- Page-at-a-time read/write, multi-page erase
- High capacity (several GB)

- Widely used as data storage mechanism in portable devices

Magnetic-disk

- Data is stored on spinning disk, and read/written magnetically
- Primary medium for the long-term storage of data; typically stores entire database.
- Data must be moved from disk to main memory for access, and written back for storage
- **direct-access** – possible to read data on disk in any order, unlike magnetic tape
- Survives power failures and system crashes
  - disk failure can destroy data: is rare but does happen

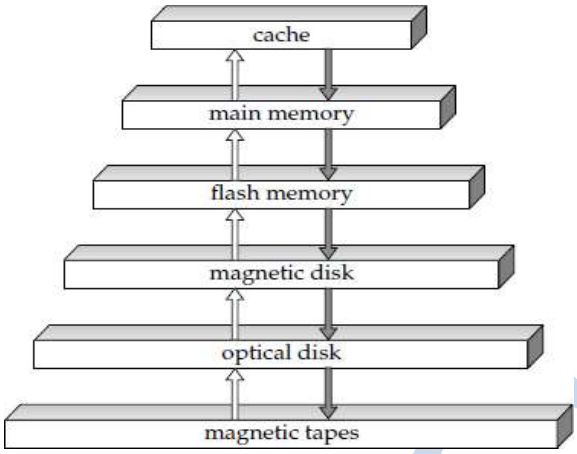
Optical storage

- non-volatile, data is read optically from a spinning disk using a laser
- CD-ROM (640 MB) and DVD (4.7 to 17 GB) most popular forms
- Write-one, read-many (WORM) optical disks used for archival storage (CD-R, DVD-R, DVD+R)
- Multiple write versions also available (CD-RW, DVD-RW, DVD+RW, and DVD-RAM)
- Reads and writes are slower than with magnetic disk
- Juke-box systems, with large numbers of removable disks, a few drives, and a mechanism for automatic loading/unloading of disks available for storing large volumes of data

Tape storage

- non-volatile, used primarily for backup (to recover from disk failure), and for archival data
- sequential-access–much slower than disk
- very high capacity (40 to 300 GB tapes available)
- tape can be removed from drive storage costs much cheaper than disk, but drives are expensive
- Tape jukeboxes available for storing massive amounts of data
- hundreds of terabytes (1 terabyte = 109 bytes) to even a petabyte(1 petabyte= 1012bytes)

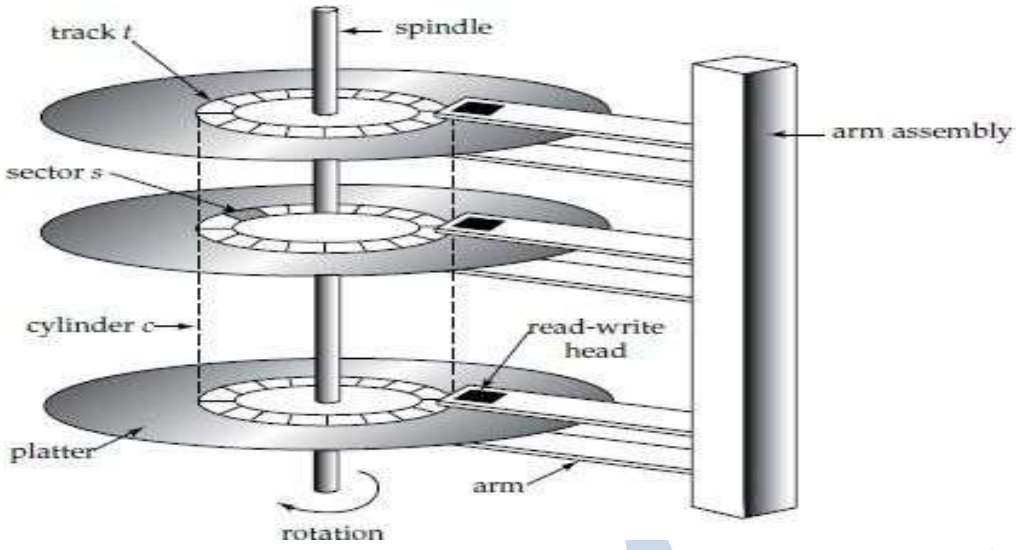
Storage Hierarchy



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**
E.g. magnetic tape, optical storage

Magnetic disks

- Positioned very close to the platter surface (almost touching it)
- Reads or writes magnetically encoded information.
- Surface of platter divided into circular **tracks**
- Over 50K-100K tracks per platter on typical hard disks
- Each track is divided into **sectors.**
- Sector size typically 512 bytes
- Typical sectors per track: 500 (on inner tracks) to 1000 (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 .Earlier generation disks were susceptible to "head-crashes "leading to loss of all data on disk

Current generation disks are less susceptible to such disastrous failures, but individual sectors may get corrupted

Disk controller–interfaces

A disk controller interfaces between the computer system and the actual hardware of the disk drive. It accepts high-level commands to read or write a sector, and initiates actions, such as moving the disk arm to the right track and actually reading or writing the data. Disk controllers also attach checksums to each sector that is written, the checksum is computed from the data written to the sector

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.
- ✓ –Would be 1/3 if all tracks had the same number of sectors, and we ignore the time to start and stop arm movement
- ✓ 4 to 10 milliseconds on typical disks

Rotational latency–time it takes for the sector to be accessed to appear under the head.

- Average latency is 1/2 of the worst case latency.
- 4 to 11 milliseconds on typical disks (5400 to 15000 r.p.m.)

