

Chapter 12: Physical Storage Systems

Brief overview of Physical Storage Media for Databases, to be aware of its incidence on the design of DBMSs

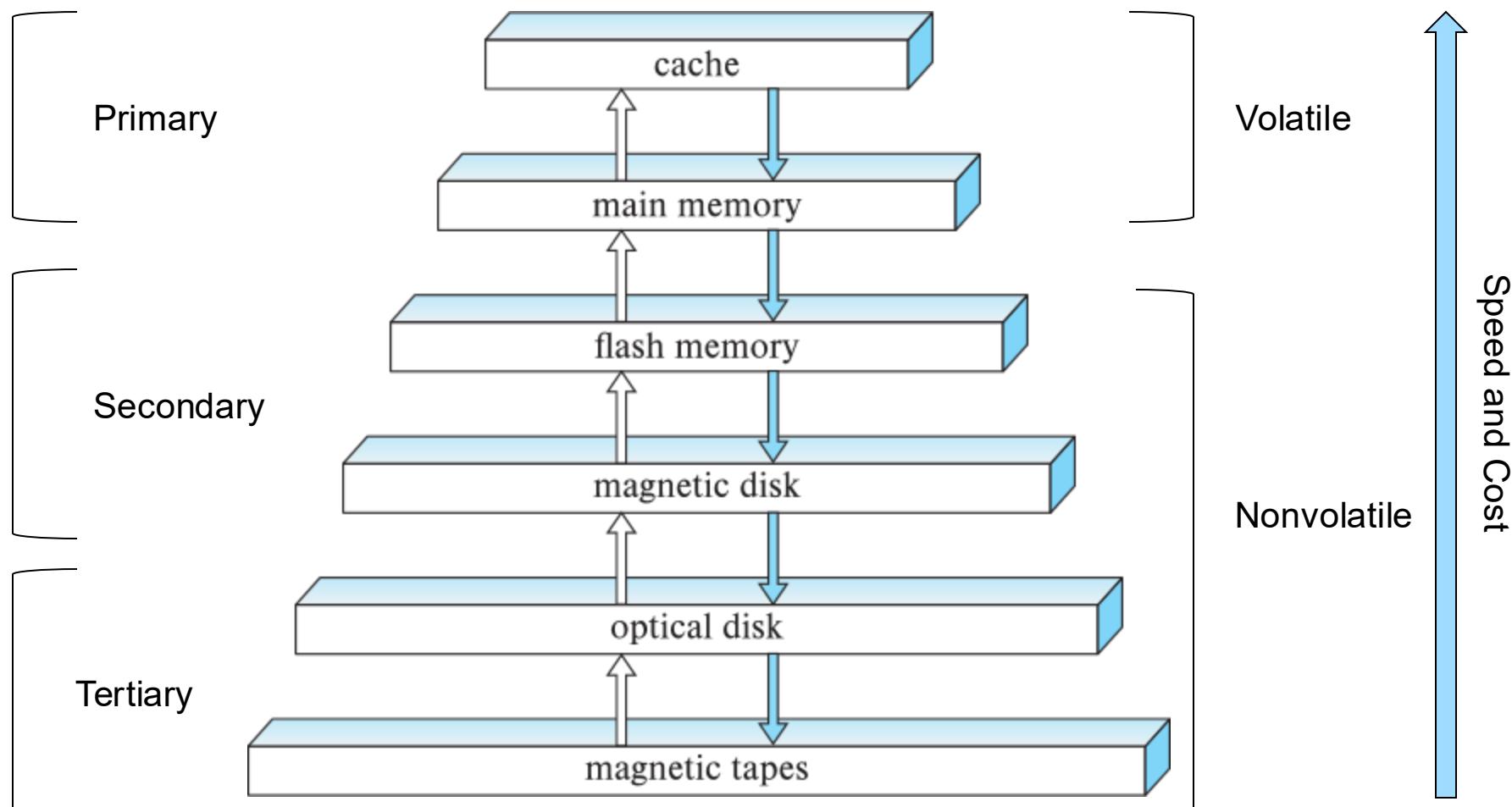
Sistemas de Bases de Dados 2024/25

Capítulo refere-se a: Database System Concepts, 7th Ed

Classification of Physical Storage Media

- In the end, a database must be physically stored in computer(s)
- Storage can be differentiated into:
 - **volatile storage:** loses contents when power is switched off
 - **non-volatile storage:**
 - Contents persist even when power is switched off.
 - Includes secondary and tertiary storage, as well as battery-backed up main-memory.
- Factors affecting choice of storage media include:
 - Speed with which data can be accessed
 - Cost per unit of data
 - Reliability

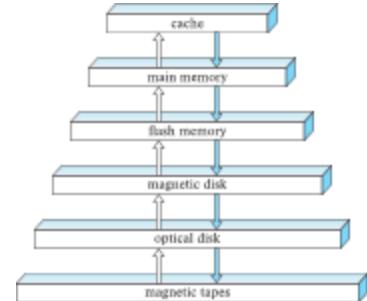
Storage Hierarchy



Storage Hierarchy (Cont.)

