - percentage of space: $< 1\%$
 - always slip if possible
 - much more efficient for streaming data
- One real scheme: Seagate Cheetah 4LP
 - 108 spare sectors every 12 cylinders
 - located on the last track of the 12-cylinder group
 - used only for remapped sectors grown during usage
 - many spare sectors on innermost cylinders
 - used to provide backstop for all slipped sectors

Computing physical location from LBN

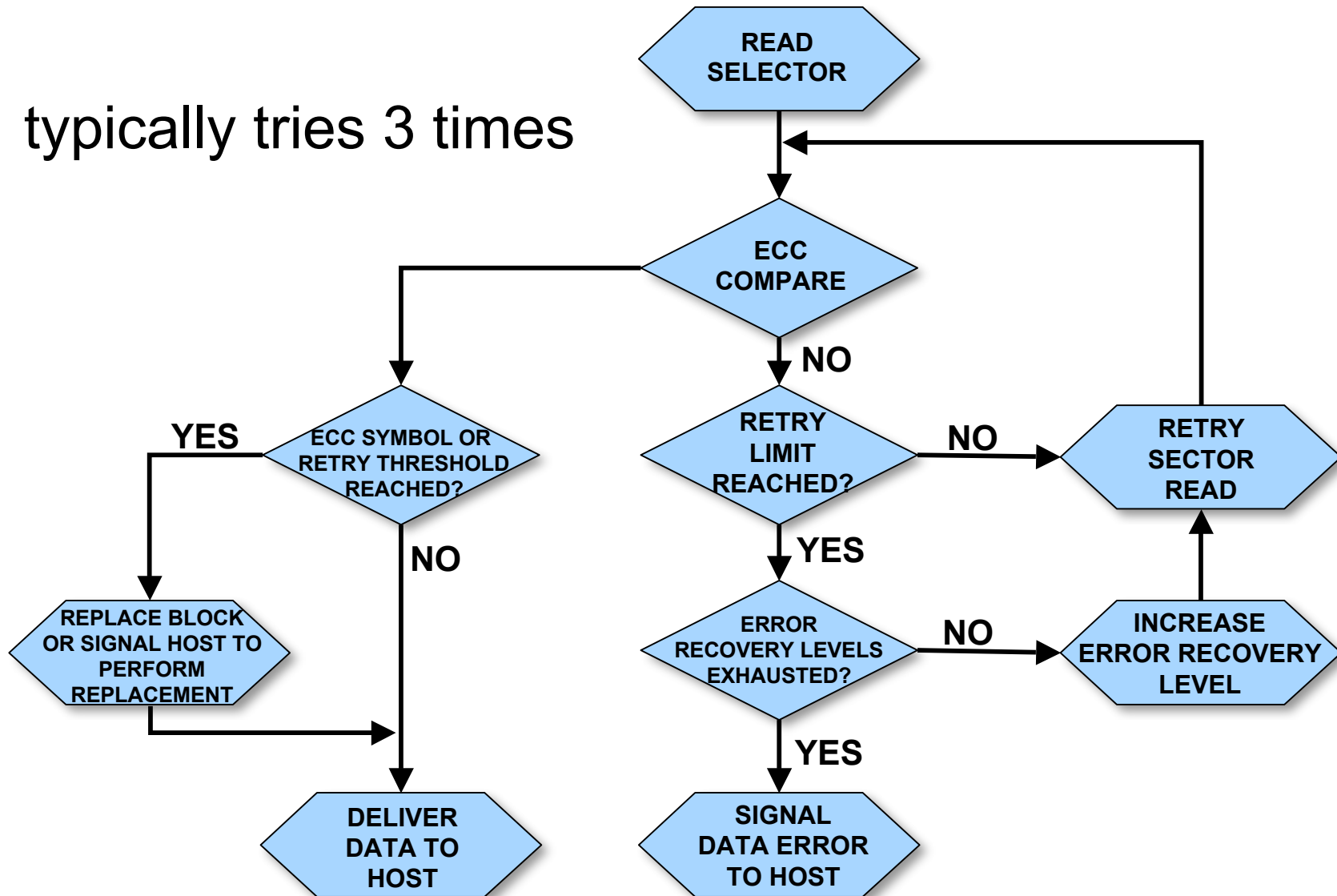
- First, check list of remapped LBNs
 - usually identifies exact physical location of replacement
- If no match, do the steps from before
 - but, also account for slipped sectors that affect desired LBN
- About 10 different management schemes
 - For any given scheme, the computations can be fairly straightforward. However, it is quite complex to discuss them all at once concretely

