Storage

Ruizhe Pang, Jingqi Sun

Department of Computer Science and Engineering, SUSTech

2025.12.5

Why Study Disk Performance? [4]

- CPU speedup: 40–60% per year
- Disk density growth: 60–80% per year
- Disk performance improvement: 7–10% per year
- **Disk becoming dominant factor in system behavior**

Motivation for Disk Modeling

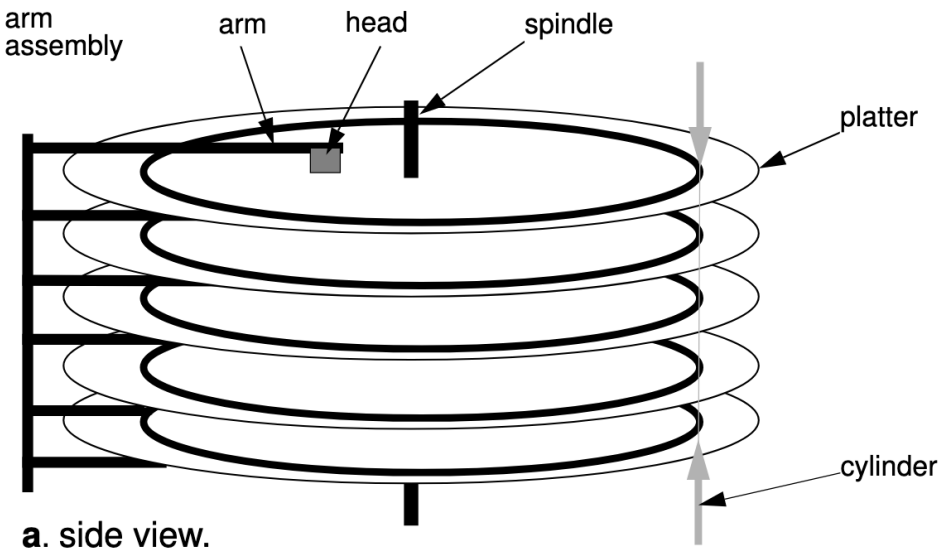
- Improve overall I/O performance
- Simulation / analytical models to compare approaches
- Model quality $\rightarrow$ conclusion reliability
- Simple models widely used, often inaccurate

High-Quality Disk Models

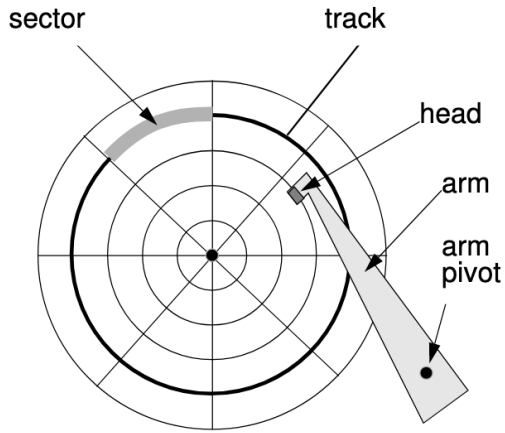
- Separate performance components
- Trade-off: modeling effort vs accuracy
- Detailed disk characteristics available

Disk Architecture

- **Mechanism:**
  - Recording: rotating disks + read/write heads
  - Positioning: arm assembly + track-following system
- **Controller:**
  - Microprocessor + buffer memory
  - SCSI interface
  - Manages data storage/retrieval
  - Maps logical addresses → physical sectors



a. side view.

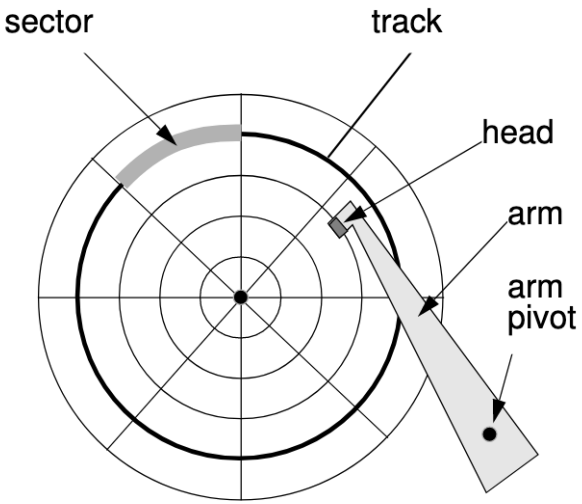


b. top view.

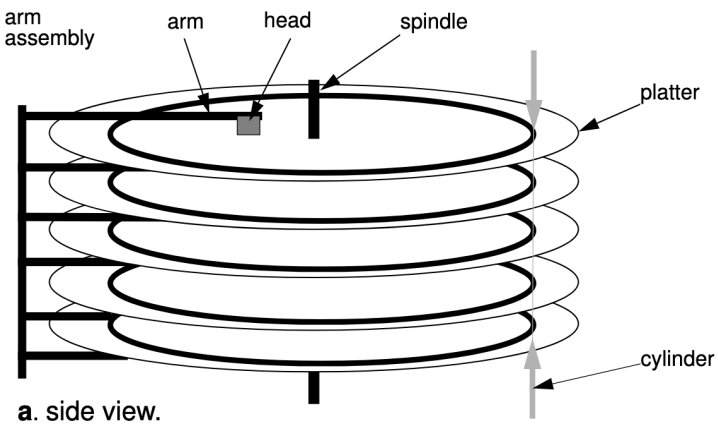
Modeling Considerations

- Not all features equally impact model accuracy
- Emphasize key performance-affecting elements

Disk Diameters



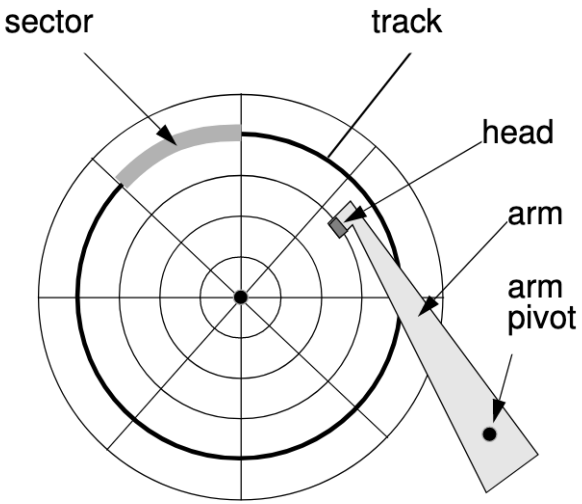
b. top view.



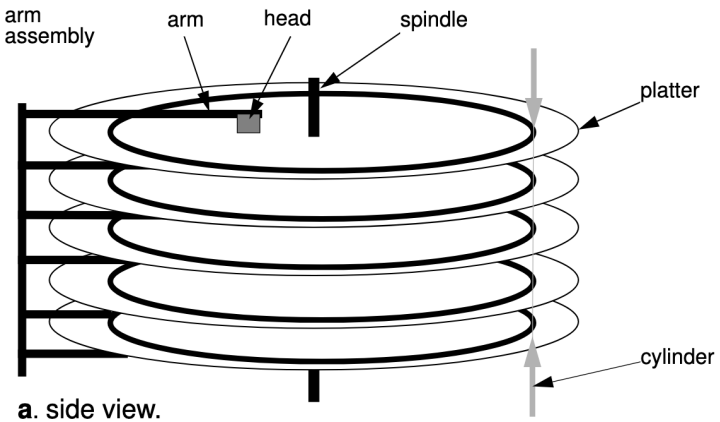
a. side view.

- Modern disks: 1.3 – 8 inches; common: 2.5, 3.5, 5.25
- Smaller disks: less storage, lower power, faster spin
- Cost-effective choice shifts as storage density increases

Storage Density Improvements



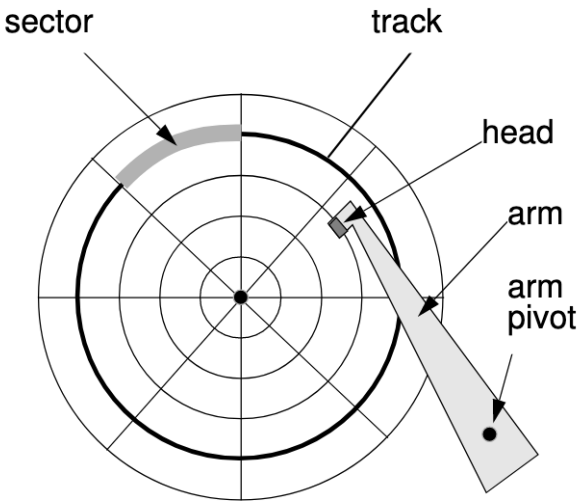
b. top view.



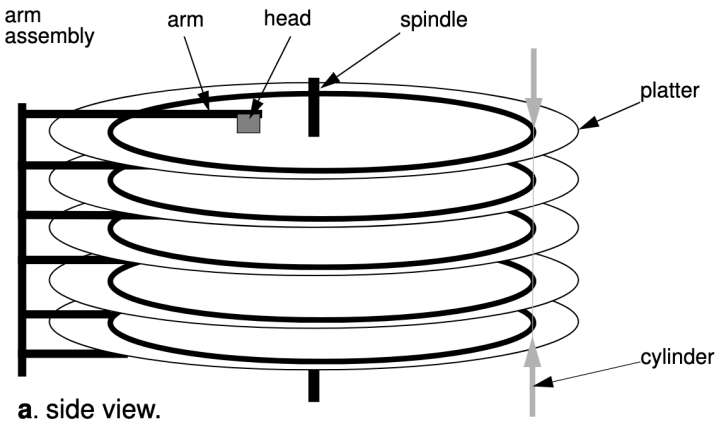
a. side view.

