Mass-Storage Systems

Mass-Storage Systems

- ✓ **Overview of Mass Storage Structure**
- ✓ **Disk Structure**
- ✓ **Storage Attachment**
- ✓ **Disk Scheduling**
- ✓ **Storage Management**
- ✓ **Swap-Space Management**
- ✓ **RAID Structure**

Objectives

- ✓ **Describe the physical structure of secondary storage devices and the resulting effects on the uses of the devices**
- ✓ **Explain the performance characteristics of mass-storage devices**
- ✓ **Evaluate disk scheduling algorithms**
- ✓ **Discuss operating-system services provided for mass storage, including RAID(*Redundant Array of Independent (or Inexpensive) Disks***

Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of computers
  - Drives rotate at 60 to 250 times per second
  - **Transfer rate** is rate at which data flow between drive and computer
  - **Positioning time (random-access time)** is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
  - **Head crash** results from disk head making contact with the disk surface
- Disks can be removable

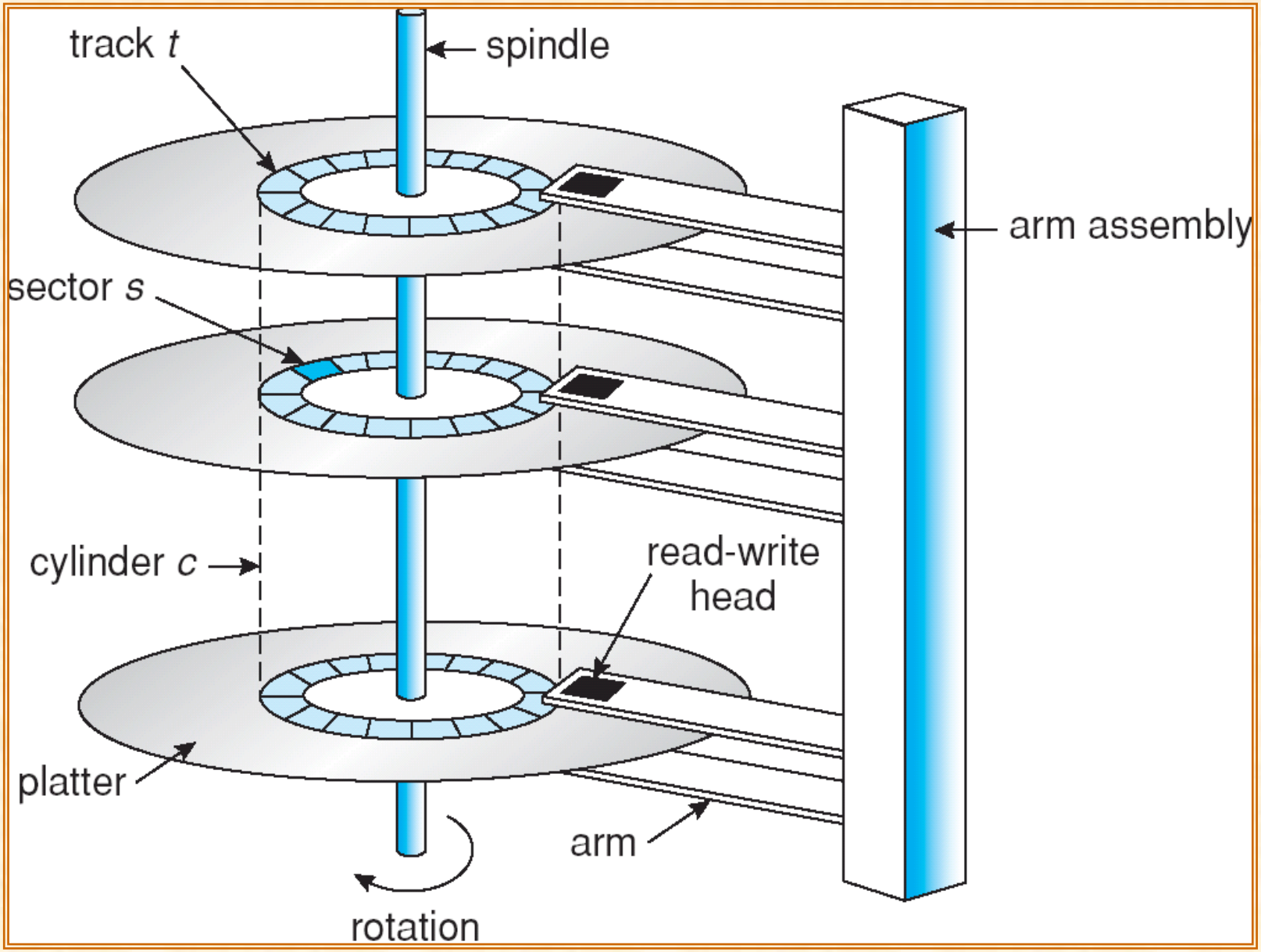
Overview of Mass Storage Structure

- Drive attached to computer via I/O bus. Busses vary, including
  - IDE (Integrated Drive Electronics),
  - EIDE (Enhanced Integrated Drive Electronics),
  - ATA (Advanced Technology Attachment),
  - SATA (Serial Advanced Technology Attachment),
  - USB (Universal Serial Bus),
  - FC (Fibre Channel),
  - SCSI (Small Computer System Interface)
  - Serial Attached SCSI (SAS)
- Host controller in computer uses bus to talk to disk controller built into drive or storage array

Overview of Mass Storage Structure

- **Magnetic tape**
  - **Was early secondary-storage medium**
  - **Relatively permanent and holds large quantities of data**
  - **Access time slow**
  - **Random access ~1000 times slower than disk**
  - **Mainly used for backup, storage of infrequently-used data, transfer medium between systems**
  - **Once data under head, transfer rates comparable to disk**
  - **20-400GB typical storage**
  - **Common technologies are 4mm, 8mm, 19mm, LTO-2 (Linear Tape-Open) and SDLT (Super Digital Linear Tape)**

Moving-head Disk Mechanism



Hard Disk Drives

- Platters range from .85” to 14” (historically)
  - Commonly 3.5”, 2.5”, and 1.8”
- Range from 30GB to 3TB per drive
- Performance
  - Transfer Rate – theoretical – 6 Gb/sec
  - Effective Transfer Rate – real – 1Gb/sec

Hard Disk Drives

- **Performance**
  - **Seek time from 3ms to 12ms – 9ms common for desktop drives**
  - **Average seek time measured or calculated based on 1/3 of tracks**
  - **Latency based on spindle speed**
    - **1 / (RPM / 60) = 60 / RPM**
  - **Average latency = ½ latency**

Hard Disk Performance

- **Access Latency = Average access time = average seek time + average latency**
  - **For fastest disk 3ms + 2ms = 5ms**
  - **For slow disk 9ms + 5.56ms = 14.56ms**
- **Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead**
- **For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a .1ms controller overhead =**
  - **5ms + 4.17ms + 0.1ms + transfer time =**
  - **Transfer time = 4KB / 1Gb/s * 8Gb / GB * 1GB / 1024^2KB = 32 / (1024^2) = 0.031 ms**
  - **Average I/O time for 4KB block = 9.27ms + .031ms = 9.301ms**

Disk Structure and Address Mapping

- **Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.**
- **The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.**
  - **Sector 0 is the first sector of the first track on the outermost cylinder.**
  - **Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost**
  - **Logical to physical address should be easy**
    - **Except for bad sectors**
    - **Non-constant # of sectors per track via constant angular velocity**

