Physical Storage Systems 2

Instructor: Beom Heyn Kim

beomheynkim@hanyang.ac.kr

Department of Computer Science

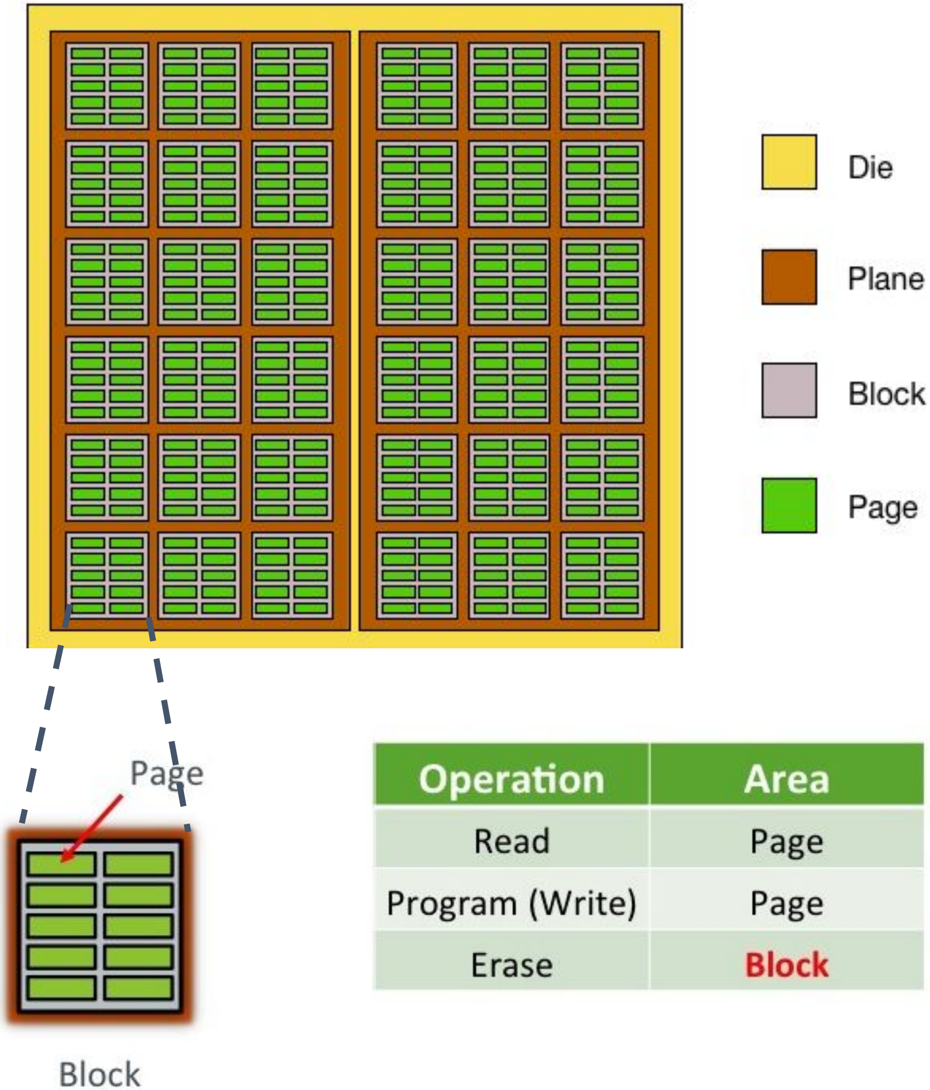
Overview

- Flash Memory
- RAID
- Dist-Block Access

Flash Memory

- Two types of flash memory:
  - NOR flash vs NAND flash
- NAND flash
  - used predominantly for data storage
  - cheaper than NOR flash
  - read/write at a page granularity (page: 512 bytes to 4 KB)
    - 20 to 100 μs for a page read, while 100 μs for a page write (random access on disks takes 5 to 10 ms)
    - Not much difference between sequential and random read
  - Page can only be written once
    - Must be erased first to allow rewrite
- **Solid state disks**
  - Built using NAND flash
  - Provide the same block-oriented disk interfaces
  - Transfer rate of up to 500 MB/s using SATA, and up to 3 GB/s using NVMe PCIe (200 MB/s with magnetic disks)