- Two main improvements:
  - Linear recording density (currently 50,000 bpi)
  - Track packing density (currently 2,500 TPI)
- Growth rate > 60
- Platters: 1–12 per disk, rotate on central spindle

Spin Speed



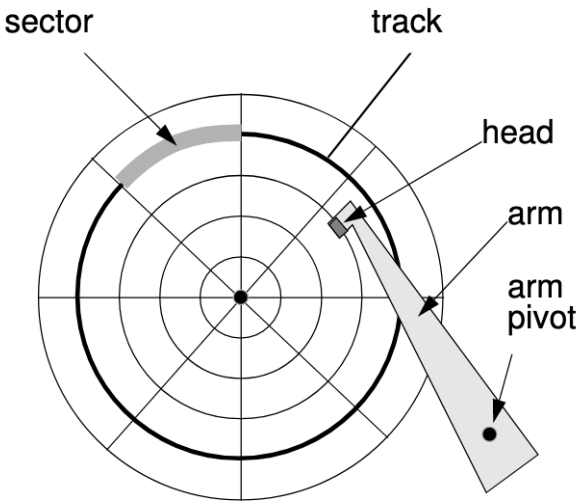
b. top view.



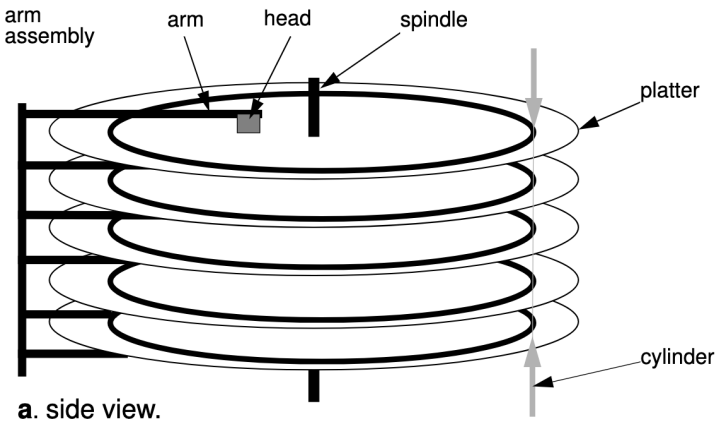
a. side view.

- Historical standard: 3,600 rpm
- Current max: 7,200 rpm
- Median speed growth 12
- Higher spin $\rightarrow$ faster transfer, lower latency, more power

Read/Write Channel



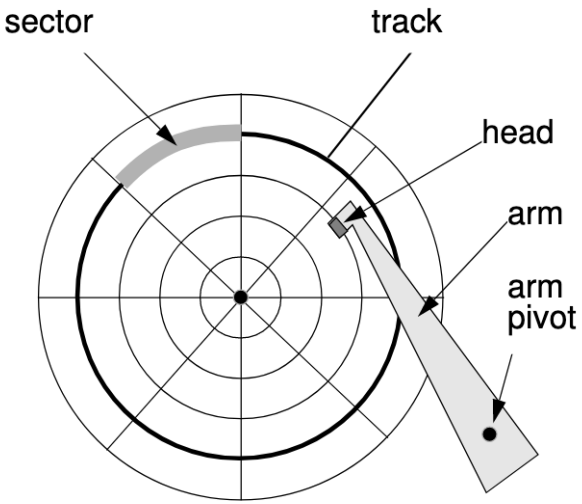
b. top view.



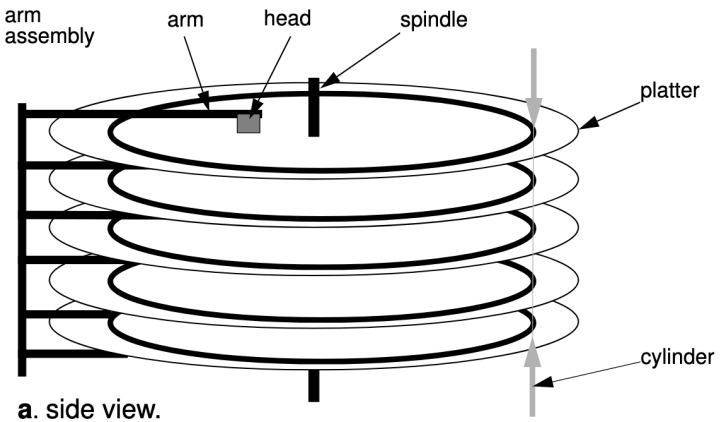
a. side view.

- One channel per disk (multichannel optional)
- Encodes/decodes data via magnetic flux
- Error correction embedded in data stream
- Multichannel $\rightarrow$ higher throughput, cost $\uparrow$, technical complexity $\uparrow$

Summary of Recording Components



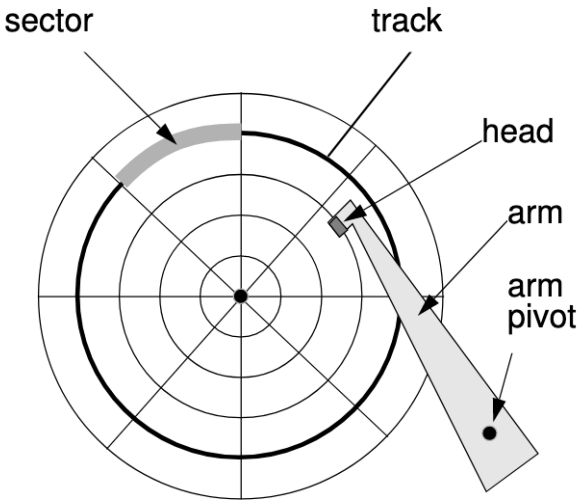
b. top view.



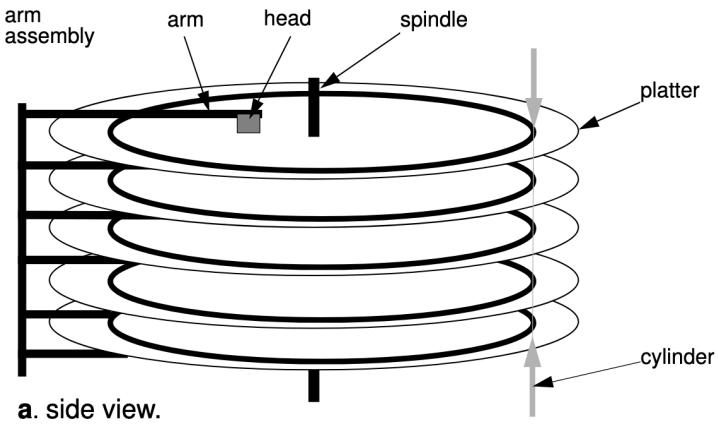
a. side view.

- Disk size, platters, and spindle define mechanical limits
- Linear & track density drive storage growth
- Spin speed affects latency & throughput
- Channel architecture governs read/write performance

Cylinders & Tracks



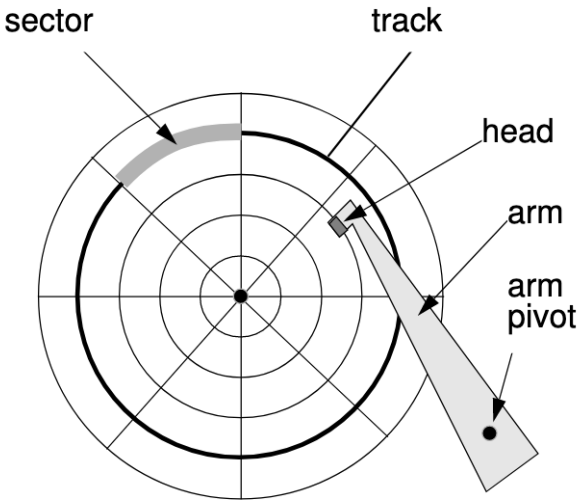
b. top view.



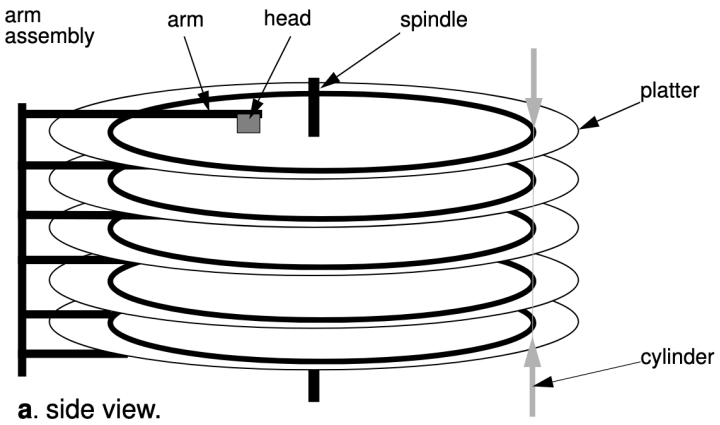
a. side view.

- Data stored in concentric tracks
- Cylinder = same track across platters
- Typical 3.5" disk: 2,000 cylinders
- Higher track density $\rightarrow$ treat tracks independently

Disk Arm & Pivot



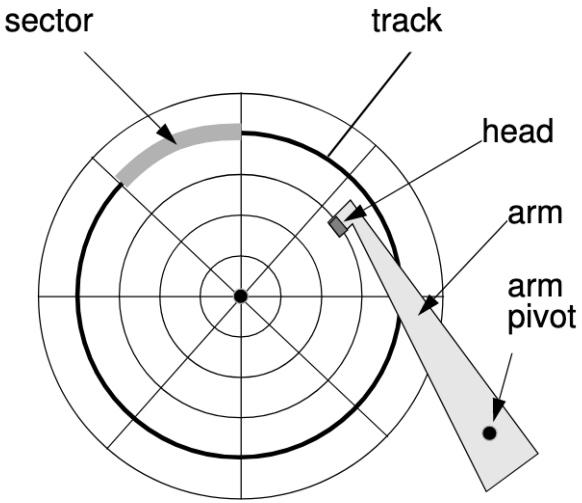
b. top view.



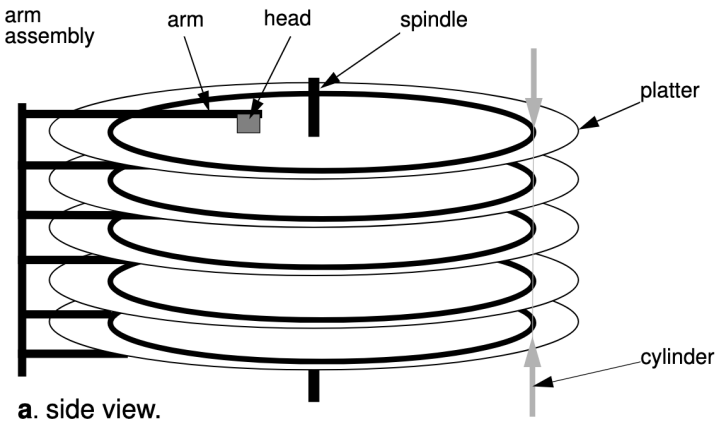
a. side view.