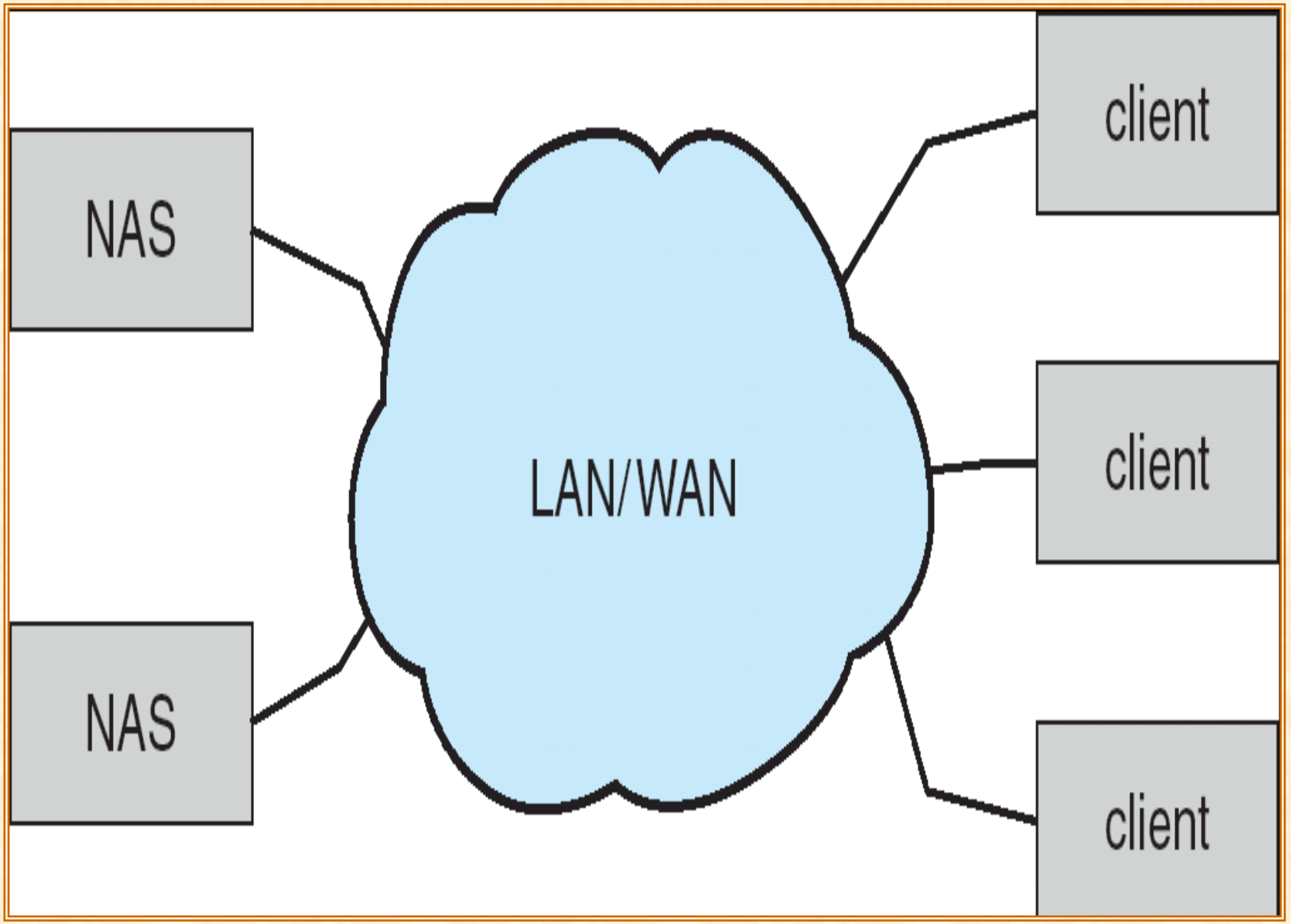
Storage Attachment

- Computers access storage in three ways
  - host-attached
  - network-attached
  - cloud
- Host attached access through local I/O ports, using one of several technologies
  - To attach many devices, use storage busses such as USB, firewire, thunderbolt
  - High-end systems use fibre channel (FC)
    - High-speed serial architecture using fibre or copper cables
    - Multiple hosts and storage devices can connect to the FC fabric

Network-Attached Storage

- Network-attached storage (NAS) is storage made available over a network rather than over a local connection (such as a bus)
  - Remotely attaching to file systems
- NFS and CIFS are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network
- iSCSI protocol uses IP network to carry the SCSI protocol
  - Remotely attaching to devices (blocks)

Network-Attached Storage



Cloud Storage

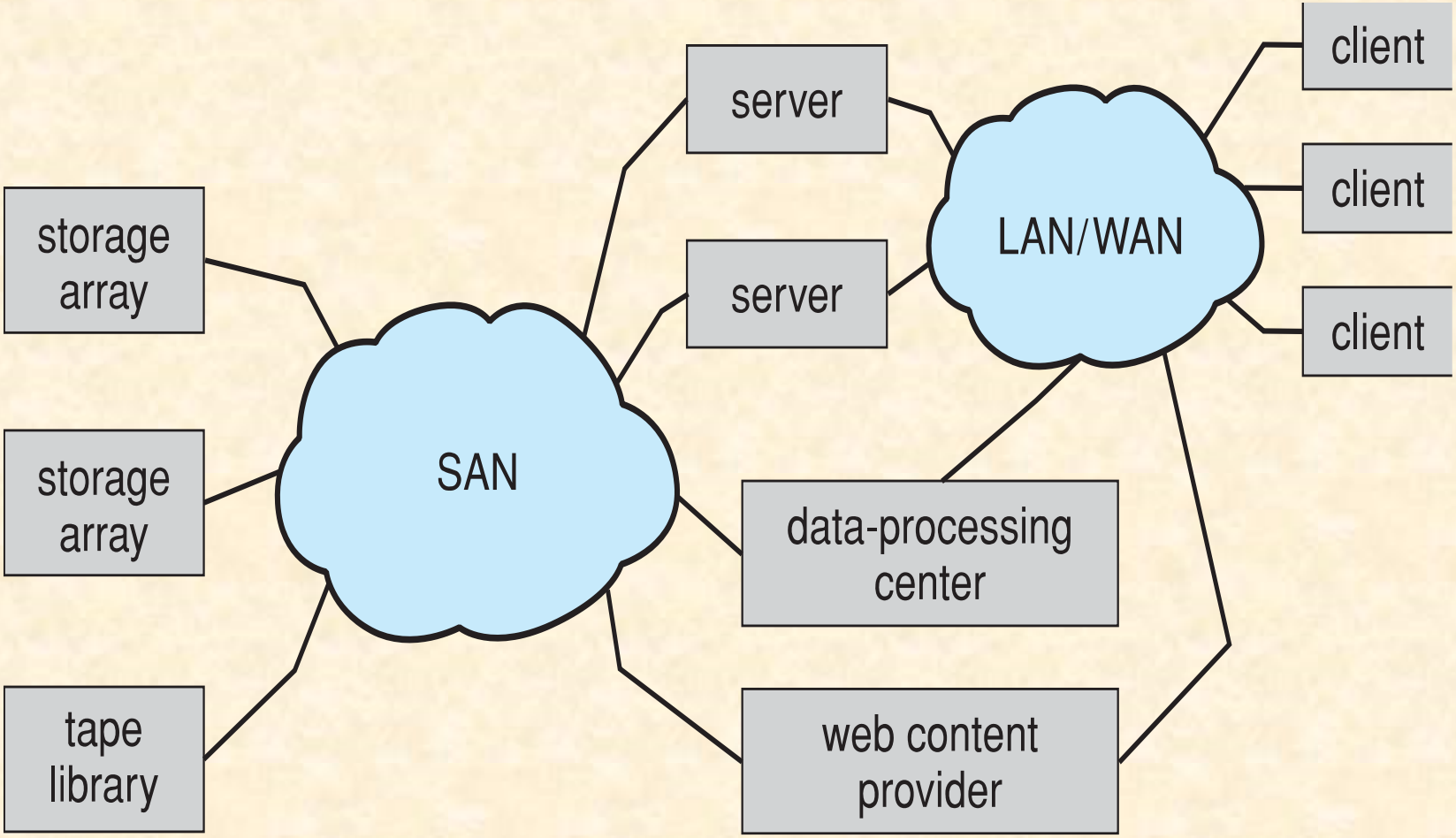
- Similar to NAS, provides access to storage across a network
  - Unlike NAS, accessed over the Internet or a WAN to remote data center
- NAS presented as just another file system, while cloud storage is API based, with programs using the APIs to provide access
  - Examples include Dropbox, Amazon S3, Microsoft OneDrive, Apple iCloud
  - Use APIs because of latency and failure scenarios (NAS protocols wouldn't work well)

Storage Array

- Can just attach disks, or arrays of disks
- Avoids the NAS drawback of using network bandwidth
- Storage Array has controller(s), provides features to attached host(s)
  - Ports to connect hosts to array
  - Memory, controlling software (sometimes NVRAM, etc)
  - A few to thousands of disks
  - RAID, hot spares, hot swap (discussed later)
  - Shared storage -> more efficiency
  - Features found in some file systems
    - Snaphots, clones, thin provisioning, replication, deduplication, etc

Storage Area Network (SAN)