- **primary storage:** Fastest media but volatile (cache, main memory).
- **secondary storage:** next level in hierarchy, non-volatile, moderately fast access time
 - Also called **on-line storage**
 - E.g., flash memory, magnetic disks
- **tertiary storage:** lowest level in hierarchy, non-volatile, slow access time
 - also called **off-line storage** and used for **archival storage**
 - e.g., magnetic tape, optical storage
 - Magnetic tape
 - Sequential access, 1 to 12 TB capacity
 - A few drives with many tapes
 - Juke boxes with petabytes (1000's of TB) of storage

Storage Hierarchy (Cont.)



■ Cache and Main Memory

- Fast access (10s to 100s of nanoseconds)
- Generally, too small (or expensive) to store the entire database
- **Volatile**

■ Flash memory

- Non-volatile fast read-access memory
- Write is not as fast – and requires prior erasure
- Limited number of erasures
- Not yet with enough capacity or competitive price to be considered for really big databases

■ Magnetic disks

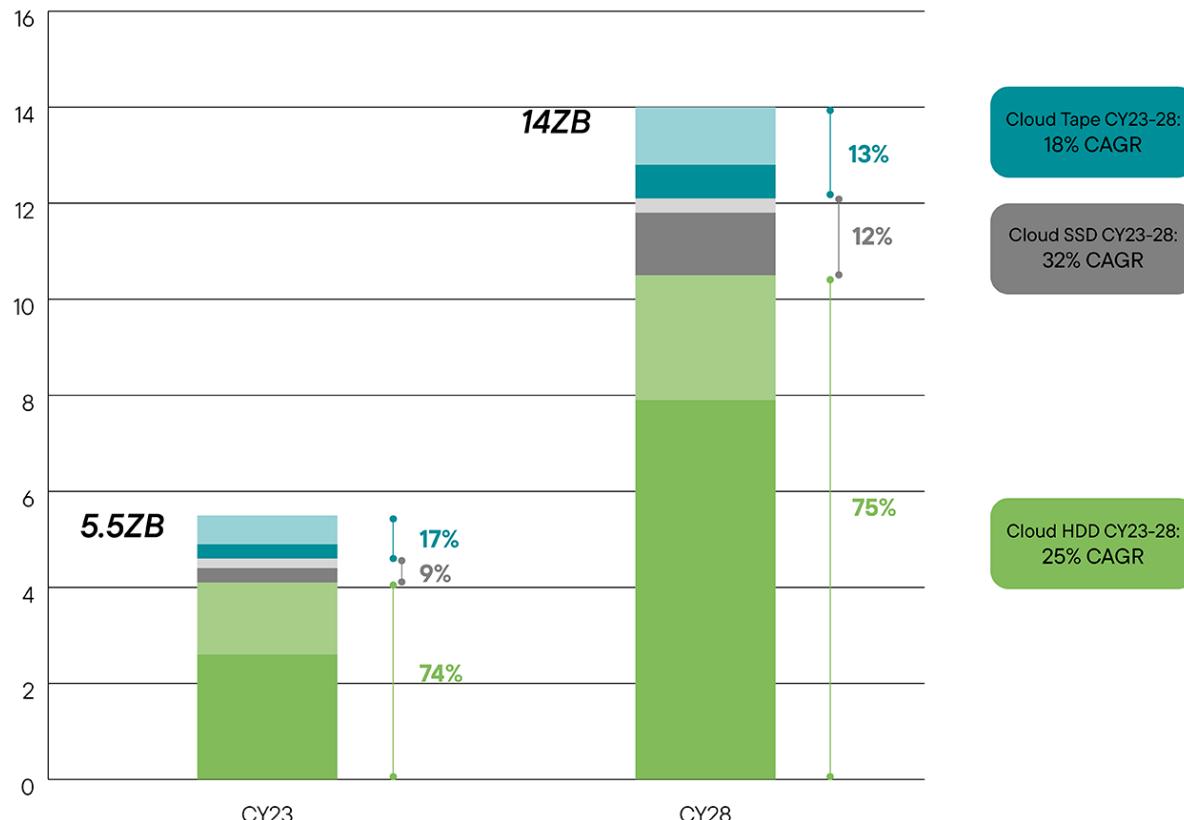
- Still the medium of choice for database storage
- Data moved to main memory for access, and written back to disk
- Reliable