When defects “grow” during operation

- First, try ECC
 - it can recover from many problems
- Next, try to read the sector again
 - often, failure to read the sector is transient
 - cost is a full rotation added to access time
- Last resort, report failure and remap sector
 - this means that the stored data has been lost
 - until next write to this LBN, reads get error response
 - new data allows the location change to take effect

Error Recovery Algorithm for READs

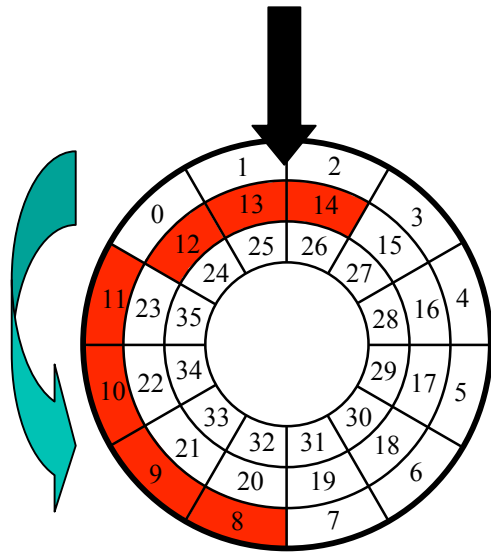
typically tries 3 times



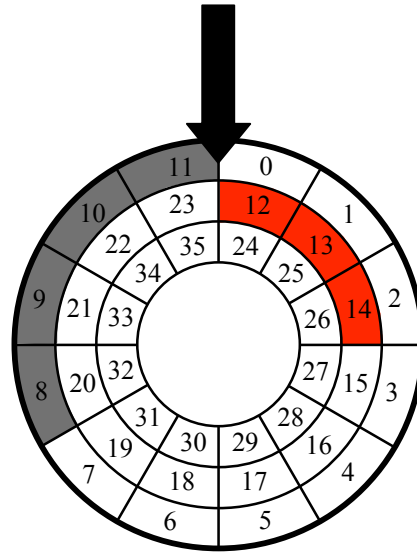
Third Complication: Skew

- Switching from one track to another takes time
 - sequential transfers would suffer full rotation
- Solution: skew
 - offset physical location of first sector to avoid extra rotation
 - selection of skew value made from switch time statistics
- Track skew
 - for when switching to next surface within a cylinder
- Cylinder skew
 - for when switching from last surface of one cylinder to first surface of next cylinder

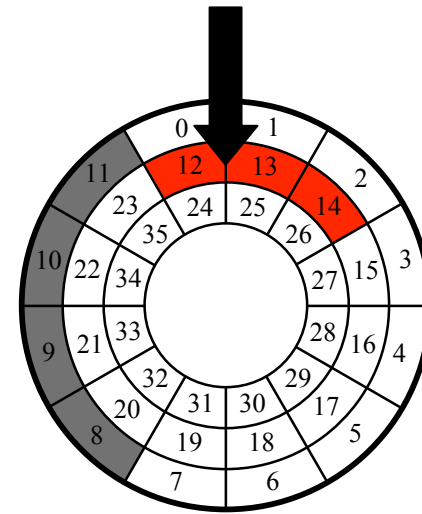
What happens to requests that span tracks?