NAND Flash Operation's Granularity

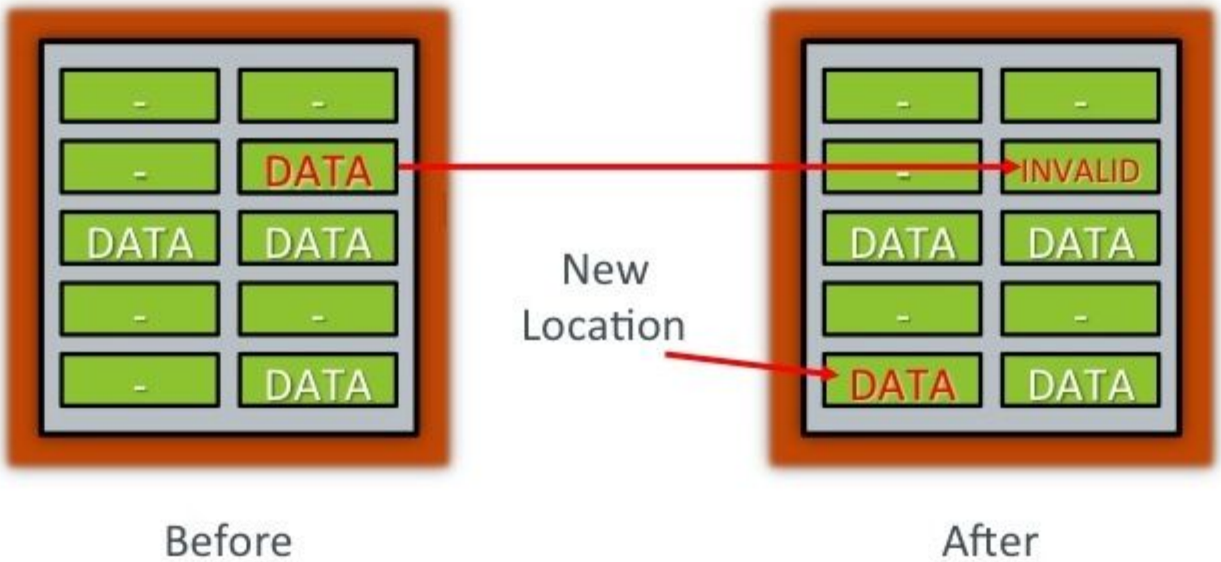


Operation	Area
Read	Page
Program (Write)	Page
Erase	**Block**

Flash Memory (Cont.)

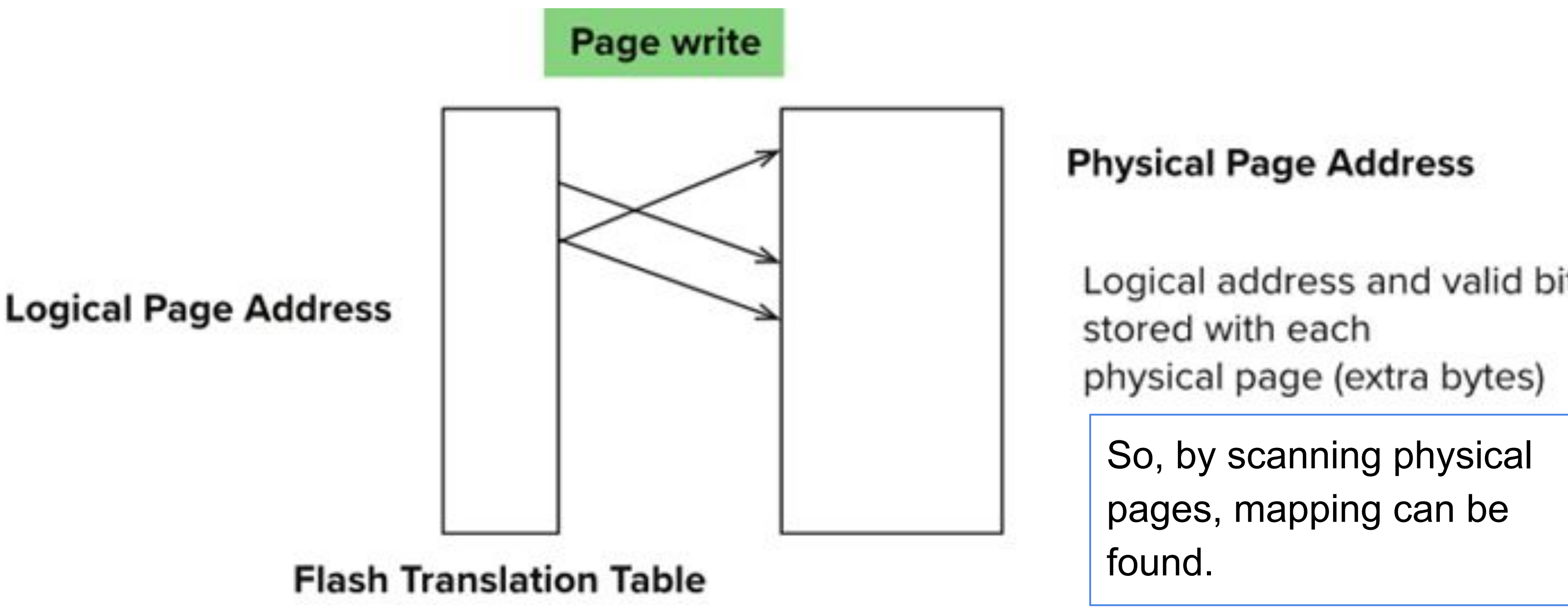
- Erase happens in units of **erase block**
  - A group of pages to erase
  - Erase block typically 256 KB to 1 MB (128 to 256 pages)
  - Takes 2 to 5 ms
  - There is a limit (around 100,000 and 1,000,000 times) to how many times a flash page can be erased
  - After 100,000 to 1,000,000 erases, erase block becomes unreliable and cannot be used
    - **wear leveling**