■ Optical disks and Tapes

- Mainly used for backups (off-line/archival storage)
- Slow access time

Storage installed

Enterprise ZB Storage Installed Base Forecast by Storage Media



Source: Seagate analysis based on IDC Global StorageSphere Forecast, 2024-2028 Doc. #US52312824, June 2024.
Lighter colors represent enterprises' non-cloud exabytes and darker colors represent their cloud exabytes.

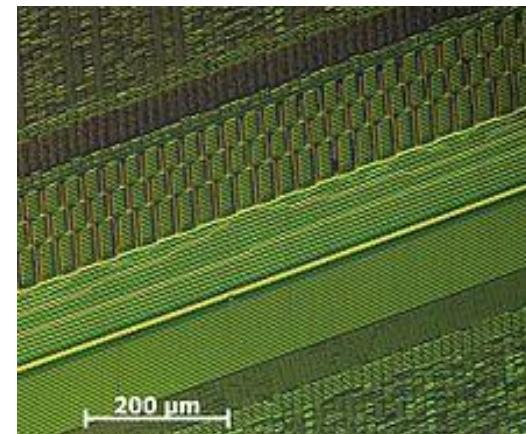
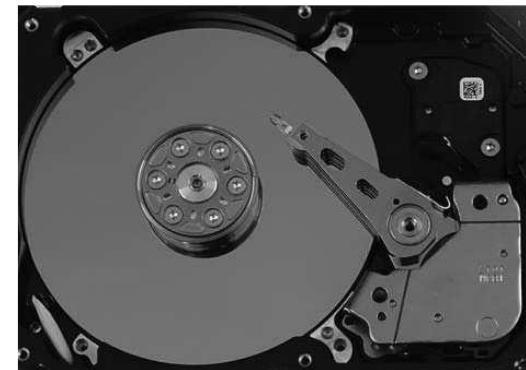
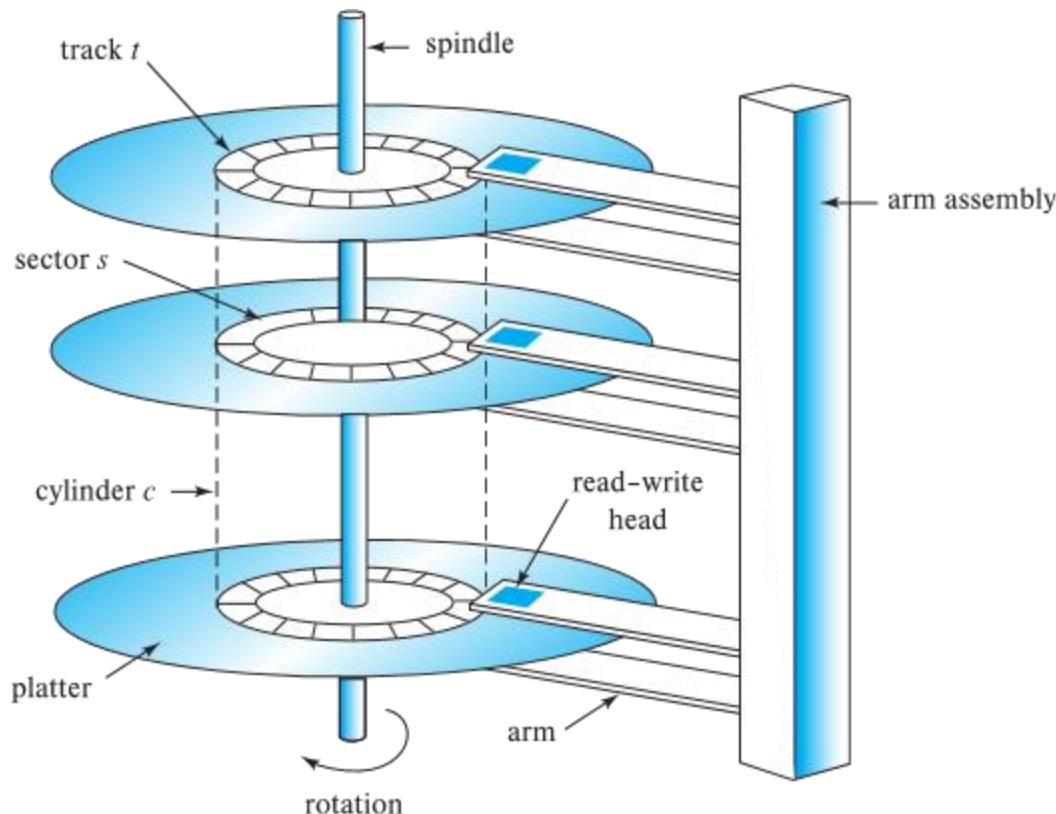
<https://www.seagate.com/as/en/blog/three-truths-about-hard-drives-and-ssds/>