Request spans 2 tracks

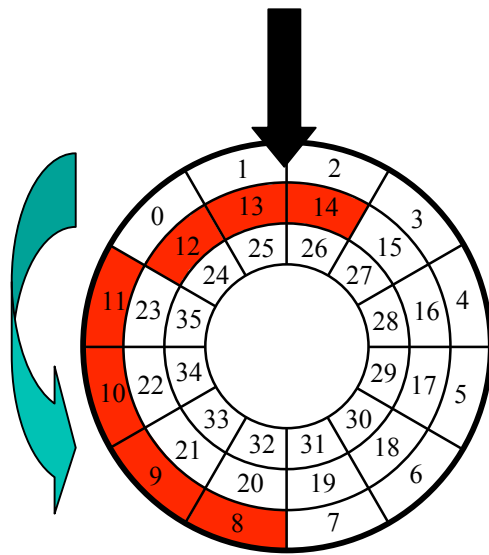


After reading first part

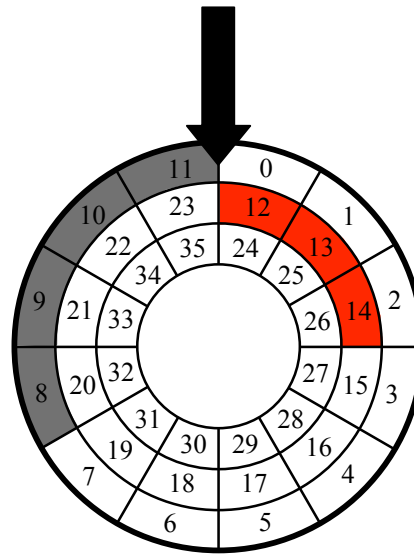


After track switch

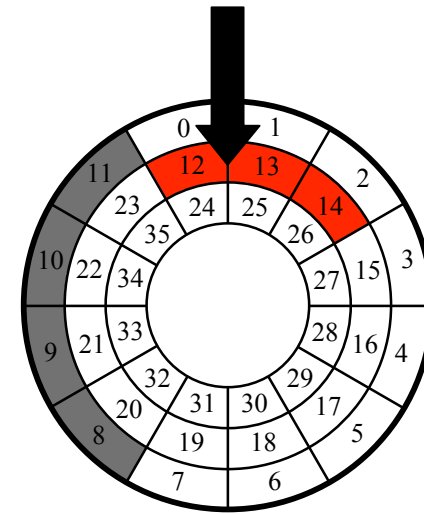
What happens to requests that span tracks?