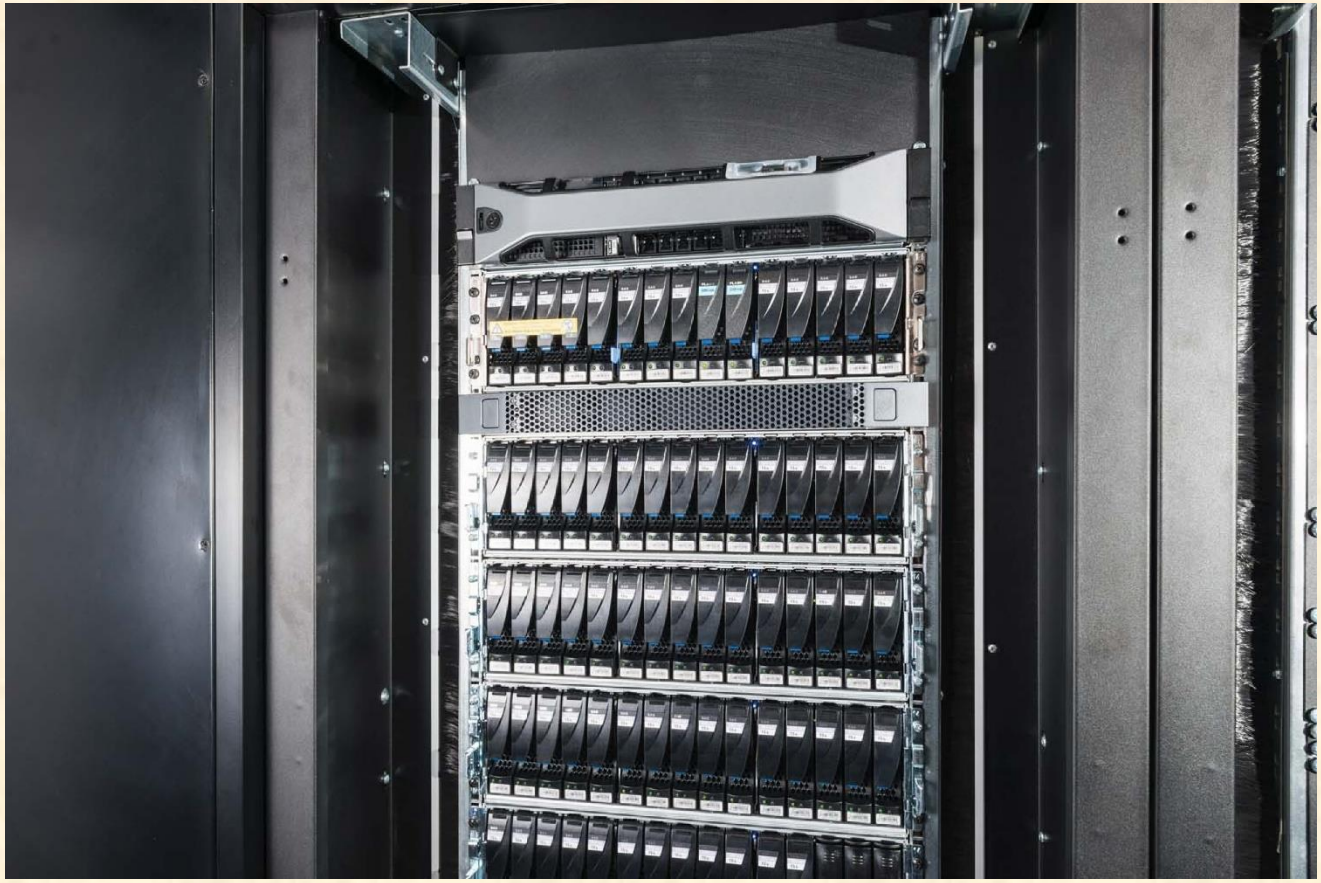
- ✓ **Common in large storage environments (and becoming more common)**
- ✓ **Multiple hosts attached to multiple storage arrays - flexible**



Storage Area Network (Cont.)

- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches or InfiniBand (IB) network
- Hosts also attach to the switches
- Storage made available via LUN Masking from specific arrays to specific servers
- Easy to add or remove storage, add new host and allocate it storage
- Why have separate storage networks and communications networks?
  - Consider iSCSI, FCOE

Storage Area Network (Cont.)



A Storage Array

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Disk speed has three parts
  - **Seek time** is the time for the disk arm to move the heads to the cylinder containing the desired sector.
  - **Rotational latency** is the additional time waiting for the disk to rotate the desired sector to the disk head.
  - **Transfer time** is the actual transfer of data between the disk and main memory

Disk Scheduling

- ✓ **Total time to service a disk request is =**

  seek time + latency time + transfer time

- ✓ **Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.**

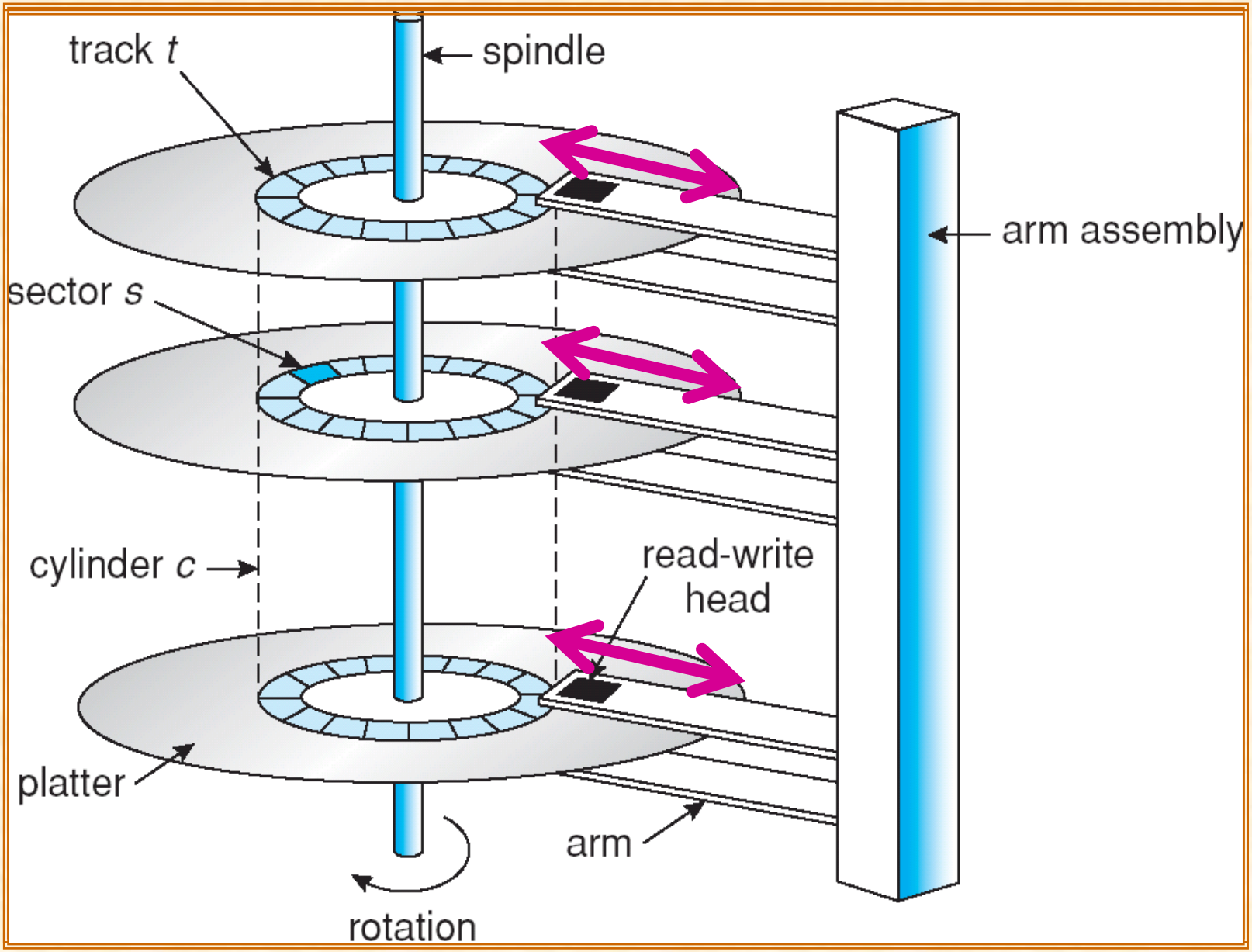
Several algorithms exist to schedule the servicing of disk I/O requests.