Storage Interfaces

- Disk interface standards families
 - **SATA** (Serial ATA)
 - SATA 3 supports data transfer speeds of up to 6 gigabits/sec
 - **SAS** (Serial Attached SCSI)
 - SAS Version 3 supports 12 gigabits/sec
 - **NVMe** (Non-Volatile Memory Express) interface
 - Works with PCIe connectors to support lower latency and higher transfer rates
 - Supports data transfer rates of up to 24 gigabits/sec
- Disks usually connected directly to computer system
- In **Storage Area Networks (SAN)**, a large number of disks are connected by a high-speed network to a number of servers
- In **Network Attached Storage (NAS)** networked storage provides a file system interface using networked file system protocol, instead of providing a disk system interface

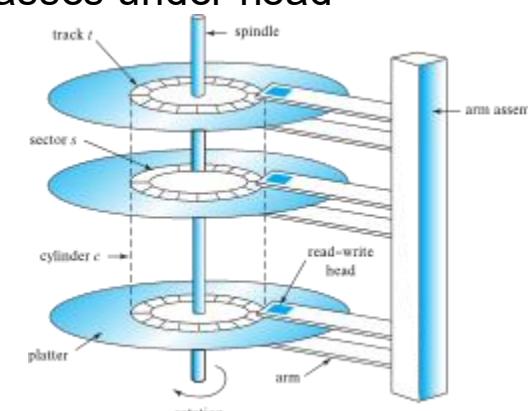
Magnetic Hard Disk Mechanism

- It is worth looking at how magnetic disks work
 - After all they are the place where most databases are stored!



Magnetic Disks

- **Read-write head**
 - Very close to platter surface, reads and writes magnetically encoded info
- Surface of platter divided into circular **tracks**
 - Over 50K-100K tracks per platter on typical hard disks
- Each track is divided into **sectors**.
 - A sector is the smallest unit of data that can be read or written.
 - Sector size is typically 512 bytes
 - Typical sectors per track: 500 to 1000 (on inner tracks) to 1000 to 2000 (on outer tracks)
- To read/write a sector
 - disk arm swings to position head on right track
 - platter spins continually; data is read/written as sector passes under head
- Head-disk assemblies
 - multiple disk platters on a single spindle (1 to 5 usually)
 - one head per platter, mounted on a common arm.
- **Cylinder i** consists of i^{th} track of all the platters



Magnetic Disks (Cont.)

- **Disk controller** – interfaces between the computer system and the disk drive hardware.
 - accepts high-level commands to read or write a sector
 - initiates actions such as moving the disk arm to the right track and actually reading or writing the data
 - Computes and attaches **checksums** to each sector to verify that data is read back correctly
 - If data is corrupted, with very high probability stored checksum won't match recomputed checksum
 - Ensures successful writing by reading back sector after writing it
 - Performs **remapping of bad sectors**

Performance Measures of Disks

- **Access time** – the time it takes from when a read or write request is issued to when data transfer begins. Consists of:
 - **Seek time** – time it takes to reposition the arm over the correct track.
 - Average seek time is 1/2 the worst case seek time.
 - 4 to 10 milliseconds on typical disks
 - **Rotational latency** – time it takes for the sector to be accessed to appear under the head.
 - 4 to 11 milliseconds on typical disks (5400 to 15000 r.p.m.)
 - Average latency is 1/2 of the above latency.
 - Overall latency is 5 to 20 msec depending on disk model
- **Data-transfer rate** – the rate at which data can be retrieved from or stored to the disk.
 - 25 to 200 MB per second max rate, lower for inner tracks

Performance Measures (Cont.)