Request spans 2 tracks



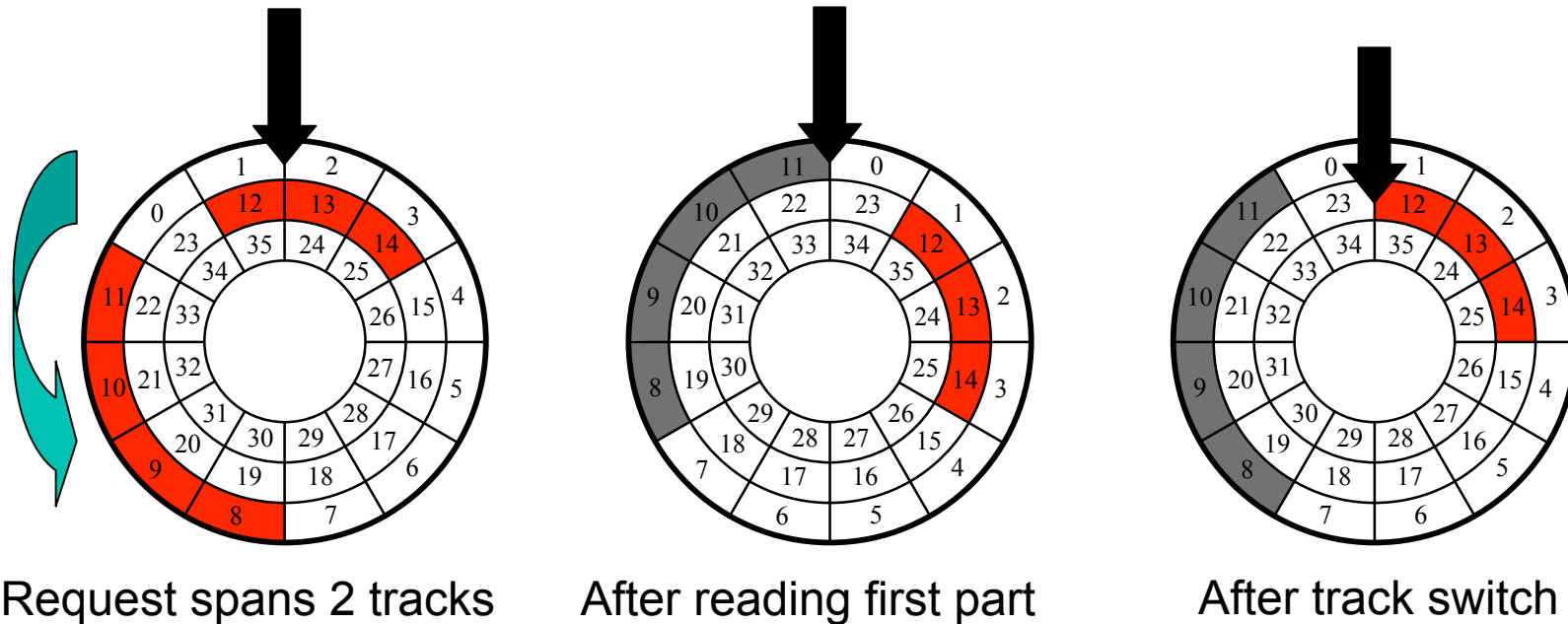
After reading first part



After track switch

Sector 12 rotates past during track switch, so full rotation needed

Same request with track skew of one sector






Track skew prevents unnecessary rotation

Examples of Track and Cylinder Skews

	Quantum Atlas 10k			IBM Ultrastar 18ES		
Skew Zone	SPT	Track	Cylinder	SPT	Track	Cylinder
1	334	64	101	390	58	102
2	324	62	98	374	56	97
3	306	56	93	364	55	95

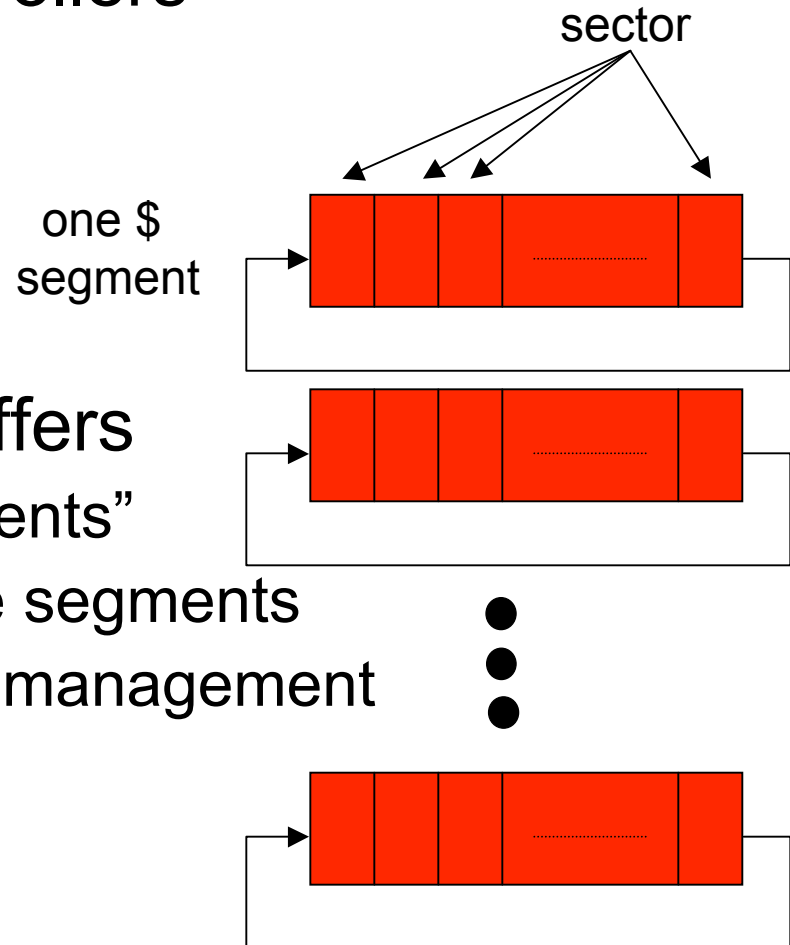
Computing Physical Location from LBN

-  Figure out cylno, surfaceno, and sectno
 - using algorithms indicated previously
-  Compute total skew for first mapped physical sector on this track
 - $\text{totalskew} = (\text{cylno} * \text{cylskew}) + (\text{surfaceno} + (\text{cylno} * (\text{surfaces}-1)) * \text{trackskew})$
-  Compute rotational offset on given track
 - $\text{offset} = (\text{totalskew} + \text{sectno}) \% \text{sectsptrack}$