Data-transfer rate–the rate at which data can be retrieved from or stored to the disk.25 to 100 MB per second max rate, lower for inner tracks. Multiple disks may share a controller, so rate that controller can handle is also important

E.g. ATA-5: 66 MB/sec, SATA: 150 MB/sec, Ultra 320 SCSI: 320 MB/s
Fiber Channel (FC2Gb): 256 MB/s

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

Optimization of Disk

Disk-Block Access
Block –a contiguous sequence of sectors from a single track data is transferred between disk and main memory in blocks
Typical block sizes today range from 4 to 16 kilobytes

Disk-arm-scheduling algorithms order pending accesses to tracks so that disk arm movement is minimized
Elevator algorithm: move disk arm in one direction (from outer to inner tracks or vice versa), processing next request in that direction, till no more requests in that direction, then reverse direction and repeat

File organization–optimize block access time by organizing the blocks to correspond to how data will be accessed
E.g. Store related information on the same or nearby blocks/cylinders.
File systems attempt to allocate contiguous chunks of blocks (e.g. 8 or 16 blocks) to a file

Files may get **fragmented** over time
E.g. if data is inserted to/deleted from the file.
Or 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

Nonvolatile write buffers speed up disk writes by writing blocks to a non-volatile RAM buffer immediately

- ✓ Non-volatile RAM: battery backed up RAM or flash memory
- ✓ Even if power fails, the data is safe and will be written to disk when power returns
- ✓ Controller then writes to disk whenever the disk has no other requests or request has been pending for some time
- ✓ Database operations that require data to be safely stored before continuing can continue without waiting for data to be written to disk

□ *Writes can be reordered to minimize disk arm movement*

Log disk–a disk devoted to writing a sequential log of block updates

- ✓ Used exactly like nonvolatile RAM
- ✓ Write to log disk is very fast since no seeks are required
- ✓ No need for special hardware (NV-RAM)
- ✓ File systems typically reorder writes to disk to improve performance

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 speedby using multiple disks in parallel, and 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)

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*106hours (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

- But seek/access time worse than for a single disk
- Bit level striping is not used much any more

Block-level striping–with *n*disks, block *i*of a file goes to disk (*i*mod *n*) + 1

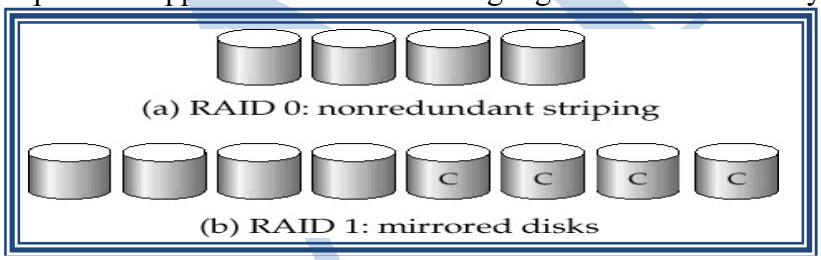
- Requests for different blocks can run in parallel if the blocks reside on different disks
- A request for a long sequence of blocks can utilize all disks in parallel

RAID levels

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 lost is not critical.

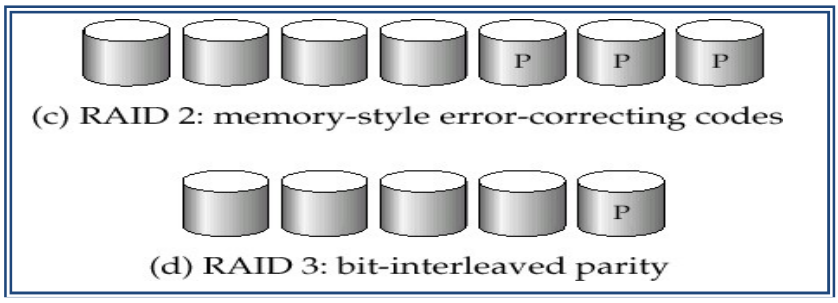
RAID Level 1: Mirrored disks with block striping
Offers best write performance.
Popular for applications such as storing log files in a database system.



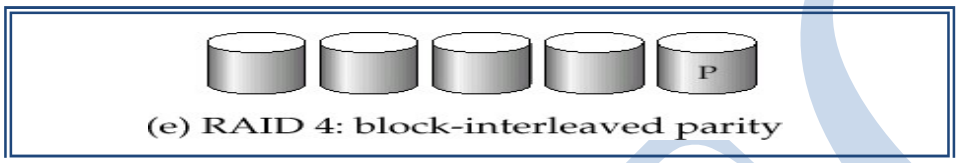
RAID Level 2: Memory-Style Error-Correcting-Codes (ECC) with bit striping.

RAID Level 3: Bit-Interleaved Parity

- a single parity bit is enough for error correction, not just detection
- When writing data, corresponding parity bits must also be computed and written to a parity bit disk
- To recover data in a damaged disk, compute XOR of bits from other disks (including parity bit
- Faster data transfer than with a single disk, but fewer I/Os per second since every disk has to participate in every I/O.



RAID Level 4: Block-Interleaved Parity; uses block-level striping, and keeps a parity block on a separate disk for corresponding blocks from *N* other disks.



- □When writing data block, corresponding block of parity bits must also be computed and written to parity disk
- □To find value of a damaged block; compute XOR of bits from corresponding blocks (including parity block) from other disks.

- Provides higher I/O rates for independent block reads than Level3
- block read goes to a single disk, so blocks stored on different disks can be read in parallel
- Before writing a block, parity data must be computed
- 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 re computing the parity value using the new values of blocks corresponding to the parity block
- More efficient for writing large amounts of data sequentially
- Parity block becomes a bottleneck for independent block writes since every block write also writes to parity disk

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.