- Each head attached to pivoted arm
- Moving one head moves all arms
- Pivot system resists shocks better than linear sliders
- Arm stiffness + power limit $\rightarrow$ max acceleration 30–40g

Seek Phases



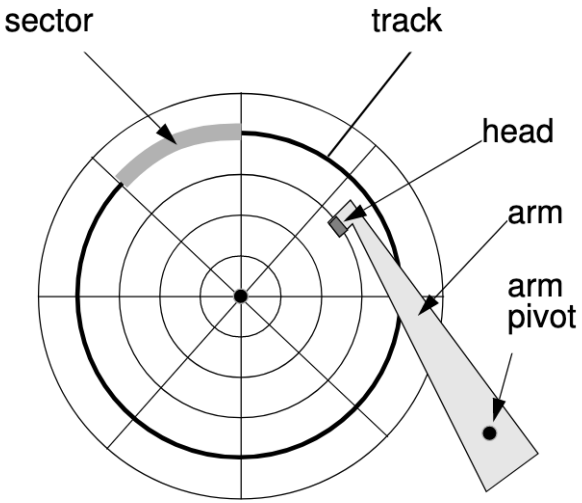
b. top view.



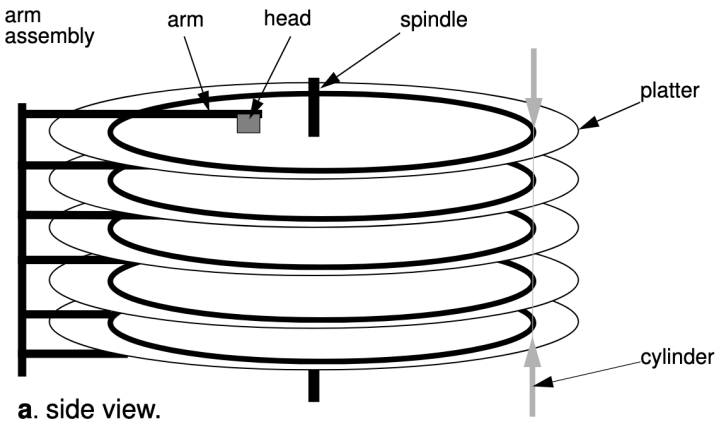
a. side view.

- Speedup → arm accelerates
- Coast → long seeks at max velocity
- Slowdown → decelerate near track
- Settle → precise positioning

Seek Times & Settling



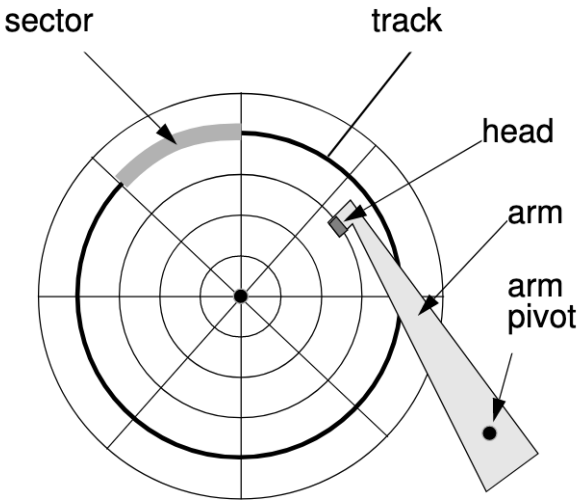
b. top view.



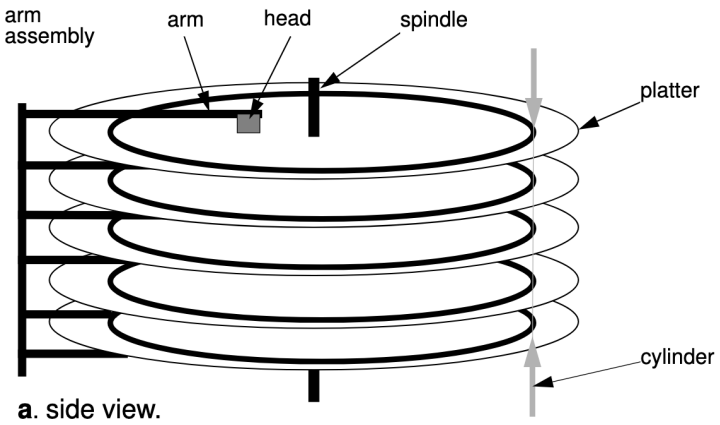
a. side view.

- Very short seeks dominated by settle (1–3 ms)
- Short seeks (<200–400 cyl) → mainly acceleration
- Long seeks → constant-speed phase dominates
- Settle fraction ↑ with smaller disks & higher track density

Average Seek Time



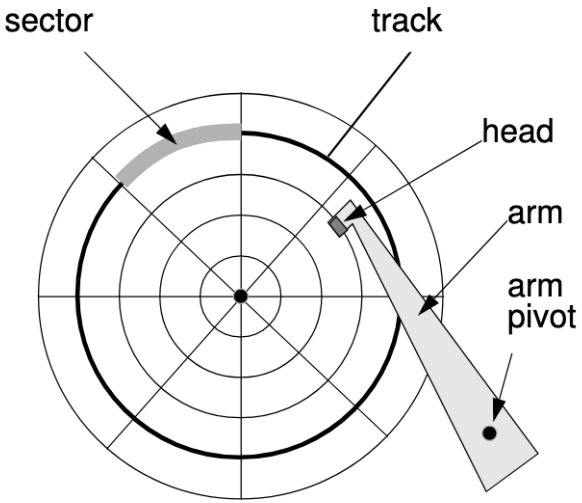
b. top view.



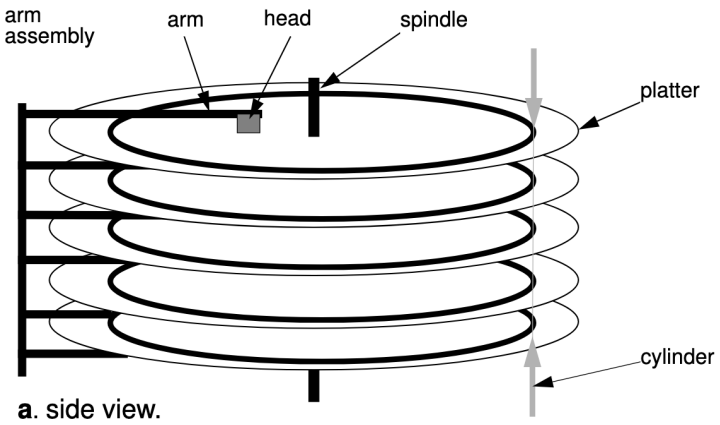
a. side view.

- Average seek often misleading
- Calculated in multiple ways: one-third stroke, full-stroke/3, weighted
- Short seeks more frequent $\rightarrow$ weighting improves accuracy
- Key for modeling: seek-time vs distance profile

Controller & Seek Info



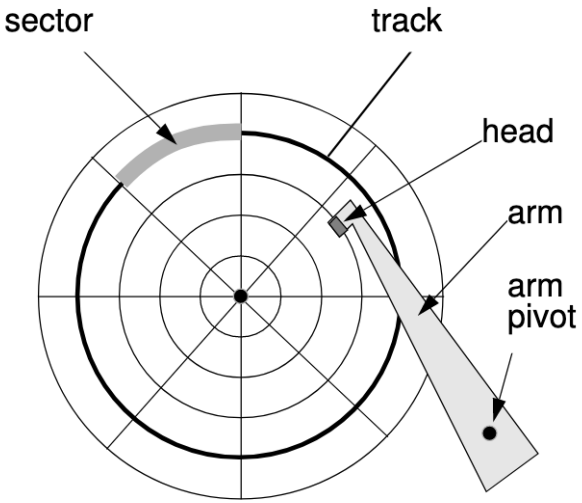
b. top view.



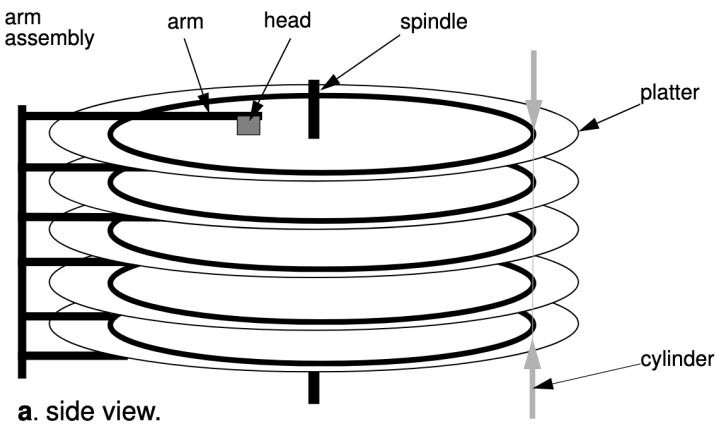
a. side view.

- Disk controller stores tabular seek-time data
- Subset of values + interpolation for intermediate distances
- Fine-grained profile $\rightarrow$ sawtooth-like
- Occasional recalibration needed (500–800 ms)

Thermal Expansion & Recalibration



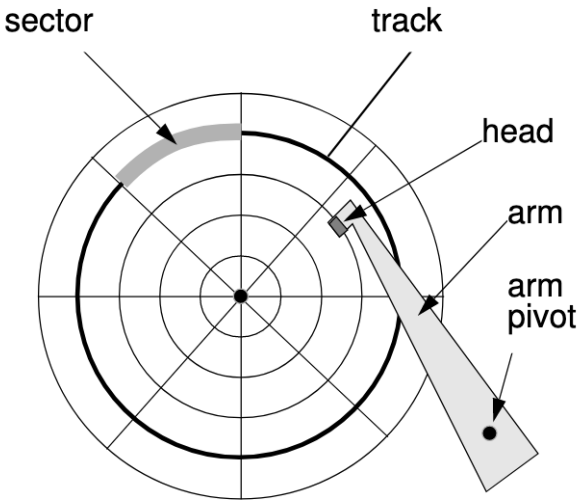
b. top view.