Basic On-disk Caching

On-disk RAM

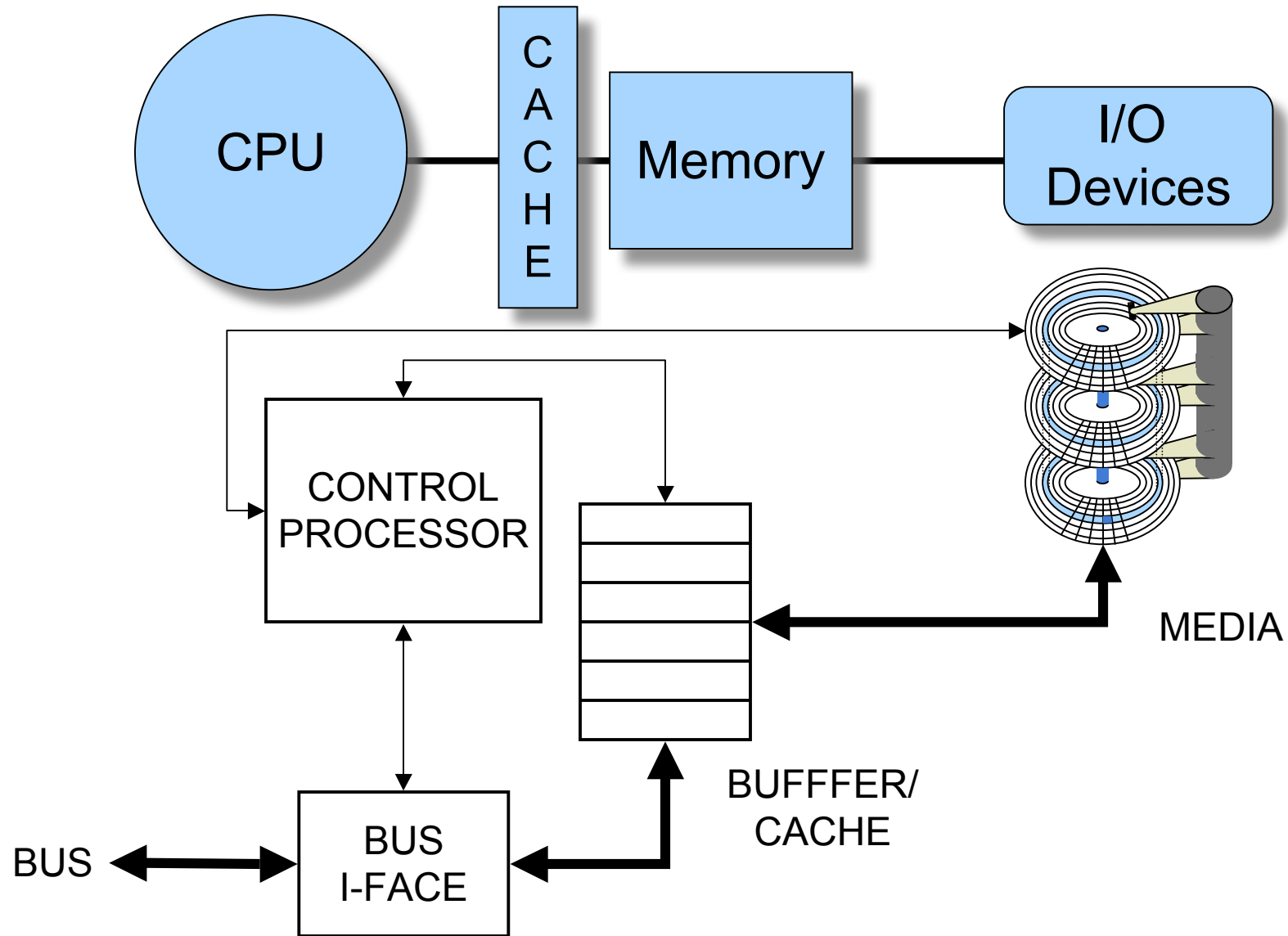
- RAM on disk drive controllers
 - firmware
 - speed matching buffer
 - prefetching buffer
 - cache
- Canonical disk drive buffers
 - several fixed-size “segments”
 - latest thing: variable-size segments
 - down the road: OS style management