- **Disk block** is a logical unit for storage allocation and retrieval
 - 4 to 16 kilobytes typically
 - Smaller blocks: more transfers from disk
 - Larger blocks: more space wasted due to partially filled blocks
- **Sequential access pattern**
 - Successive requests are for successive disk blocks
 - Disk seek required only for first block
 - (if the blocks are stored successively in the disk)
- **Random access pattern**
 - Successive requests are for blocks that can be anywhere on disk
 - Each access requires a seek
 - Transfer rates are low since a lot of time is wasted in seeks
- **I/O operations per second (IOPS)**
 - Number of random block reads that a disk can support per second
 - 50 to 200 IOPS on current generation magnetic disks

Performance Measures (Cont.)

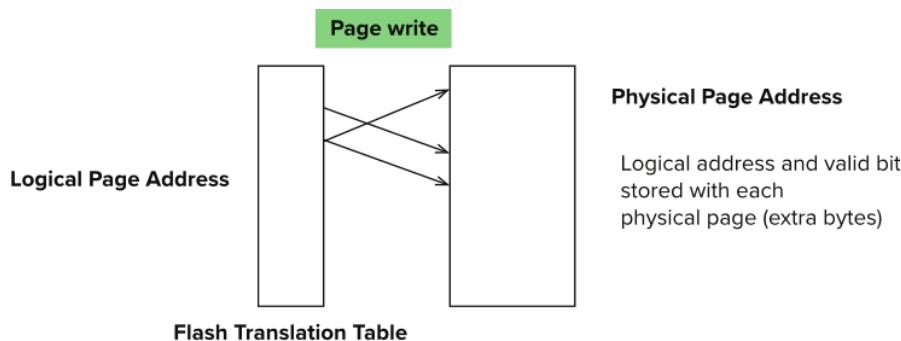
- **Mean time to failure (MTTF)** – the average time the disk is expected to run continuously without any failure.
 - Typically 3 to 5 years
 - Probability of failure of new disks is quite low, corresponding to a “theoretical MTTF” of 500,000 to 1,200,000 hours for a new disk
 - E.g., an MTTF of 1,200,000 hours for a new disk means that given 1000 relatively new disks, on an average one will fail every 1200 hours
 - MTTF decreases as disk ages

Flash Storage

- NOR flash vs NAND flash
- NAND flash
 - used widely for storage, cheaper than NOR flash
 - requires page-at-a-time read (page: 512 bytes to 4 KB)
 - 20 to 100 microseconds for a page read
 - Not much difference between sequential and random read
 - Page can only be written once
 - Must be erased to allow rewrite
- **Solid state disks**
 - Use standard block-oriented disk interfaces, but store data on multiple flash storage devices internally
 - Transfer rate of up to 500 MB/sec using SATA, and up to 3 GB/sec using NVMe PCIe

Flash Storage (Cont.)

- Erase happens in units of **erase block**
 - Takes 2 to 5 millisecs
 - Erase block typically 256 KB to 1 MB (128 to 256 pages)
- **Remapping** of logical page addresses to physical page addresses avoids waiting for erase
- **Flash translation table** tracks mapping
 - also stored in a label field of flash page
 - remapping carried out by **flash translation layer**