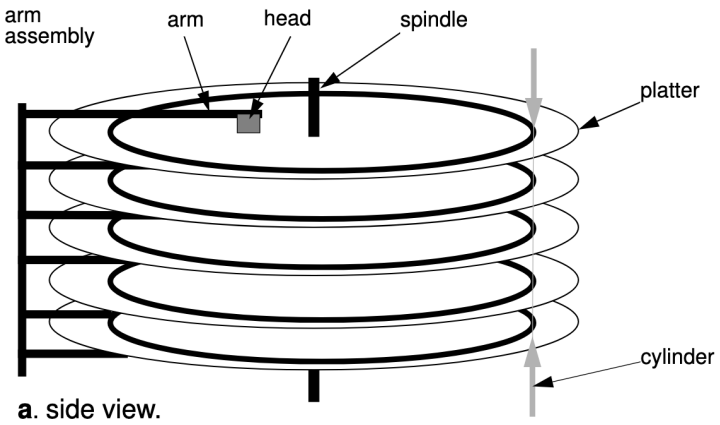
a. side view.

- Temperature, bearing stickiness $\rightarrow$ recalibration
- Triggered by temperature change or timers
- Most frequent at power-on
- Steady-state: once every 15–30 min
- Firmware may allow host-controlled scheduling

Track-Following System



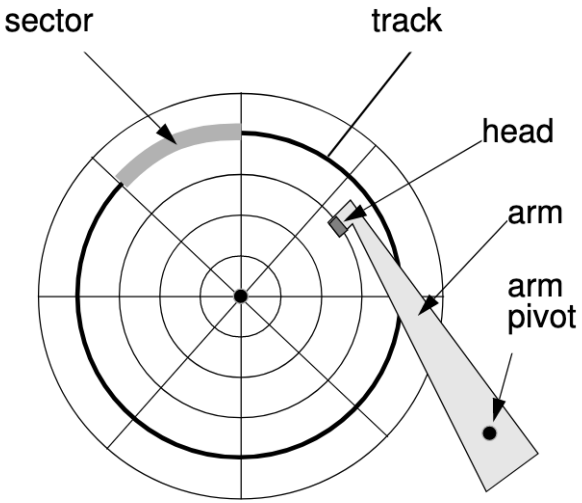
b. top view.



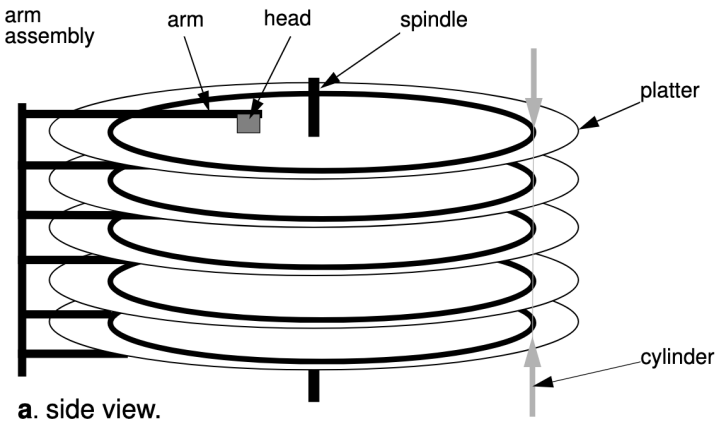
a. side view.

- Fine-tunes head after seek
- Uses embedded or dedicated positioning info
- Head & track switches included
- Time for switch 0.5–1.5 ms
- Head-switching $\rightarrow$ approaching track-switch times with high density

Optimistic Read Settling



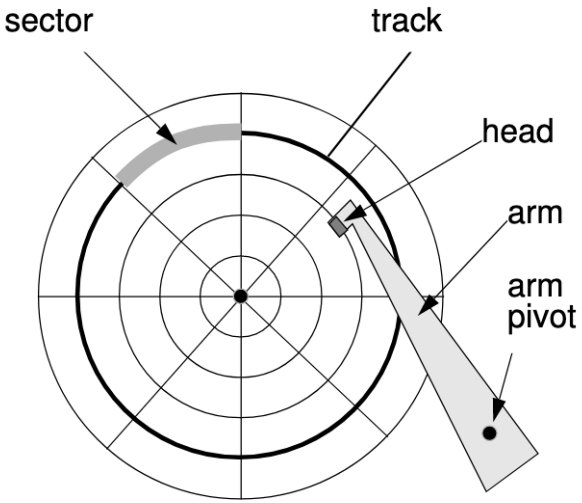
b. top view.



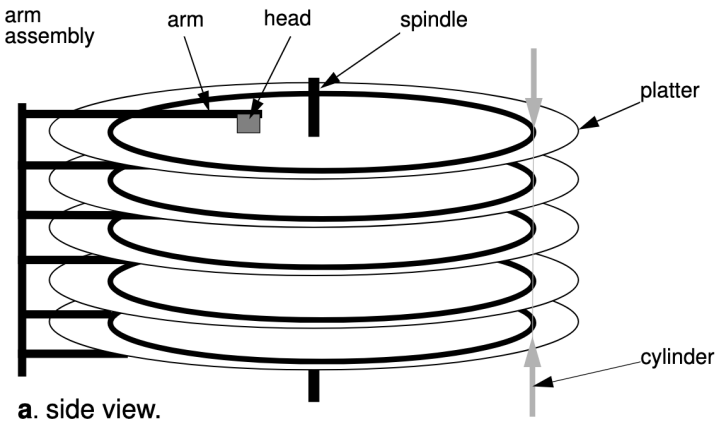
a. side view.

- Reads may start before settle completes
- Misreads → corrected by error correction
- Not applied to writes → avoid data loss
- Read vs write settle difference 0.75 ms

Data Layout



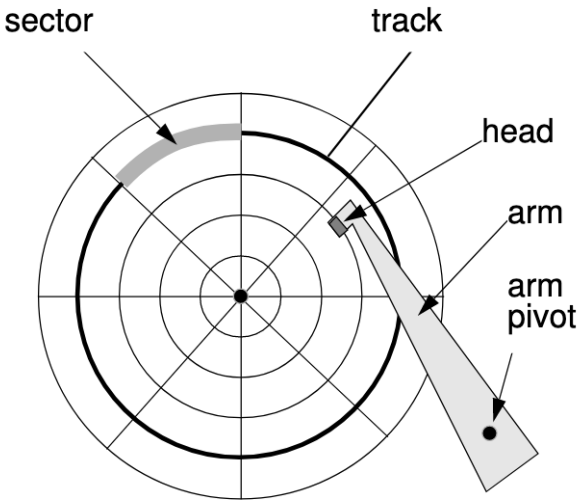
b. top view.



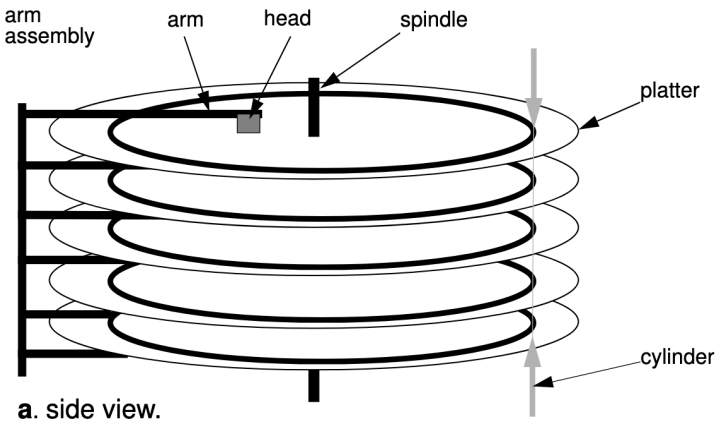
a. side view.

- SCSI disk: linear vector of blocks (256–1024 B)
- Blocks mapped to physical sectors
- Logical vs physical $\rightarrow$ hides bad sectors
- Low-level performance optimizations

Zoning



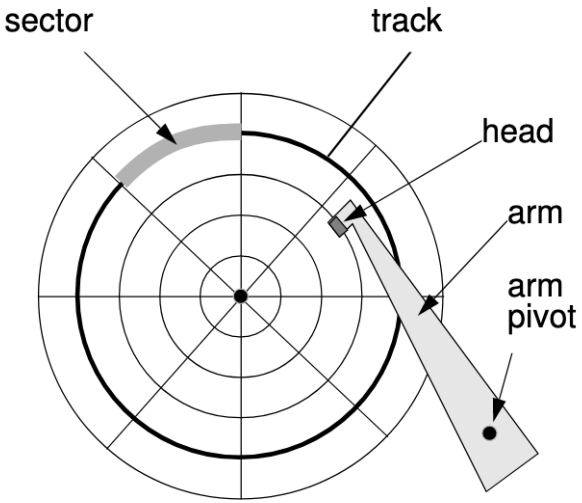
b. top view.



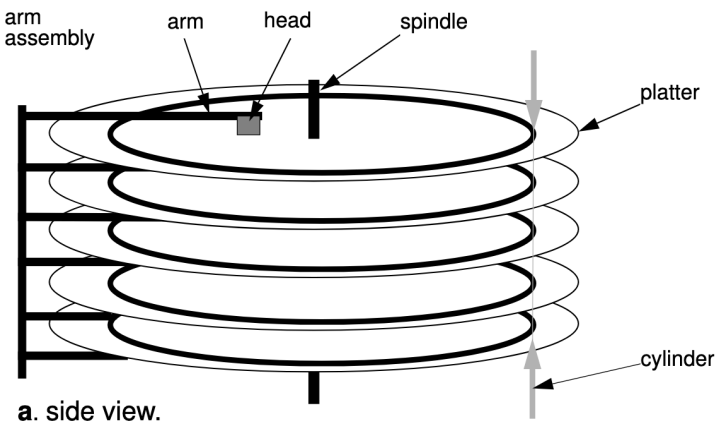
a. side view.