1. **FCFS scheduling**
2. **SSTF scheduling**
3. **SCAN scheduling**
4. **C-SCAN scheduling**
5. **LOOK scheduling**
6. **C-LOOK scheduling**

FCFS - First-Come, First-Served

- **FCFS scheduling: Requests serviced in order of arrival**
  - **Advantages**
    - **Fair**
    - **Prevents indefinite postponement**
    - **Low overhead**
  - **Disadvantages**
    - **Potential for extremely low throughput**
      - **FCFS typically results in a random seek pattern because it does not reorder requests to reduce service delays**

Disk Read Write head movement



FCFS

Queue: 98, 183, 37, 122, 14, 124, 65, 67. Head pointer 53

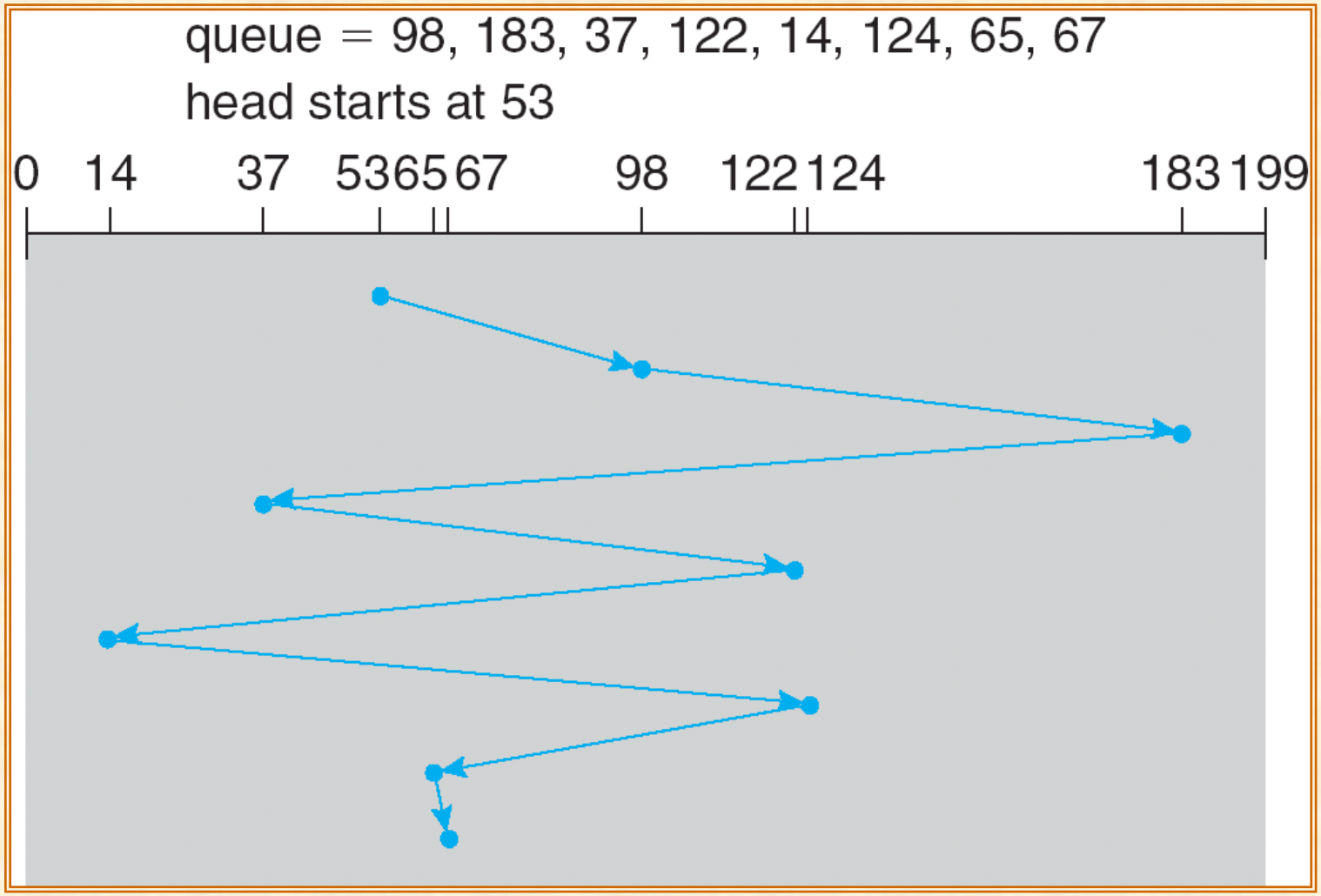


Illustration shows total head movement of 640 cylinders (45+85+146+85+108+110+59+2)

SSTF – Shortest Seek Time First

- Selects the request with the minimum seek time from the current head position.
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.
- Advantages
  - Higher throughput and lower response times than FCFS
  - Reasonable solution for batch processing systems
- Disadvantages
  - Does not ensure fairness
  - Possibility of indefinite postponement
  - High variance of response times
  - Response time generally unacceptable for interactive systems

SSTF

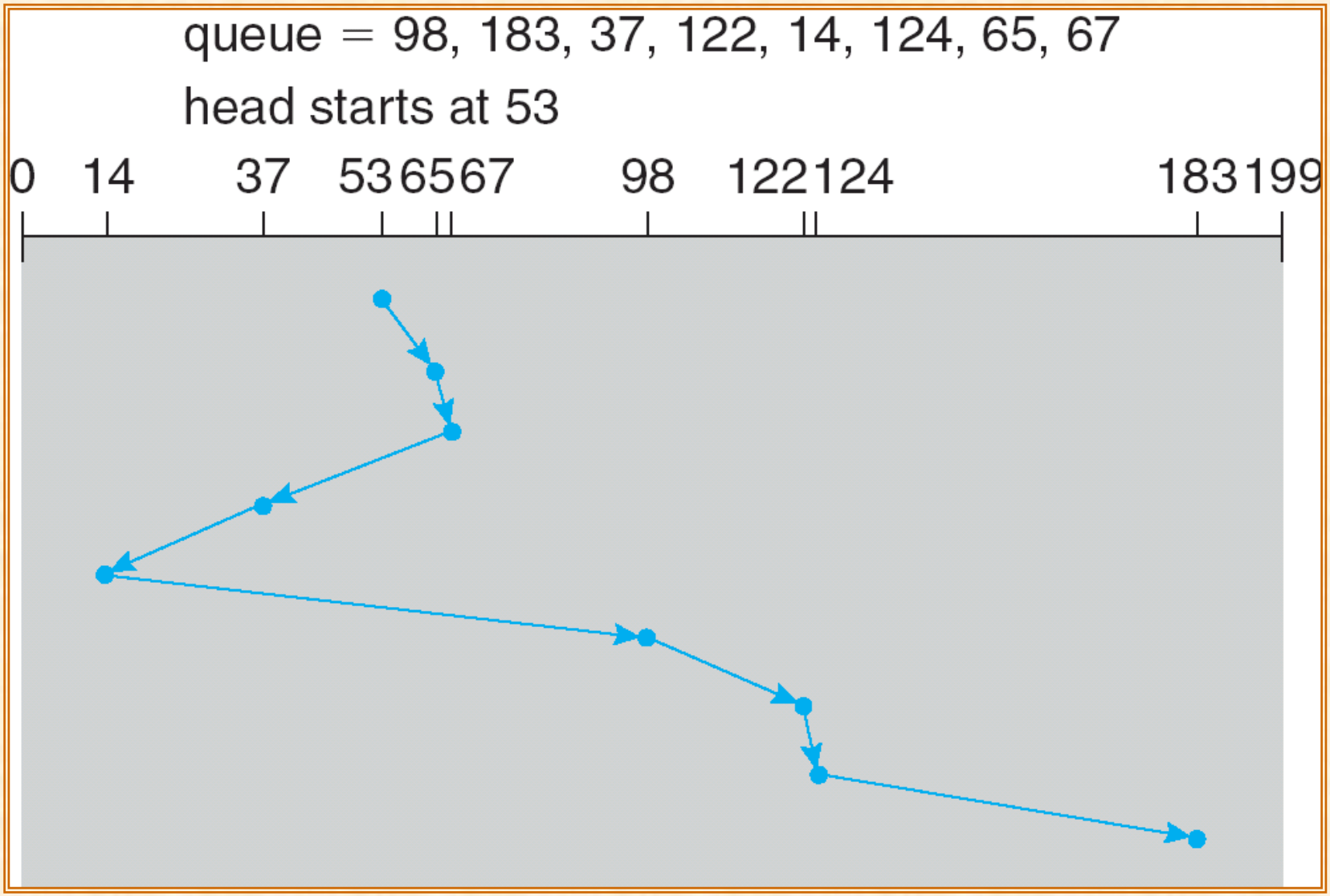


Illustration shows total head movement of 236 cylinders

SCAN

- ✓ **Recognition of the dynamic nature of the request queue leads to the SCAN algorithm**
- ✓ **The read write head or disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk,**
- ✓ **At the other end, the direction of head movement is reversed and servicing continues.**
- ✓ **Sometimes called the ELEVATOR algorithm.**