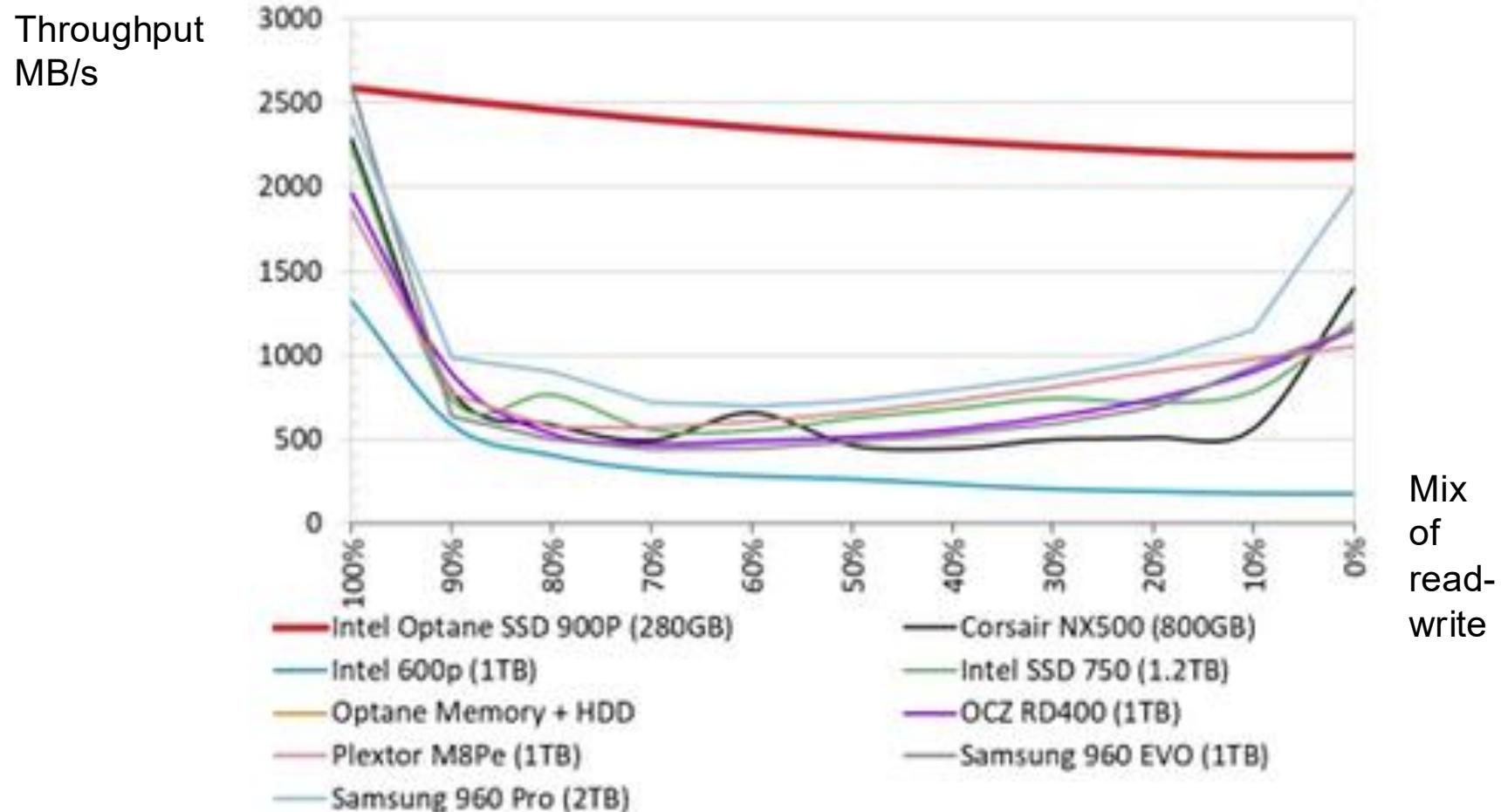
- After 100,000 to 1,000,000 erases, erase block becomes unreliable and cannot be used

SSD Performance Metrics

- 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

3D-XPoint

- Discontinued... But it was good!



https://en.wikipedia.org/wiki/3D_XPoint

Project Silica

- Store data in glass using lasers
 - 7TB of capacity in the size of a DVD
 - Store data for thousands of years



RAID

- **RAID: Redundant Arrays of Independent Disks**
 - disk organization techniques that manage a large numbers of disks, providing a view of a single disk of
 - **high capacity** and **high speed** by using multiple disks in parallel,
 - **high reliability** by storing data redundantly, so that data can be recovered even if a disk fails
- 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

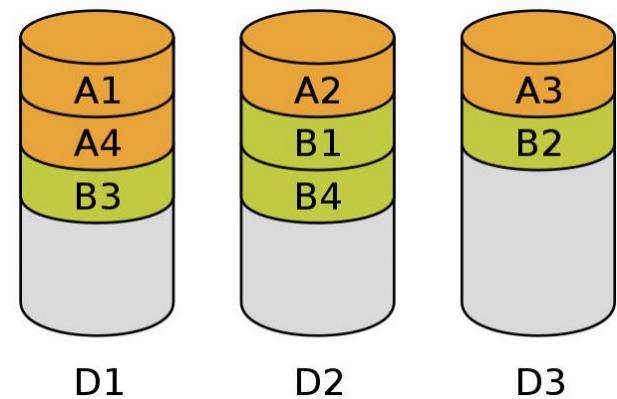


Improvement of Reliability via Redundancy

- **Redundancy** – store extra information that can be used to rebuild information lost in a disk failure
- E.g., **Mirroring** (or **shadowing**)
 - Duplicate every disk. Logical disk consists of two physical disks.
 - 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 a disk fails, and its mirror disk also fails before the system is repaired
 - Probability of combined event is very small
 - Except for dependent failure modes such as fire or building collapse or electrical power surges
- **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)

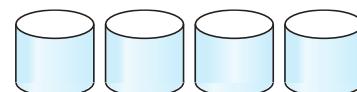
Improvement in Performance via Parallelism