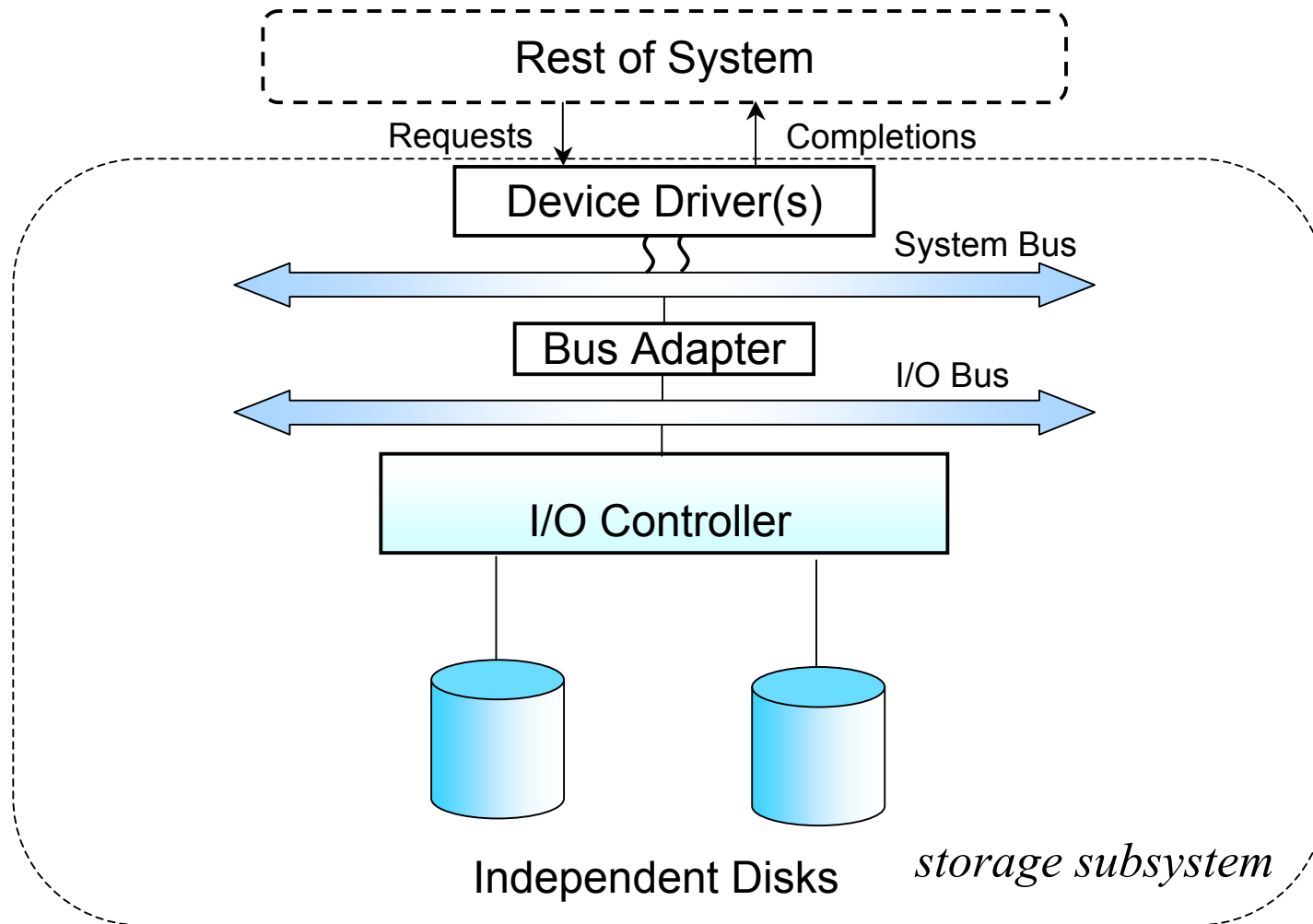
Prefetching and Caching

- Prefetching
 - sequential prefetch essentially free until next request arrives
 - and until track boundary
 - Note: **physically** sequential sectors are prefetched
 - usefulness depends on access patterns
 - Example algorithms
 - prefetch until buffer is full or next request arrives
 - MIN and MAX values for prefetching
 - if track $n-1$ and n have been READ, prefetch track $n+1$
- Caching
 - data in buffer segments retained as cache
 - most of the benefit comes from prefetching

Disk Drive – Complete System?



Not really, recall this...



Mark Kryder's Slides

CTO of Seagate

What's next ...

- Next lecture: 9/19
- Readings will be posted tomorrow