Remapping Logical Pages



Remapping of logical page addresses to physical page addresses avoids waiting for erase

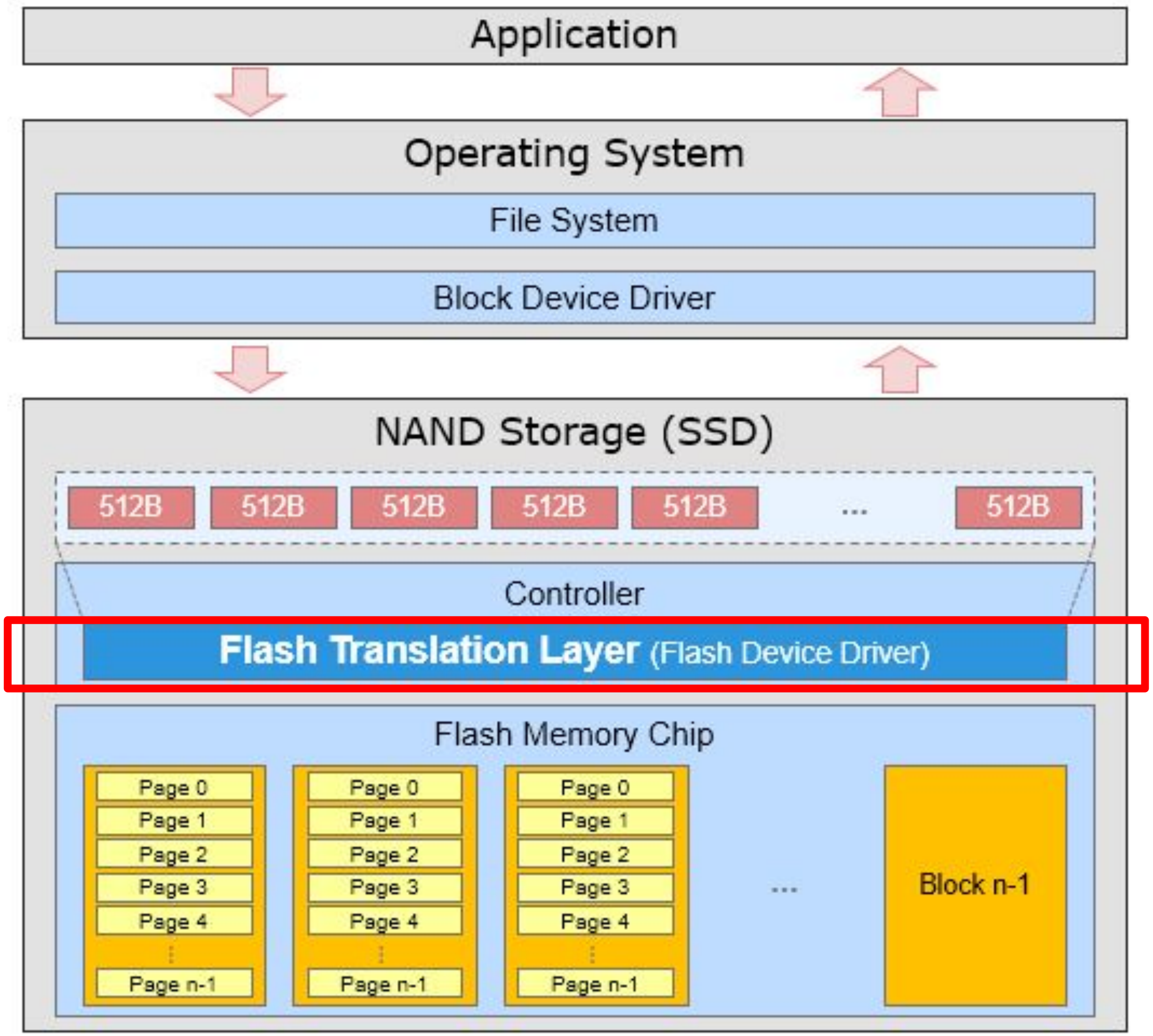
Flash Translation Table



So, by scanning physical pages, mapping can be found.

For quick access, in-memory **Flash Translation Table** tracks mapping

Flash Translation Layer



Remapping is carried out by
Flash Translation Layer (FTL)

SSD Performance

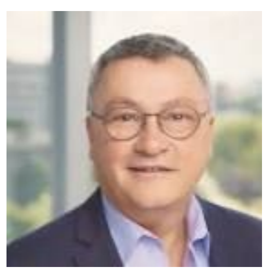
- Random reads/writes per second
  - Typical 4KB reads: 10,000 reads per second (10,000 IOPS)
  - Typical 4KB writes: 40,000 IOPS
  - SSDs support parallel reads
    - Typical 4KB reads:
      - 100,000 IOPS with 32 requests in parallel (QD-32) on SATA
      - 350,000 IOPS with QD-32 on NVMe PCIe
    - Typical 4KB writes:
      - 100,000 IOPS with QD-32, even higher on some models
- Data transfer rate for sequential reads/writes
  - 400 MB/sec for SATA3, 2 to 3 GB/sec using NVMe PCIe
- **Hybrid disks**: combine small amount of flash cache with larger magnetic disk

Overview

- Flash Memory
- RAID
- Dist-Block Access

RAID

Redundant
Array of
Independent
Drives



The term "RAID" was invented by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987.

RAID

- **RAID: Redundant Arrays of Independent Disks**
  - **high reliability** by storing data redundantly, so that data can be recovered even if a disk fails
  - **high capacity** and **high speed** by using multiple disks in parallel
- The chance that some disk out of a set of *N* disks will fail is much higher than the chance that a specific single disk will fail.
  - E.g., a system with 100 disks, each with MTTF of 100,000 hours (approx. 11 years), will have a system MTTF of 1000 hours (approx. 41 days)
  - Techniques for using redundancy to avoid data loss are critical with large numbers of disks