- 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.
- **Block-level striping** – with n disks, block i of a file goes to disk $(i \bmod n) + 1$
 - 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
- **Bit-level striping** – split the bits of each byte across multiple disks (not used much anymore)
 - 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
 - Not used anymore

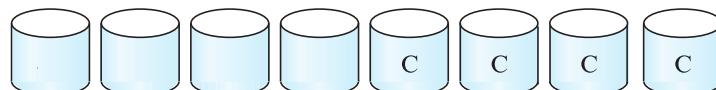


RAID Levels

- 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



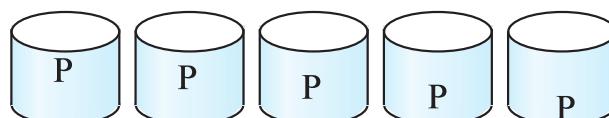
(b) RAID 1: mirrored disks

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 since the parity blocks can be computed from the sequence being written
 - To recover data for a block, compute XOR of bits from all other blocks in the set including the parity block

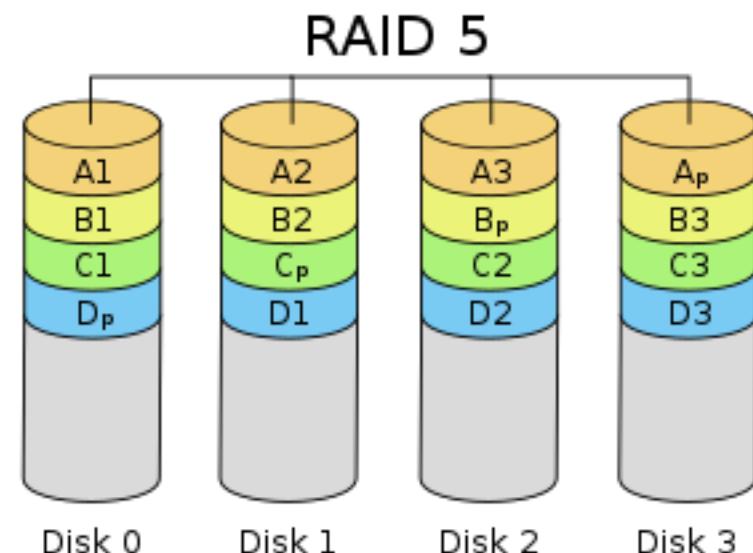
RAID Levels (Cont.)

- **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 $(n \bmod 5) + 1$, with the data blocks stored on the other 4 disks.



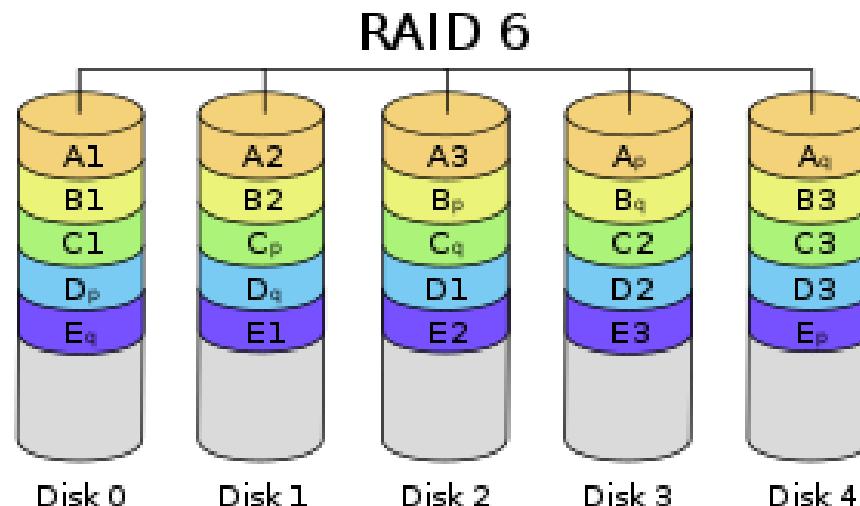
(c) RAID 5: block-interleaved distributed parity