- Outer tracks longer → more sectors
- Cylinders grouped into zones (3–20)
- Outer zones → higher data transfer rates
- Example: HP C2240 → 3.1 MBps inner, 5.3 MBps outer

Track Skewing



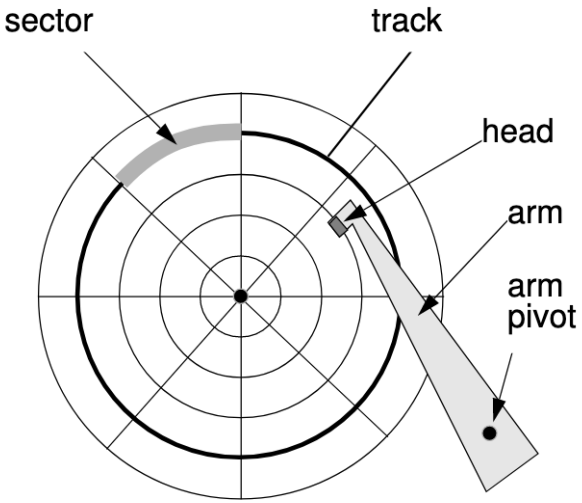
b. top view.



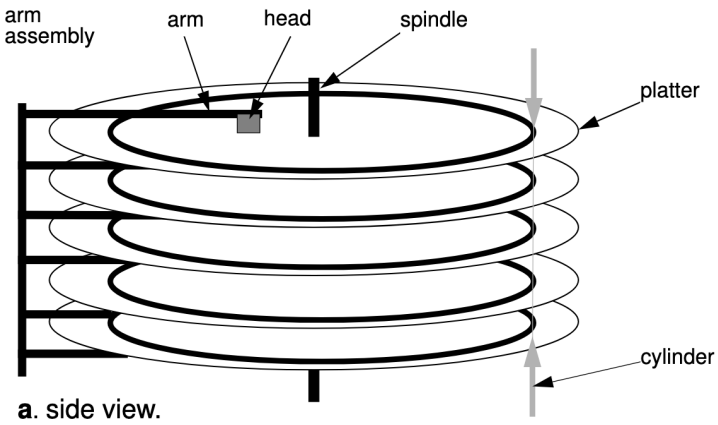
a. side view.

- Logical sector 0 offset per track
- Reduces delays across track/cylinder switches
- Each zone has its own skew factors

Sparing (Defect Management)



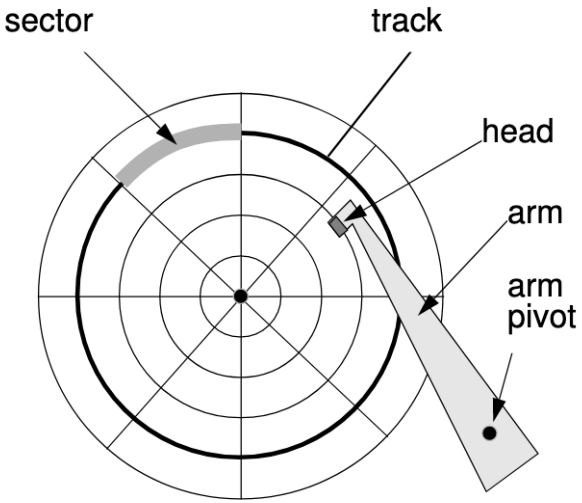
b. top view.



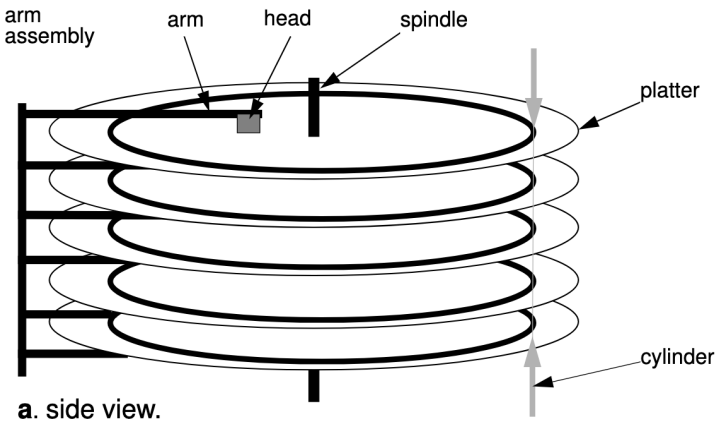
a. side view.

- Bad sectors detected during manufacturing
- Remapped $\rightarrow$ single-sector or slip-track
- Trade-off: performance vs space vs defect rate
- Example: HP C2240 uses both slip-track + single-track remapping

Summary of Positioning Components



b. top view.



a. side view.

- Tracks & cylinders define layout
- Arm & pivot → head movement & acceleration
- Seek phases + settle → total seek time
- Track-following ensures positioning
- Data layout, zoning, skewing, sparing → efficiency & reliability

Disk Controller Functions

- Mediates access to disk mechanism
- Runs track-following system
- Transfers data between disk and host
- Manages embedded cache
- Built on specialized microprocessors with DSP and hardware interfaces

Controller Trends

- Increasingly powerful controllers
- Replace dedicated electronics with firmware
- Handles sophisticated interfaces
- Controller overhead slowly declining: 0.3–1.0 ms

Bus Interface Overview

- Key aspects: topology, transfer rate, overhead
- SCSI bus widely used (up to 40 MBps)
- Alternative serial interfaces emerging (Fibre Channel)
- Multiple devices $\rightarrow$ bus contention possible

SCSI Synchronous Mode

- Synchronous mode $\rightarrow$ maximum bus speed
- Early SCSI: 5 MBps, Fast SCSI: 10 MBps
- Fast/Wide SCSI: 20 MBps
- Maximum transfer negotiated between host & disk

Bus Contention

- Multiple devices sharing bus $\rightarrow$ delays
- Important for large transfers or high controller overhead
- Low-level bus protocol overhead: few μ s if idle
- Disconnect/reconnect cycle: 200 μ s $\rightarrow$ allows higher overall throughput

Disk Buffering and Fence

- Older architectures: no buffering $\rightarrow$ wait entire revolution
- SCSI drives: speed-matching buffer masks bus/mechanism asynchrony
- Fence = amount read into buffer before bus transfer
- Write requests can overlap head repositioning up to buffer size

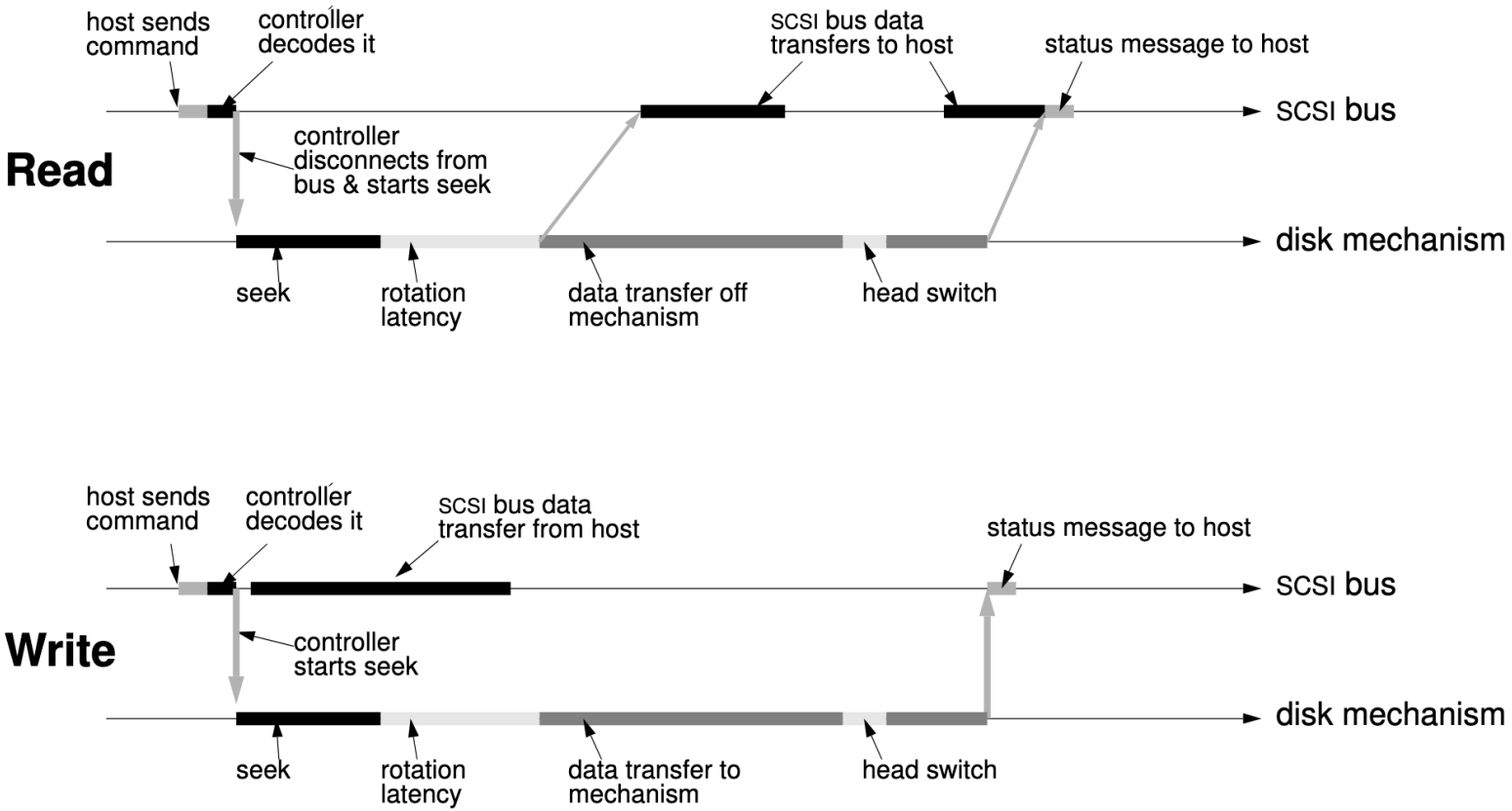
Read Caching

- Cache size: 64 KB – 1 MB (limited by cost & space)
- Read-ahead: prefetch data expected to be requested soon
- Partial cache hits: may serve from cache or bypass it
- Early "on-arrival" caching $\rightarrow$ zero-latency read for whole tracks