Improvement of Reliability via Redundancy

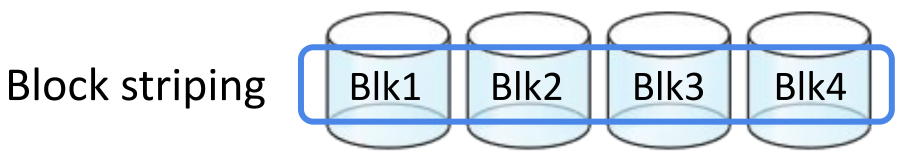
- **Redundancy** – store extra information that can be used to rebuild information lost in a disk failure
- E.g., **Mirroring** (or **shadowing**)
  - Logical disk consists of two physical disks.
  - Duplicate every disk.
    - Every write is carried out on both disks
    - Reads can take place from either disk
  - If one disk in a pair fails, data still available in the other
    - Data loss would occur only if both disks fail (e.g., fire or building collapse or electrical power surges)
      - The probability is very low
- **Mean time to data loss** depends on mean time to failure, and **mean time to repair**
  - E.g., MTTF of 100,000 hours, mean time to repair of 10 hours gives mean time to data loss of $500*10^6$ hours (or 57,000 years) for a mirrored pair of disks (ignoring dependent failure modes)

Improvement in Performance via Parallelism

- Two main goals of parallelism in a disk system:
  1. Load balance multiple small accesses to increase throughput
  2. Parallelize large accesses to reduce response time.
- Improve transfer rate by striping data across multiple disks.
- **Bit-level striping** – split the bits of each byte across multiple disks
  - In an array of eight disks, write bit *i* of each byte to disk *i.*
  - Each access can read data at eight times the rate of a single disk.
  - But seek/access time worse than for a single disk
    - Bit level striping is not used much anymore
- **Block-level striping** – with *n* disks, block *i* of a file goes to disk (*i* mod *n*) + 1
  - Requests for different blocks can run in parallel if the blocks reside on different disks
  - A request for a long sequence of blocks can utilize all disks in parallel