SCAN

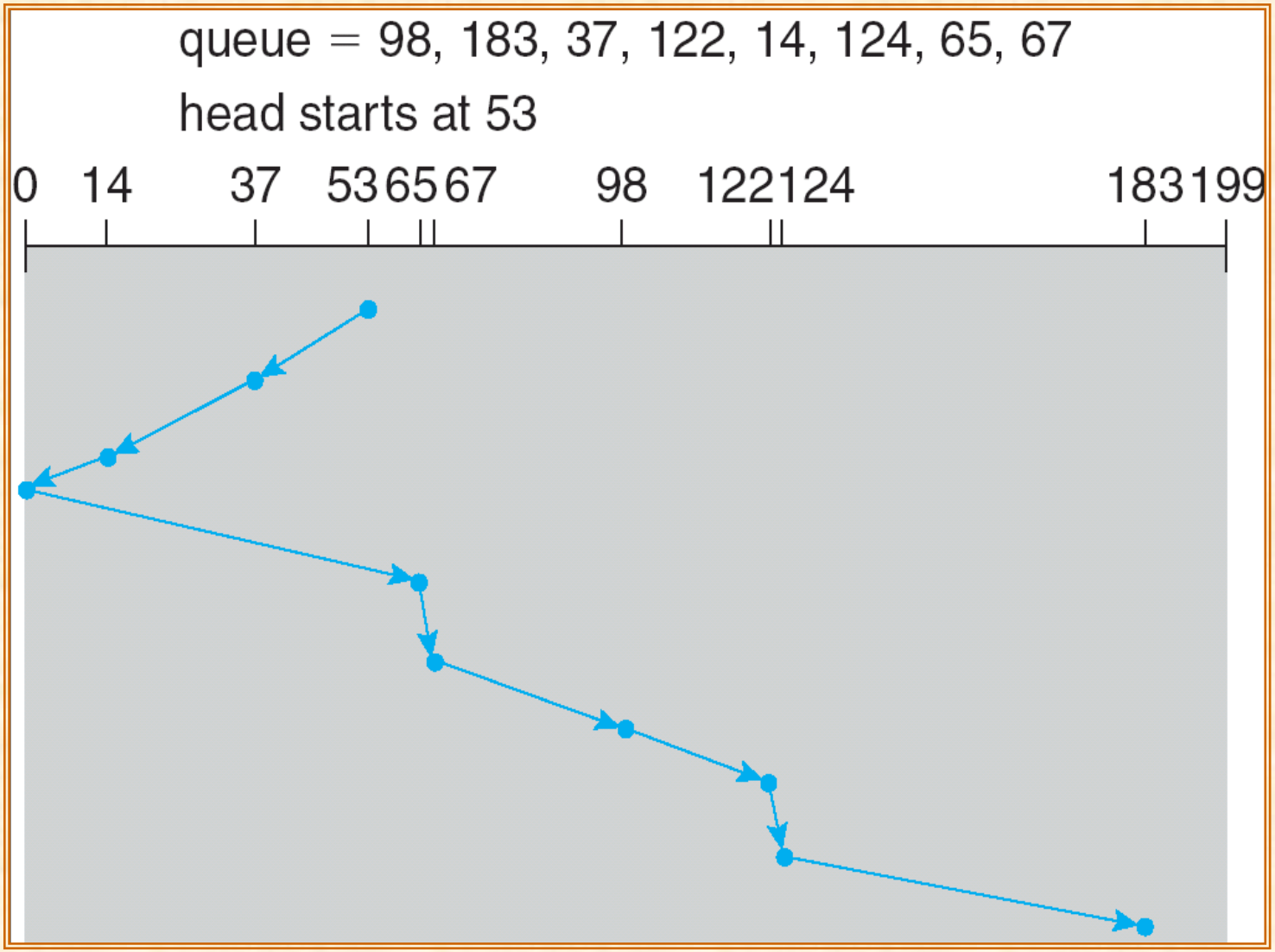


Illustration shows total head movement of 208 cylinders.

C-SCAN – Circular SCAN

- **Provides a more uniform wait time than SCAN.**
- **The head moves from one end of the disk to the other. servicing requests as it goes.**
- **When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.**
- **Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.**

C-SCAN

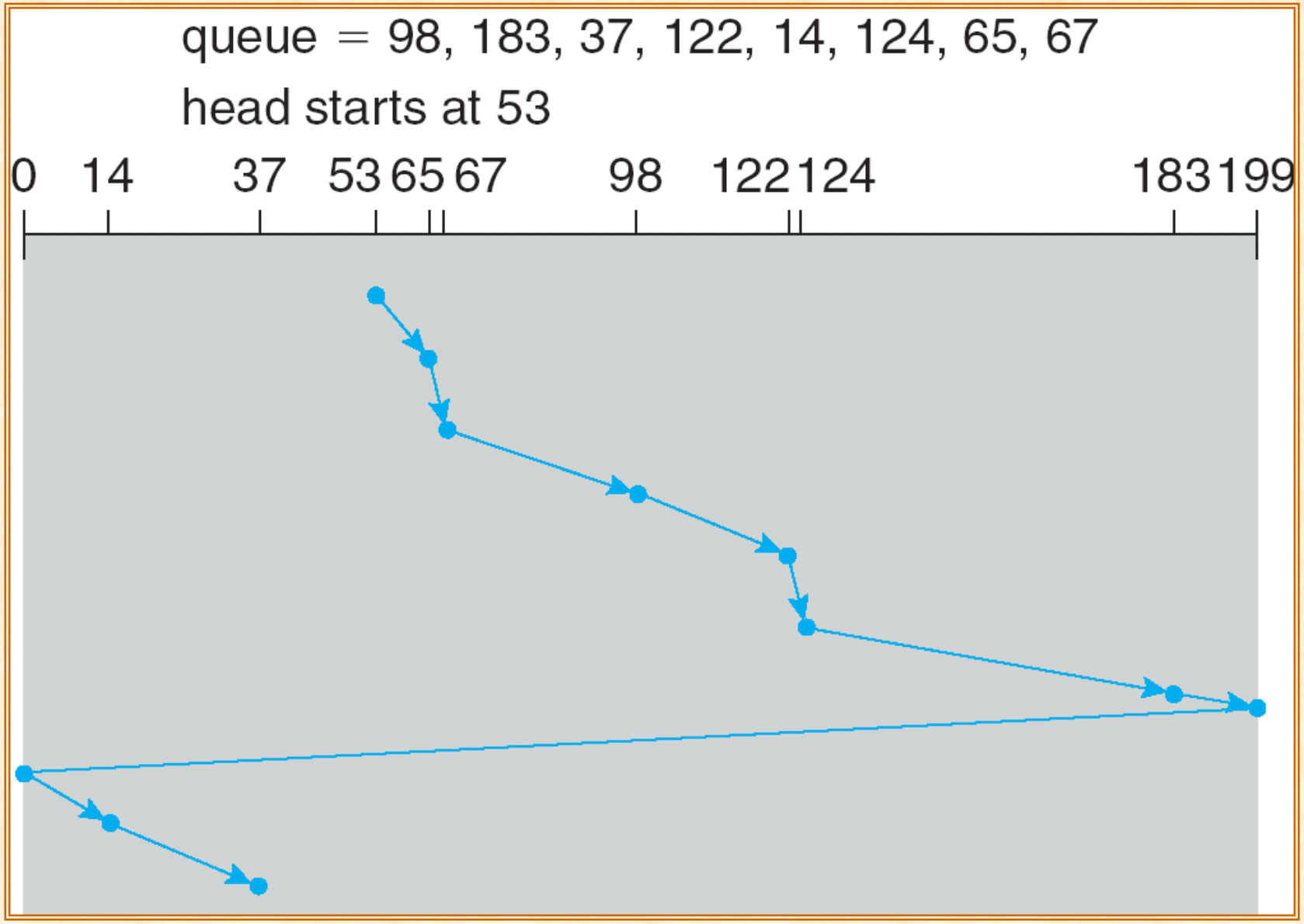


Illustration shows total head movement of 382 cylinders.