Read-Ahead Policies

- Continue reading where last request left off
- Or wait a full revolution for disk and host processing time
- Optimal for sequential reads
- Aggressive: cross track/cylinder boundaries $\rightarrow$ better sequential throughput
- Trade-off: degrades random access due to unabortable head switches

Read Write Graph



Write Caching

- Cache is volatile $\rightarrow$ careful management required
- Immediate reporting: writes acknowledged once in cache
- Allows back-to-back sequential writes
- Metadata writes often excluded from immediate reporting

Nonvolatile Write Cache

- Battery-backed RAM → retains data on power loss
- Benefits:
  - Accept all writes fitting in buffer
  - Reduced latency
  - Overwritten data → fewer physical writes
  - Controller can schedule writes optimally

Handling Read Hits in Cache

- Buffered copy treated as primary
- Ensures correct data returned before disk write completes
- Nonvolatile cache simplifies this
- Controller must track hits for both reads and writes

Command Queuing

- Multiple outstanding requests supported
- Controller determines optimal execution order
- Host may provide constraints (e.g., priority)
- Disk rotation awareness $\rightarrow$ better scheduling

Read & Write Overlap

- Write to buffer can overlap head repositioning
- Read-ahead overlaps rotation latency
- Maximizes bus and mechanism utilization
- Efficient caching critical for performance modeling

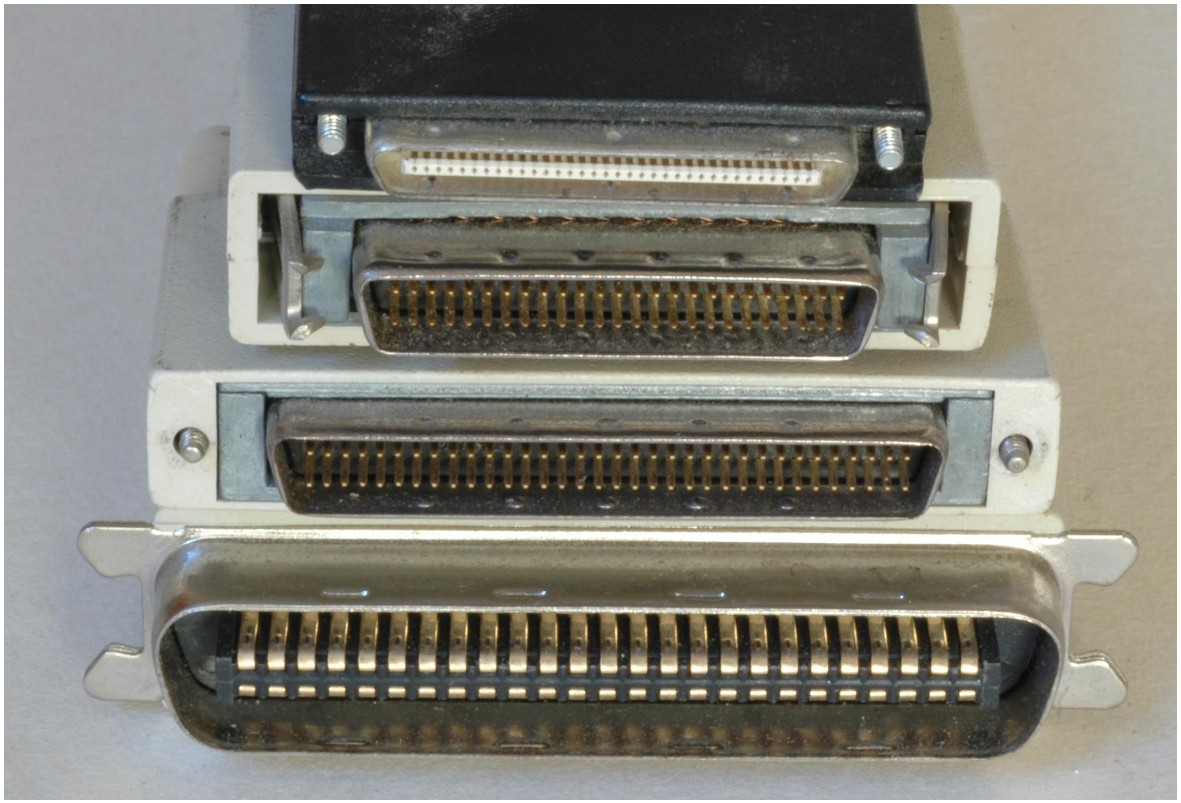
Multi-stream Caching

- Single read-ahead cache $\rightarrow$ single sequential stream only
- Interleaved streams $\rightarrow$ no benefit
- Segmented cache solves problem: e.g., 256 KB $\rightarrow$ 8×32 KB segments
- Controller configurable for multiple simultaneous streams

Summary of Disk Controller

- Controller manages mechanism, cache, and bus
- Handles SCSI protocol & command queuing
- Buffers and fences mask asynchronous transfers
- Read/write caching essential for performance
- Nonvolatile cache + intelligent scheduling $\rightarrow$ latency & throughput optimization

SCSI



ATA

ATA vs. SCSI Debate

- Often phrased as ATA vs. SCSI, but interface is least significant difference
- Differences in mechanics, materials, electronics, firmware
- PS drives with SCSI interface exist; ES drives in high-end PCs
- No inherent reason ES can't have ATA interface

Personal Storage (PS)

- Key quality: Cost commensurate with system
- Low cost dominates design
- First low-cost hard discs from PC market pressure
- No hard drives in early PCs; too big and expensive

Enterprise Storage (ES)

- Used on large systems: Big, expensive, large data access
- Support many users simultaneously
- Properties: Aggregation, random access to large data, reliability, performance critical
- Failure impacts business; faster service supports more users

Key Requirements: Cost

- Constant pressure to reduce costs despite complexity
- More logic for encoding, error correction, servo with areal density improvements
- Greater precision, lower tolerances for noise
- Components more complex yet less costly

Key Requirements: Seek Performance

- Faster head movement: Higher performance magnets, faster processors, lower-mass actuators
- Sophisticated modeling for ES to optimize structures
- Avoid throttling by resonances

Key Requirements: Rotational Latency

- Improved by faster spin
- PS slower to adopt; only when no marginal cost
- ES drives innovation; PS adopts when cheap
- ES: Costly innovation; PS: Cost savings

Key Requirements: Aggregation

- ES in groups; interactions decrease performance
- Rotational vibration from seeking drives
- FC/SCSI/SAS attach more drives than IDE

Key Requirements: Reliability

- Varies with usage: PS several hours/day; ES 24/7
- PS compromises for cost; less suited to stresses

History of Interfaces

- ATA: Programmed I/O, host processor handles all
- SCSI: External chip for independent operation
- ATA adding queuing; lacks multiple CPU, variable block, dual porting
- ES functionality adds complexity to PS

Mechanics Overview

- Affect reliability, seek time, acoustics, environmental resistance
- ES: Higher rpm, tolerance for disturbance
- More rigidity, mass, bandwidth servos
- Higher power, more heat

Head/Disc Assembly (HDA)

- ES: Tighter tolerances, better sealing, filters, desiccant, absorbents
- PS: Compromises to save cost, e.g., no O-rings or desiccant
- ES: More shrouding for air flow, cooling
- Size/stiffness of casting/cover impact acoustics and vibration

Actuator

- Larger magnets for faster seeks, but higher cost
- ES: Lower resistance coils, cooling features
- Bi-stable latch for performance
- Independent bonding for rigidity
- PS: Cost first, then seek performance

Spindle

- Higher rpm challenge: Keep head on track
- ES: More expensive motors, captured at both ends
- Fluid bearings minimize runout
- PS: Cantilever design for cost

Electronics

- More integrated; fewer components
- ES: More silicon, e.g., 2x ASIC gates, SRAM, flash
- Support multiple initiators, tagged commands
- Two processors: Servo and interface/read-write
- PS: Single processor for all tasks

Memory

- ES: Firmware >2x ATA; more flash and SRAM
- Complex command set, queues require more space
- Vendor-specific extensions

Magnetics: Heads

- Similar areal density push
- ES: Higher rpm for data rates
- Writing: Benefits from velocity; ES stretches capability
- Reading: Harder at high rpm due to noise
- ES: More expensive electronics for SNR

Magnetics: Materials

- Aluminum substrate traditional
- Glass: Better uniformity, stiffness; but harder deposition
- AFC media: Additional layers for higher density, more complexity