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



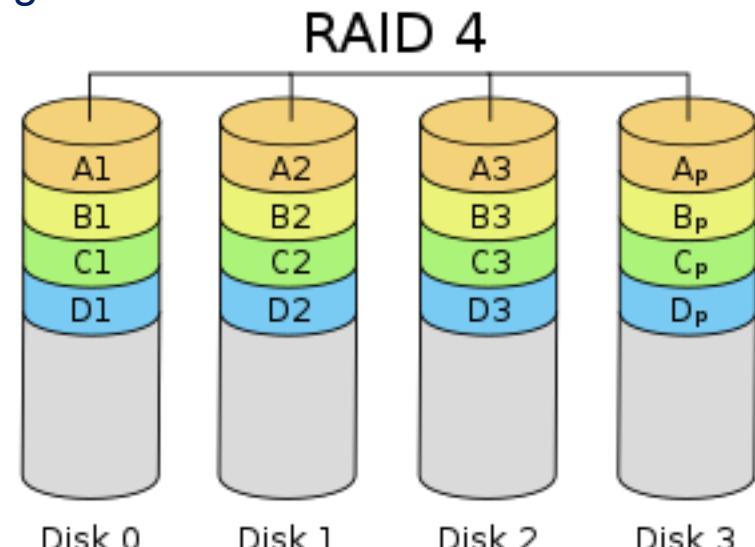
RAID Levels (Cont.)

- **RAID Level 5 (Cont.)**
 - Block writes occur in parallel if the blocks and their parity blocks are on different disks.
- **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.
 - Better reliability than Level 5 at a higher cost
 - Becoming more important as storage sizes increase



RAID Levels (Cont.)

- Other levels (not used in practice):
 - **RAID Level 2:** Memory-Style Error-Correcting-Codes (ECC) with bit striping.
 - **RAID Level 3:** Bit-Interleaved Parity
 - **RAID Level 4:** Block-Interleaved Parity; uses block-level striping, and keeps a parity block on a separate **parity disk** for corresponding blocks from N other disks.
 - RAID 5 is better than RAID 4, since with RAID 4 with random writes, parity disk gets much higher write load than other disks and becomes a bottleneck



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

Entry level server

Processador

AMD EPYC 4244P - 6c/12t - 3.8GHz/5.1GHz 2024

-15,00 €

+ IVA/mês

AMD EPYC 4245P - 6c/12t - 3.9GHz/5.4GHz 2026

Included

Memória

32GB DDR5 ECC On-Die 5600MHz

+22,00 €

+ IVA/mês

64GB DDR5 ECC On-Die 5600MHz

+44,00 €

+ IVA/mês

128GB DDR5 ECC On-Die 5600MHz

+63,00 €

+ IVA/mês

Armazenamento

2x 960GB SSD NVMe Soft RAID

+21,60 €

+ IVA/mês

4x 960GB SSD NVMe Soft RAID

+38,00 €

+ IVA/mês

2x 960GB SSD NVMe + 2x 1.92TB SSD NVMe Soft RAID

+54,40 €

+ IVA/mês

2x 960GB SSD NVMe + 2x 3.84TB SSD NVMe Soft RAID

+100,00 €

+ IVA/mês

2x 960GB SSD NVMe + 2x 7.68TB SSD NVMe Soft RAID

+149,20 €

+ IVA/mês

4x 3.84TB SSD NVMe Soft RAID

+178,40 €

+ IVA/mês

4x 7.68TB SSD NVMe Soft RAID

+298,40 €

+ IVA/mês

NO HDD CONFIGURATIONS (JUST FOR STORAGE)

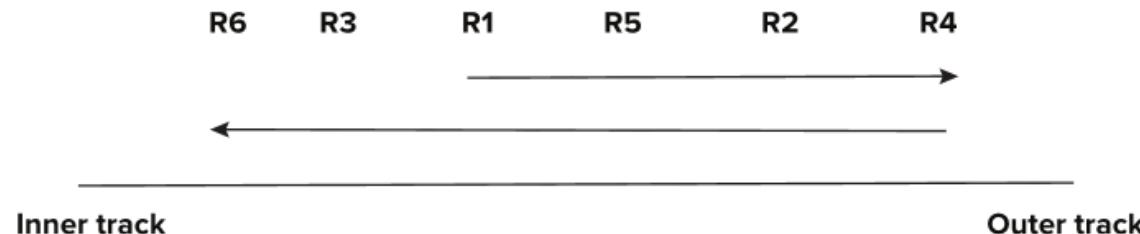
Base 99.99 €/month

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

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**



Optimization of Disk Block Access (Cont.)

- **File organization**

- Allocate blocks of a file in as contiguous a manner as possible
- Allocation in units of **extents** (sequences of contiguous blocks)
- Files may get **fragmented**
 - E.g., if free blocks on disk are scattered, and newly created file has its blocks scattered over the disk
 - Sequential access to a fragmented file results in increased disk arm movement
 - Some systems have utilities to **defragment** the file system, in order to speed up file access

- **Non-volatile write buffers**

- Some recovery algorithms require writing blocks by order, and so without non-volatile write buffers performance is substantially impacted.

End of Chapter 12