C-LOOK

- ✓ **Slightly modified versions of SCAN and C-SCAN scheduling are called LOOK scheduling and C-LOOK scheduling**
- ✓ **Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.**

C-LOOK

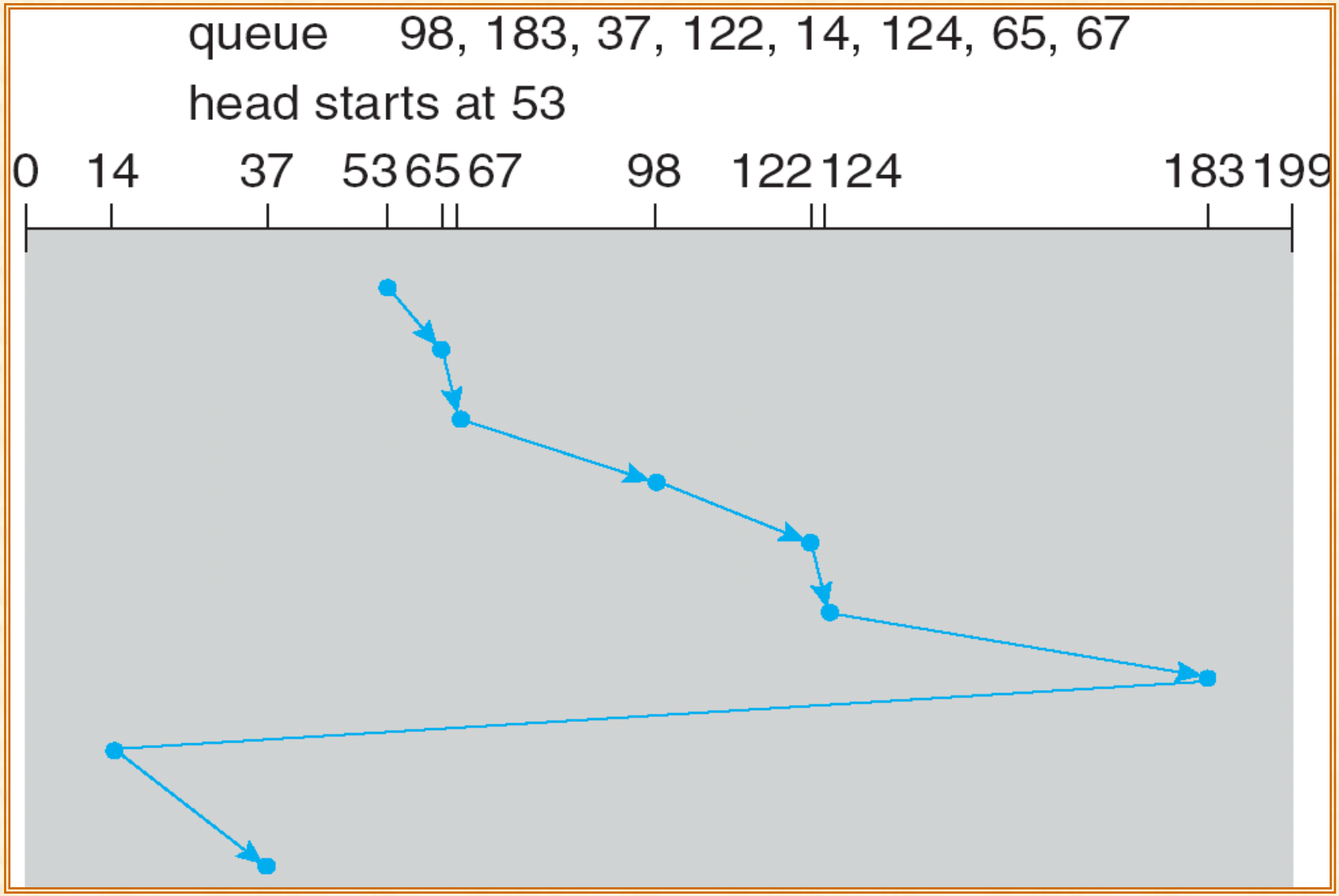


Illustration shows total head movement of 338 cylinders.

Selecting a Disk-Scheduling Algorithm

- Given so many algorithms, how do we choose a particular algorithm?
  - SSTF is common and has a natural appeal
  - SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
  - Performance depends on the number and types of requests.
  - Requests for disk service can be influenced by the file-allocation method.
  - The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary.
  - Either SSTF or LOOK is a reasonable choice for the default algorithm.

Selecting a Disk-Scheduling Algorithm

- **To avoid starvation Linux implements deadline scheduler**
  - **Maintains separate read and write queues, gives read priority**
    - **Because processes more likely to block on read than write**
  - **Implements four queues: 2 x read and 2 x write**
    - **1 read and 1 write queue sorted in Logical Block Addressing (LBA) order, essentially implementing C-SCAN**
    - **1 read and 1 write queue sorted in FCFS order**
    - **All I/O requests sent in batch sorted in that queue's order**
    - **After each batch, checks if any requests in FCFS older than configured age (default 500ms)**
      - **If so, LBA queue containing that request is selected for next batch of I/O**
- **In RHEL 7 also NOOP (** The Noop I/O scheduler implements a simple first-in first-out (FIFO) scheduling algorithm **) and completely fair queueing scheduler (CFQ)** also available, defaults vary by storage device

Storage Device Management

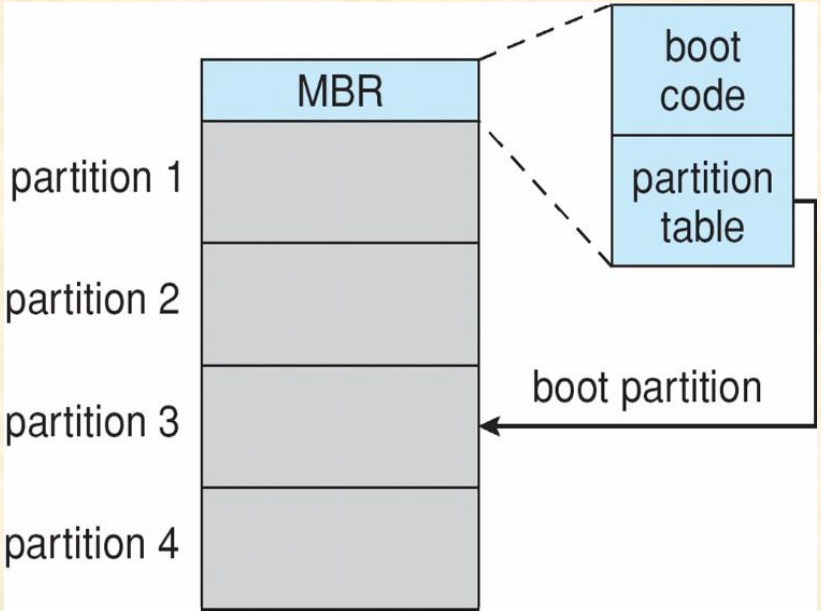
- **Disk formatting:**
  - **Low-level formatting or physical formatting — Dividing a disk into sectors that the disk controller can read and write.**
  - **Each sector has a header containing the sector number and space for an error correction code (ECC)**
- **To use a disk to hold files, the operating system still needs to record its own data structures on the disk.**
  - **Partition the disk into one or more groups of cylinders and format logically a disk before it can be used**
  - **Logical formatting or "making a file system" writes an initial, blank directory onto the disk, and may install a FAT, inodes, free space lists etc.**
  - **To increase efficiency most file systems group blocks into clusters**
    - **Disk I/O done in blocks**
    - **File I/O done in clusters**

Storage Device Management (cont.)

- Root partition contains the OS, other partitions can hold other Oses, other file systems, or be raw
  - Mounted at boot time
  - Other partitions can mount automatically or manually
- At mount time, file system consistency checked
  - Is all metadata correct?
    - If not, fix it, try again
    - If yes, add to mount table, allow access
- Boot block can point to boot volume or boot loader set of blocks that contain enough code to know how to load the kernel from the file system
  - Or a boot management program for multi-OS booting

Storage Device Management (cont.)

- Raw disk access for apps that want to do their own block management, keep OS out of the way (databases for example)
- Boot block initializes system
  - The bootstrap is stored in ROM, firmware
  - Bootstrap loader program stored in boot blocks of boot partition
- Methods such as sector sparing used to handle bad blocks