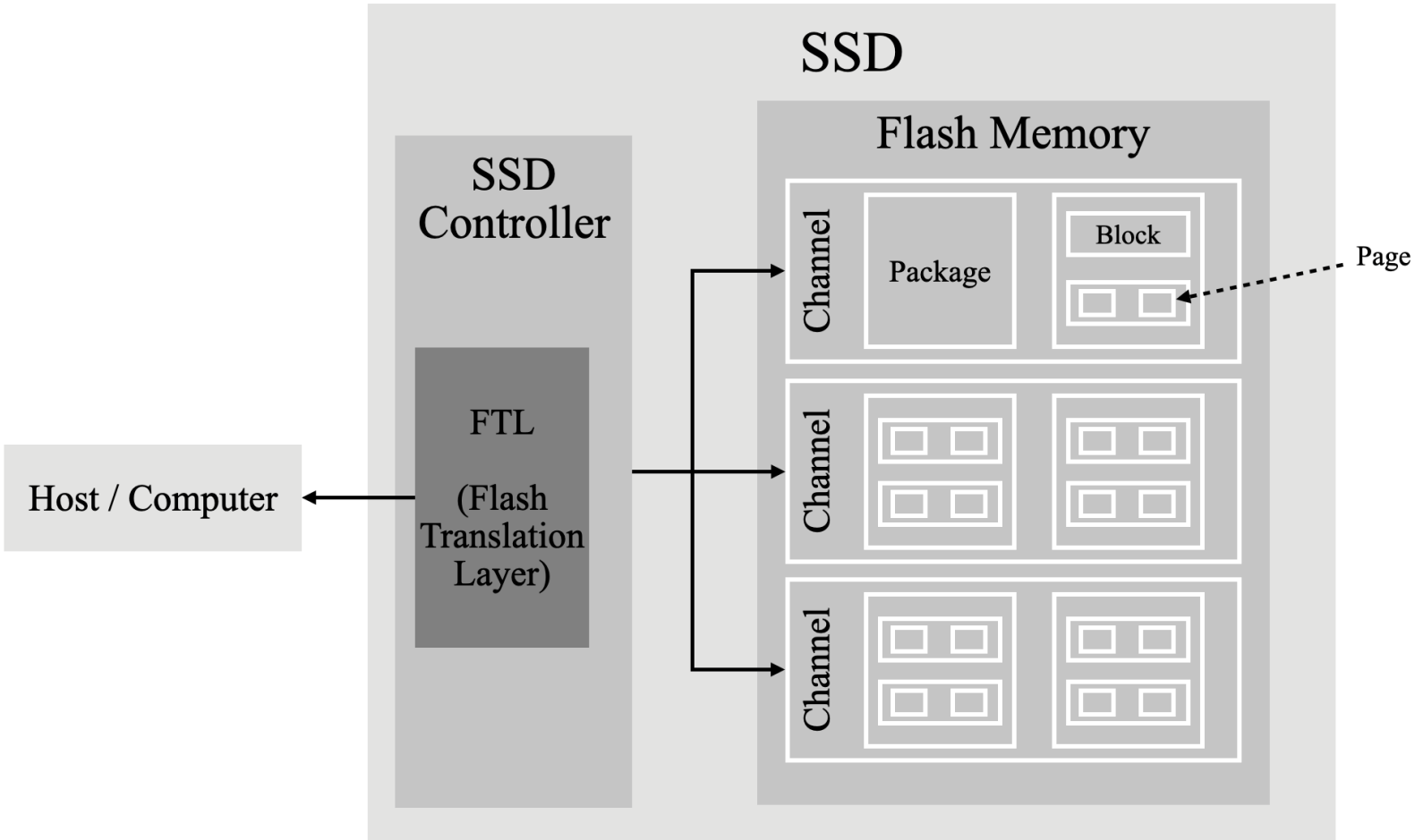
Manufacturing

- ES: Longer build/test times for reliability
- Characterization, flaw analysis
- PS: Shorter to save cost

Unwritten Contract of Solid State Drives

The Unwritten Contract of Solid State Drives

Structure of Solid State Drives (SSD)



Contract of SSDs

- **Written Contract**
  - Defined and documented by standards
  - Specify correctness rules: command format, data integrity, etc.
  - Violations lead to Errors or Command rejection
- **Unwritten Contract**
  - Implicitly expected behaviors not documented
  - Performance expectations: latency, throughput, etc.
  - Violations lead to performance degradation

Unwritten Contract

Implicit performance rules required for optimal SSD behavior:

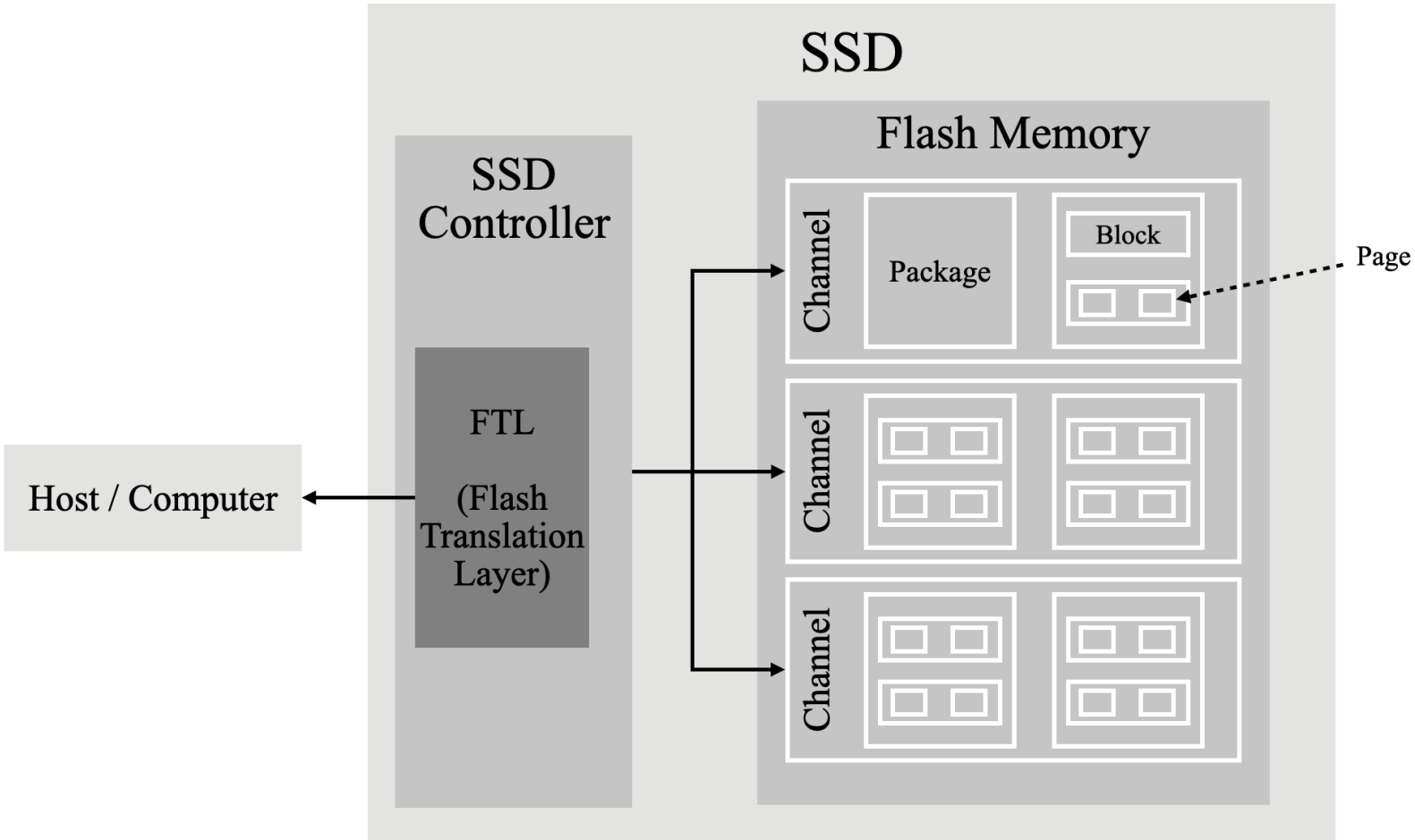
- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Request Scale

Implicit performance rules required for optimal SSD behavior:

- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Request Scale



Request Scale

SSD clients should issue large data requests or multiple concurrent requests.

Locality

Implicit performance rules required for optimal SSD behavior:

- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Locality

Why Locality Matters in SSDs

- **No in-place updates → Page mapping:**
  - *Read*: Page level
  - *Write*: Page level
  - *Erase*: Block level

- **Insufficient RAM for full mapping → On-demand FTLs**
  - Map pages loaded from flash to RAM on demand
  - Localized accesses reduce page loads

Locality

SSD clients should access with locality.

Aligned Sequentiality

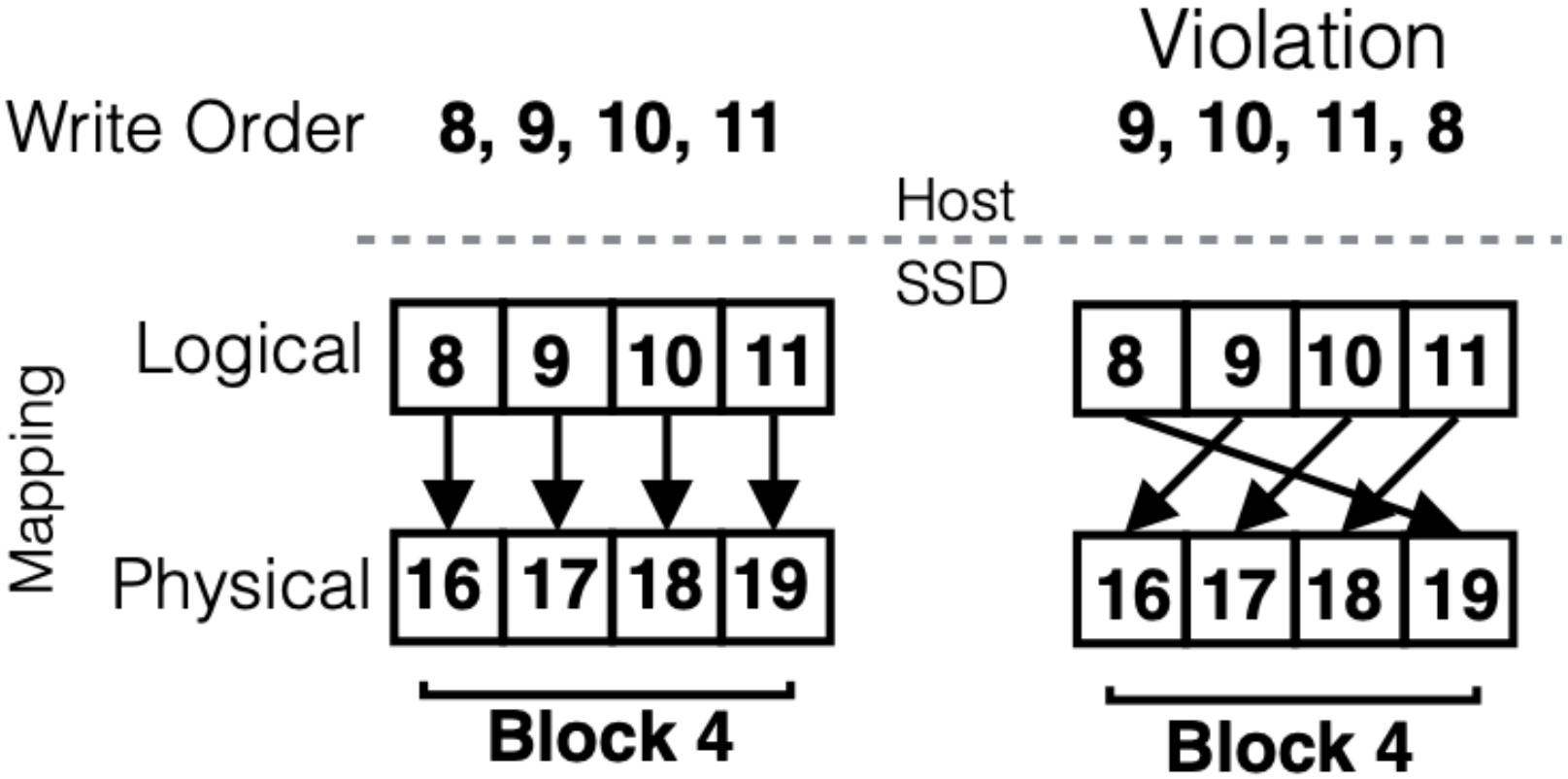
Implicit performance rules required for optimal SSD behavior:

- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Aligned Sequentiality

- **Hydrid mapping**
  - Combine page-level and block-level mapping
  - Reduces mapping table size and RAM usage

Aligned Sequentiality



Aligned Sequentiality

SSDs with hybrid FTLs should start writing at the aligned beginning of a block boundary and write sequentially.