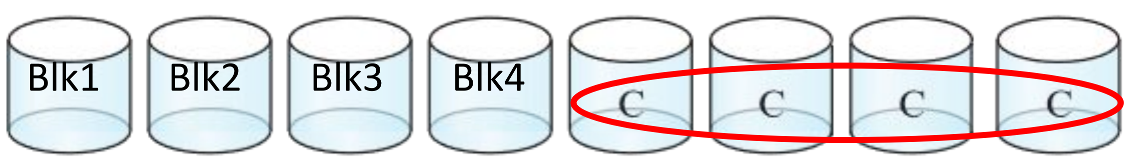
RAID Levels

- Schemes to provide redundancy at lower cost by using disk striping combined with parity bits
  - Different RAID organizations, or RAID levels, have differing cost, performance and reliability characteristics
- **RAID Level 0**: Block striping; non-redundant.
  - Used in high-performance applications where data loss is not critical.
- **RAID Level 1**: Mirrored disks with block striping
  - Offers best write performance.
  - Popular for applications such as storing log files in a database system.



(a) RAID 0: nonredundant striping

C: A second copy of the data



(b) RAID 1: mirrored disks (a.k.a. RAID 1+0 or RAID 10)

RAID Levels (Cont.)

- **Parity blocks**: Parity block *j* stores XOR of bits from block *j* of each disk
  - When writing data to a block *j*, parity block *j* must also be computed and written to disk
    - Can be done by using old parity block, old value of current block and new value of current block (2 block reads + 2 block writes)
    - Or by recomputing the parity value using the new values of blocks corresponding to the parity block
      - More efficient for writing large amounts of data sequentially
  - To recover data for a block, compute XOR of bits from all other blocks in the set including the parity block

Consider following block striping:

Disk 1 =1111

Disk 2 =1110

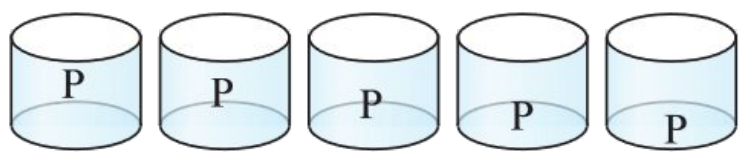
Disk 3 =1100

Disk 4 =1000

1. What is the parity block?
2. If Disk 1 gets updated to 1100, can you recompute parity block? (Try both methods)
3. Let's say you lost Disk 2, restore it by using the parity block

RAID Levels (Cont.)

- **RAID Level 5:** Block-Interleaved Distributed Parity; partitions data and parity among all *N* + 1 disks, rather than storing data in *N* disks and parity in 1 disk.
  - E.g., with 5 disks, parity block for *n*th set of blocks is stored on disk (*n mod* 5) + 1, with the data blocks stored on the other 4 disks.
  - Block writes occur in parallel if the blocks and their parity blocks are on different disks.

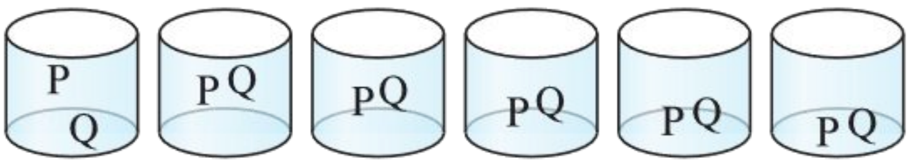


(c) RAID 5: block-interleaved distributed parity

P0	0	1	2	3
4	P1	5	6	7
8	9	P2	10	11
12	13	14	P3	15
16	17	18	19	P4

RAID Levels (Cont.)

- **RAID Level 6**: P+Q Redundancy scheme; similar to Level 5, but stores two error correction blocks (P, Q) instead of single parity block to guard against multiple disk failures. This scheme uses error-correcting codes such as the Reed-Solomon codes. (see https://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction)
  - Better reliability than Level 5 at a higher cost (slower write performance)
    - Becoming more important as storage sizes increase



(d) RAID 6: P + Q redundancy

- There are **other RAID levels but not used in practice:**
  - **RAID 2, RAID 3, RAID 4**

Choice of RAID Level

- Factors in choosing RAID level
  - Monetary cost
  - Performance: Number of I/O operations per second, and bandwidth during normal operation
  - Performance during failure
  - Performance during rebuild of failed disk
    - Including time taken to rebuild failed disk
- RAID 0 is used only when data safety is not important
  - E.g., data can be recovered quickly from other sources

Choice of RAID Level (Cont.)