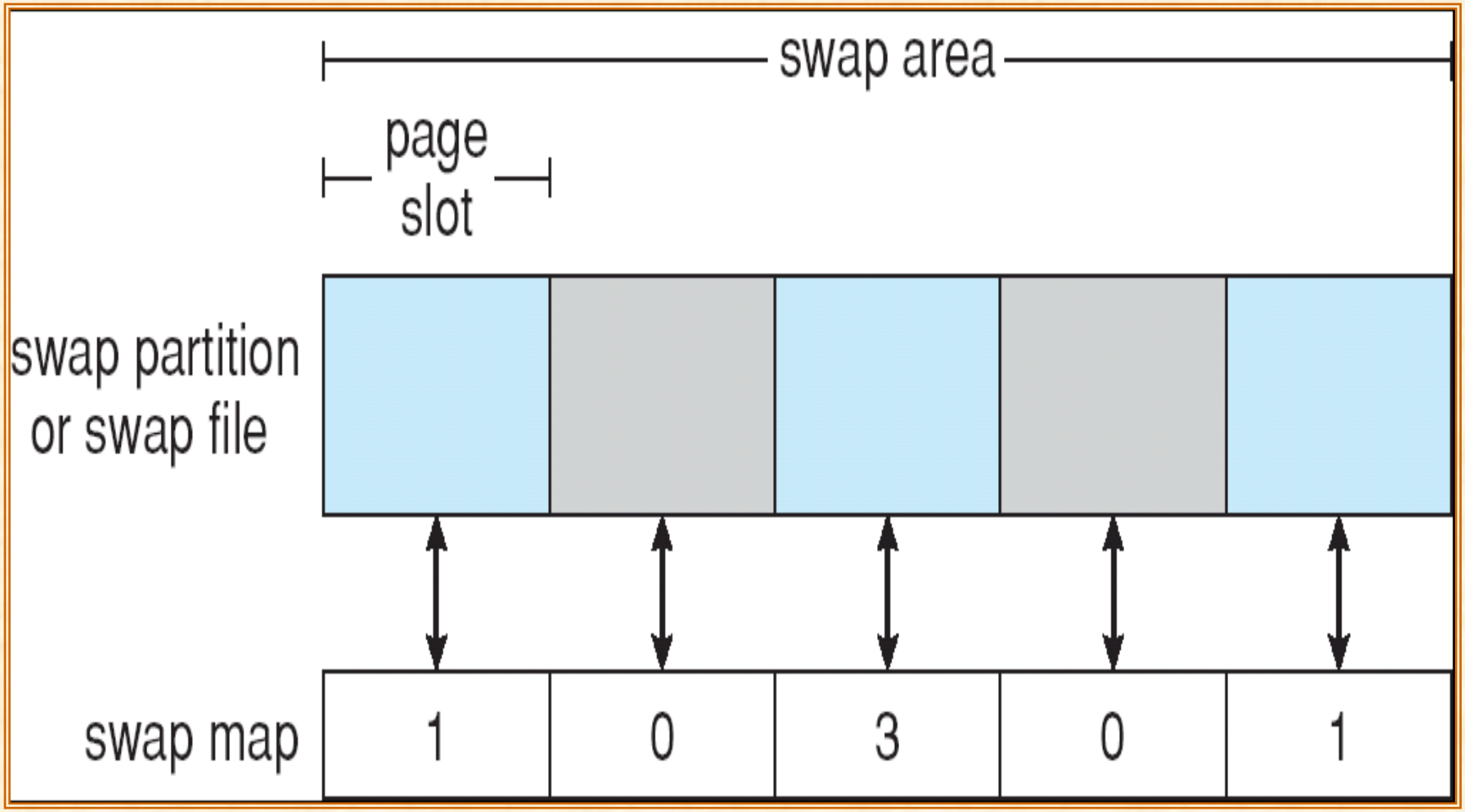


Booting from secondary storage in Windows

Swap-Space Management

- Used for moving entire processes (swapping), or pages (paging), from DRAM to secondary storage when DRAM not large enough for all processes
- Operating system provides swap space management
  - Secondary storage slower than DRAM, so important to optimize performance
  - Usually multiple swap spaces possible – decreasing I/O load on any given device
  - Best to have dedicated devices
  - Can be in raw partition or a file within a file system (for convenience of adding)

Data Structures for Swapping on Linux Systems



Disk Reliability and RAID Structure

- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively.
- Disk striping uses a group of disks as one storage unit.
  - Also called interleaving
  - Improves speed
- RAID (Redundant Arrays of Inexpensive / Independent Disks) schemes
  - improve performance
  - improve the reliability of the storage system by storing redundant data.

RAID Structure

- RAID –multiple disk drives provides reliability via redundancy
- Increases the mean time to failure
- Mean time to repair – exposure time when another failure could cause data loss
- Mean time to data loss based on above factors
- If mirrored disks fail independently, consider disk with 1300,000 mean time to failure and 10 hour mean time to repair
  - Mean time to data loss is $100,000^2 / (2 * 10) = 500 * 10^6$ hours, or 57,000 years!
- Frequently combined with NVRAM to improve write performance

RAID (Cont.)

- Disk striping uses a group of disks as one storage unit
- RAID is arranged into six different levels
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - Mirroring or shadowing (RAID 1) keeps duplicate of each disk
    - This solution is costly, because twice as many disks are used to store the same number of data

RAID (Cont.)

- Striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1) provides high performance and high reliability
- Block interleaved parity (RAID 4, 5, 6) uses much less redundancy

✓ RAID within a storage array can still fail if the array fails, so automatic replication of the data between arrays is common

✓ Frequently, a small number of hot-spare disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them

RAID levels

RAID, a category of disk drives that employ two or more drives in combination for fault tolerance and performance. RAID disk drives are used frequently on servers but aren't generally necessary for personal computers.

There are number of different RAID levels:

Level 0 -- Striped Disk Array without Fault Tolerance: Provides *data striping* (spreading out blocks of each file across multiple disk drives) but no redundancy. This improves performance but does not deliver fault tolerance. If one drive fails then all data in the array is lost.

Level 1 -- Mirroring and Duplexing: Provides disk mirroring. Level 1 provides twice the read transaction rate of single disks and the same write transaction rate as single disks.

Level 2 -- Error-Correcting Coding: Not a typical implementation and rarely used, Level 2 stripes data at the bit level rather than the block level.

RAID levels

Level 3 -- Bit-Interleaved Parity: Provides byte-level striping with a dedicated parity disk. Level 3, which cannot service simultaneous multiple requests, also is rarely used.

Level 4 -- Dedicated Parity Drive: A commonly used implementation of RAID, Level 4 provides block-level striping (like Level 0) with a parity disk. If a data disk fails, the parity data is used to create a replacement disk. A disadvantage to Level 4 is that the parity disk can create write bottlenecks.

Level 5 -- Block Interleaved Distributed Parity: Provides data striping at the byte level and also stripe error correction information. This results in excellent performance and good fault tolerance. Level 5 is one of the most popular implementations of RAID.

Level 6 -- Independent Data Disks with Double Parity: Provides block-level striping with parity data distributed across all disks.

RAID levels

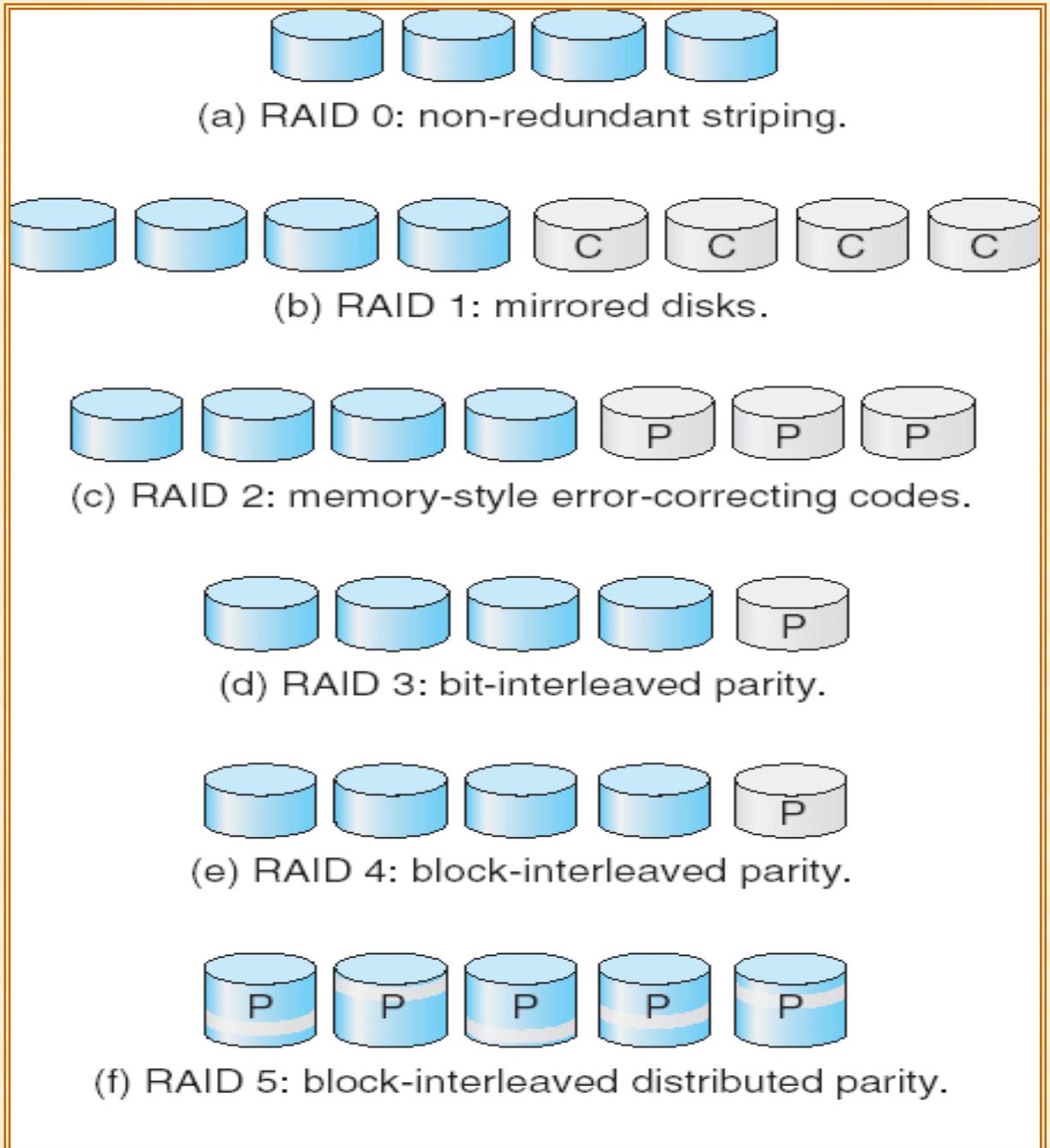
Level 0+1 – A Mirror of Stripes: Not one of the original RAID levels, two RAID 0 stripes are created, and a RAID 1 mirror is created over them. Used for both replicating and sharing data among disks.