Grouping by Death Time

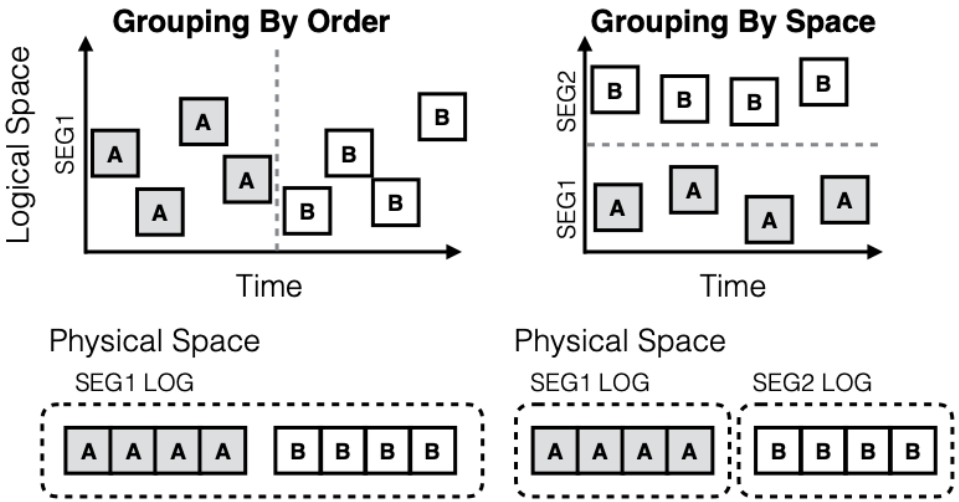
Implicit performance rules required for optimal SSD behavior:

- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Grouping by Death Time

- Death time: the time that the page is discarded or updated
- Zombie window: time that both valid and invalid pages coexist in a block
- Zombie block: block in zombie window - higher Garbage Collection (GC) cost

Grouping by Death Time



Reduce Zombie Windows:

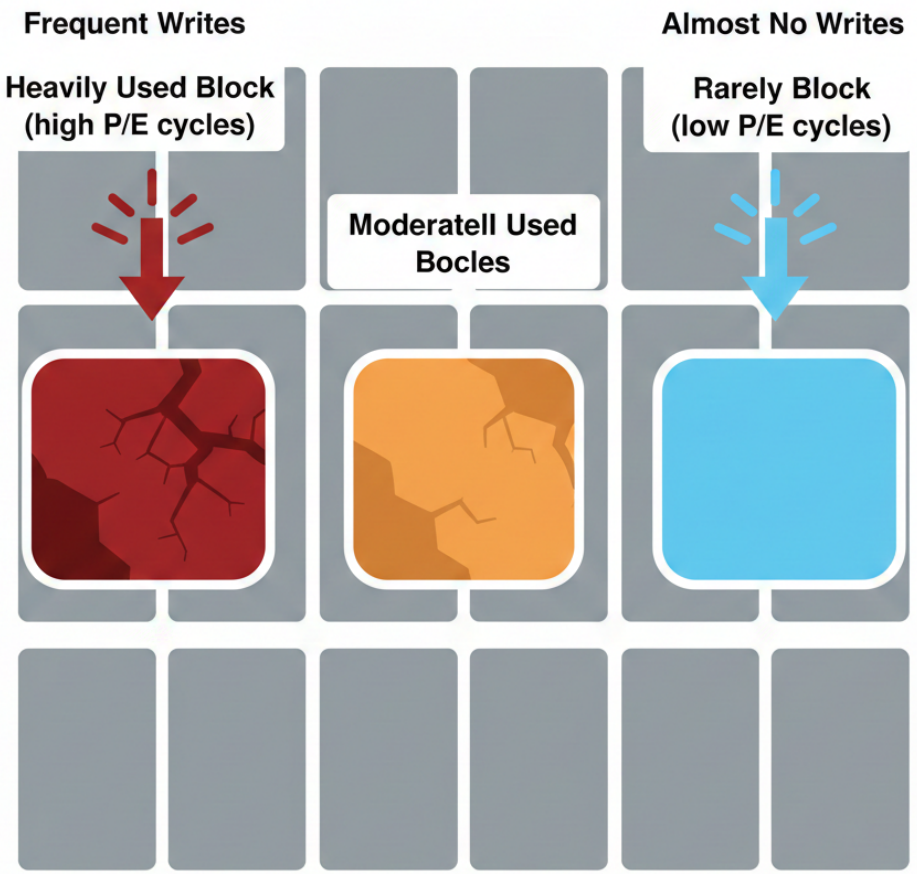
- Grouping by order
- Grouping by space

Uniform Data Lifetime

Implicit performance rules required for optimal SSD behavior:

- Request Scale
- Locality
- Aligned Sequentiality
- Grouping by Death Time
- Uniform Data Lifetime

Uniform Data Lifetime



Uniform Data Lifetime

Solutions:

- **Dynamic Wear Leveling**: evens the P/E count by using a less-used block when a new block is needed
- **Static Wear Leveling**: periodically moves static data from blocks with low P/E counts to blocks with high P/E counts

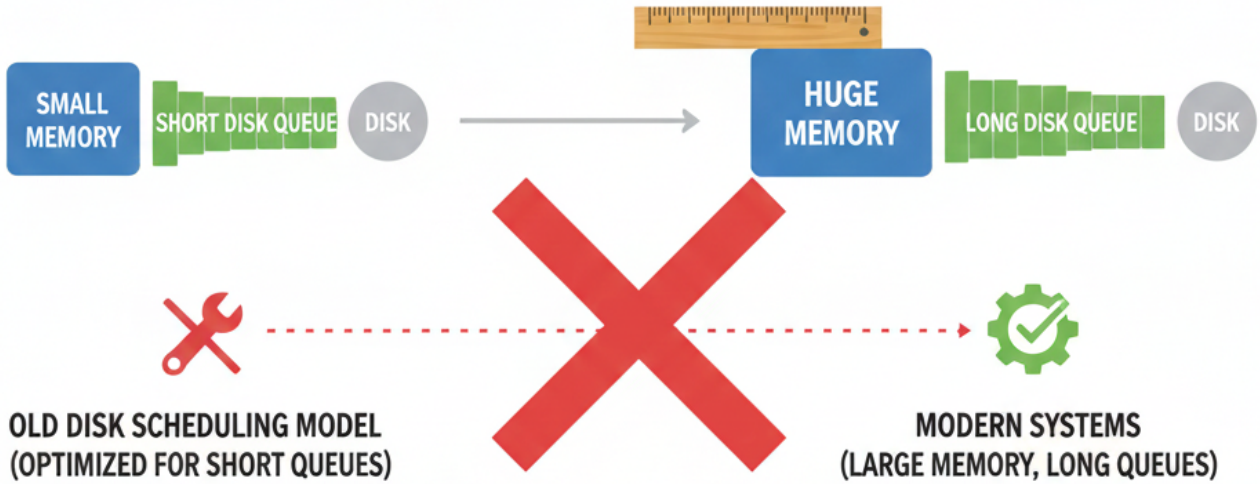
Uniform Data Lifetime

clients of SSDs should create data with similar lifetimes.

Disk Scheduling Revisited

Disk Scheduling Revisited

Background



MISALAGNMENT:
OLD OPTIMIZATION APPLIED TO NEW SYSTEMS

Related Works

- FCFS (First-Come-First-Serve)
  - Simple but poor performance
- SSF (Shortest-Seek-Time-First)
  - Reduces seek time but may cause starvation
- SCAN (Elevator Algorithm)
  - Fairer than SSF but longer wait times
- C-SCAN (Circular SCAN)
  - More uniform wait times than SCAN

Problems

Access Time = Seek Time + Rotational Latency + Transfer Time

Proposed Methods

Shortest Time First (STF)

- Grouped Shortest Time First (GSTF)
- Weighted Shortest Time First (WSTF)

GSTF

Grouped Shortest Time First (GSTF)

- Divide requests into cylinder groups
- Apply STF within each group
- **Trade-off:**
  - Larger group size $\rightarrow$ higher disk utilization
  - Larger group size $\rightarrow$ longer maximum response time

WSTF

Weighted Shortest Time First (WSTF)

- Prioritize requests with the lowest weighted time
- **Weight Calculation:** $T_{\mathrm{W}} = T_{\mathrm{real}} \frac{M-E}{M}$
  - T_{real}: actual estimated time
  - E: elapsed time since request arrival
  - M: maximum wait time threshold

[1] D. Anderson and J. Dykes, “More than an {Interface—SCSI} vs.{ATA},” in *2nd USENIX Conference on File and Storage Technologies (FAST 03)*, 2003.

[2] J. He, S. Kannan, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, “The unwritten contract of solid state drives,” in *Proceedings of the twelfth European conference on computer systems*, 2017, pp. 127–144.

[3] M. Seltzer, P. Chen, and J. Ousterhout, “Disk scheduling revisited,” in *Proceedings of the winter 1990 USENIX technical conference*. Washington, DC, 1990, pp. 313–323.

[4] C. Ruemmler and J. Wilkes, “An introduction to disk drive modeling,” *Computer*, vol. 27, no. 3, pp. 17–28, 1994.

Thanks!