- Level 1 provides much better write performance than level 5
  - Level 5 requires at least 2 block reads and 2 block writes to write a single block, whereas Level 1 only requires 2 block writes
- Level 1 had higher storage cost than level 5
- Level 5 is preferred for applications where writes are sequential and large (many blocks), and need large amounts of data storage
- RAID 1 is preferred for applications with many random/small updates
- Level 6 gives better data protection than RAID 5 since it can tolerate two disk (or disk block) failures
  - Increasing in importance since latent block failures on one disk, coupled with a failure of another disk can result in data loss with RAID 1 and RAID 5.

Hardware Issues

- **Software RAID**: RAID implementations done entirely in software, with no special hardware support
- **Hardware RAID**: RAID implementations with special hardware
  - Use non-volatile RAM to record writes that are being executed
  - Beware: power failure during write can result in corrupted disk
    - E.g., failure after writing one block but before writing the second in a mirrored system
    - Such corrupted data must be detected when power is restored
      - Recovery from corruption is similar to recovery from failed disk
      - NV-RAM helps to efficiently detect potentially corrupted blocks
        - Otherwise all blocks of disk must be read and compared with mirror/parity block

Hardware Issues (Cont.)

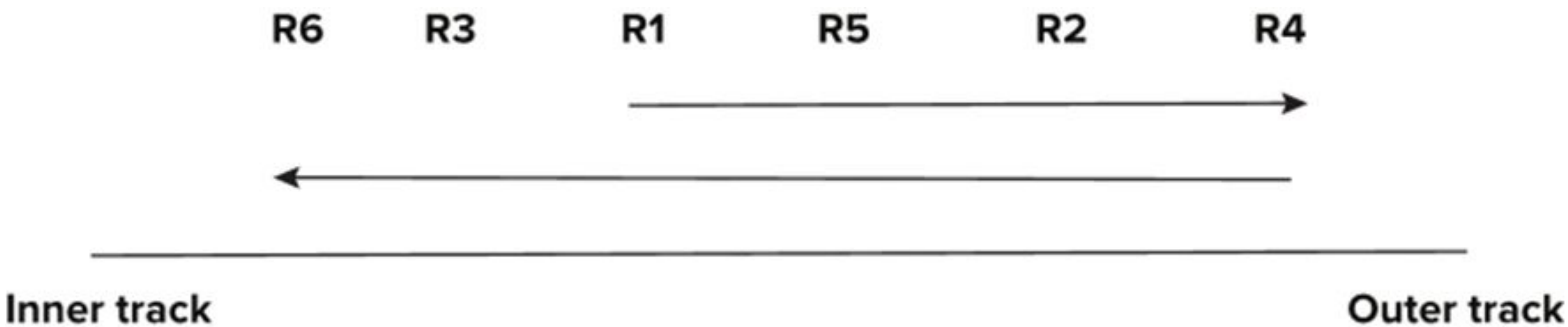
- **Latent failures**: data successfully written earlier gets damaged
  - can result in data loss even if only one disk fails
- **Data scrubbing**:
  - continually scan for latent failures, and recover from copy/parity
- **Hot swapping**: replacement of disk while system is running, without power down
  - Supported by some hardware RAID systems,
  - reduces time to recovery, and improves availability greatly
- Many systems maintain spare disks which are kept online, and used as replacements for failed disks immediately on detection of failure
  - Reduces time to recovery greatly
- Many hardware RAID systems ensure that a single point of failure will not stop the functioning of the system by using
  - Redundant power supplies with battery backup
  - Multiple controllers and multiple interconnections to guard against controller/interconnection failures

Overview

- Flash Memory
- RAID
- Dist-Block Access

Optimization of Disk-Block Access

- **Buffering:** in-memory buffer to cache disk blocks
- **Read-ahead:** Read extra blocks from a track in anticipation that they will be requested soon
- **Disk-arm-scheduling** algorithms re-order block requests so that disk arm movement is minimized
- **elevator algorithm**



Assignments

- Reading: Ch12.4-12.6
- Practice Excercises: 12.3, 12.4, 12.6, 12.7

Solutions to the Practice Excercises:
https://www.db-book.com/Practice-Exercises/index-solu.html

The End