Level 10 – A Stripe of Mirrors: Not one of the original RAID levels, multiple RAID 1 mirrors are created, and a RAID 0 stripe is created over these.

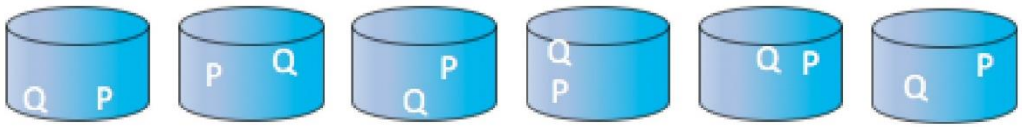
Level 7: A trademark of Storage Computer Corporation that adds caching to Levels 3 or 4.

RAID S: EMC Corporation's proprietary striped parity RAID system used in its Symmetrix storage system

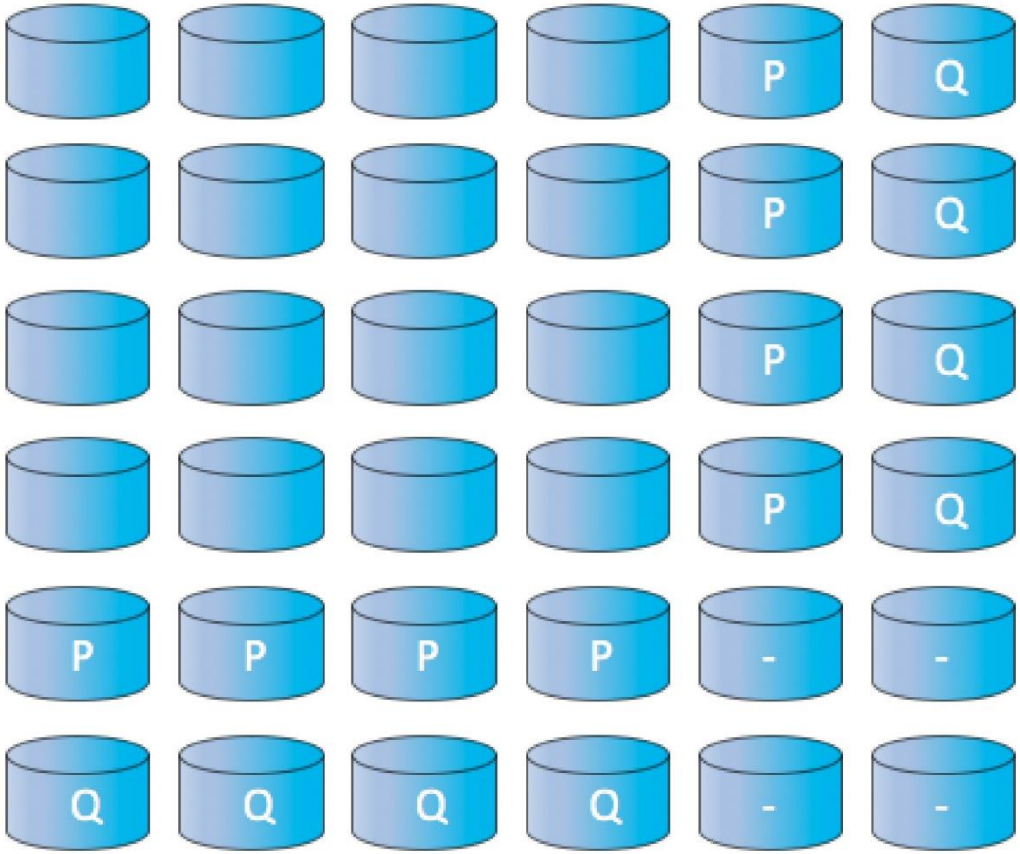
RAID Levels


C
C
C
C
P
P
P
P
P
P
P
P
P
P
(a) RAID 0: non-redundant striping.
(b) RAID 1: mirrored disks.
(c) RAID 2: memory-style error-correcting codes.
(d) RAID 3: bit-interleaved parity.
(e) RAID 4: block-interleaved parity.
(f) RAID 5: block-interleaved distributed parity.

RAID Levels

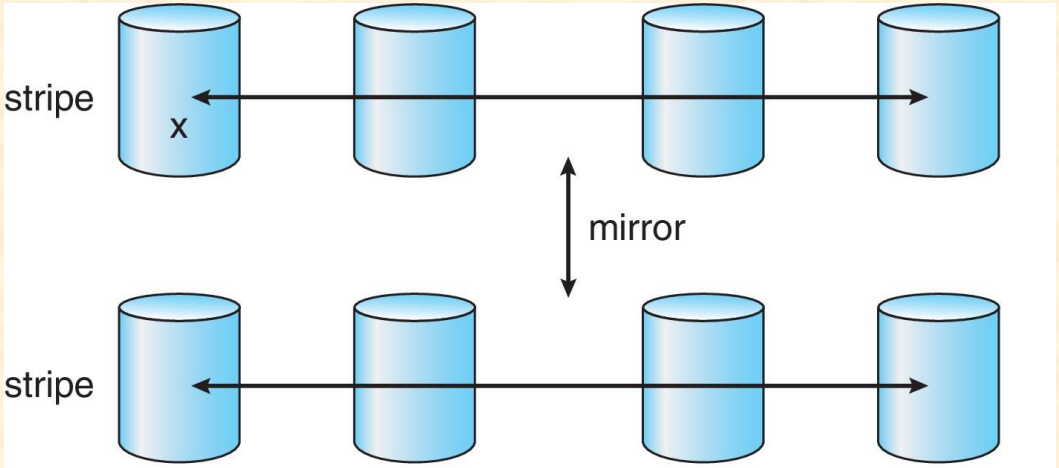


(e) RAID 6: P + Q redundancy.

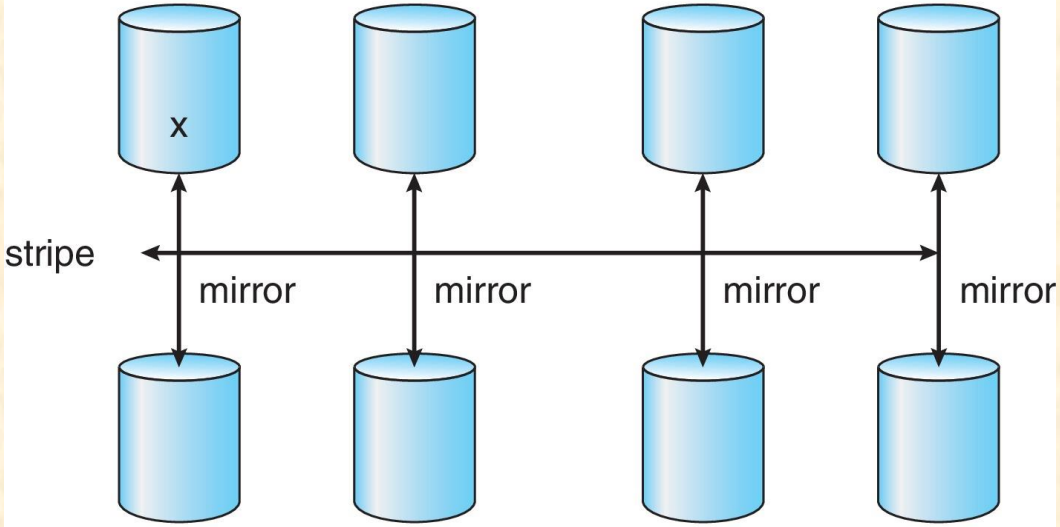


(f) Multidimensional RAID 6.

RAID (0 + 1) and (1 + 0)



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

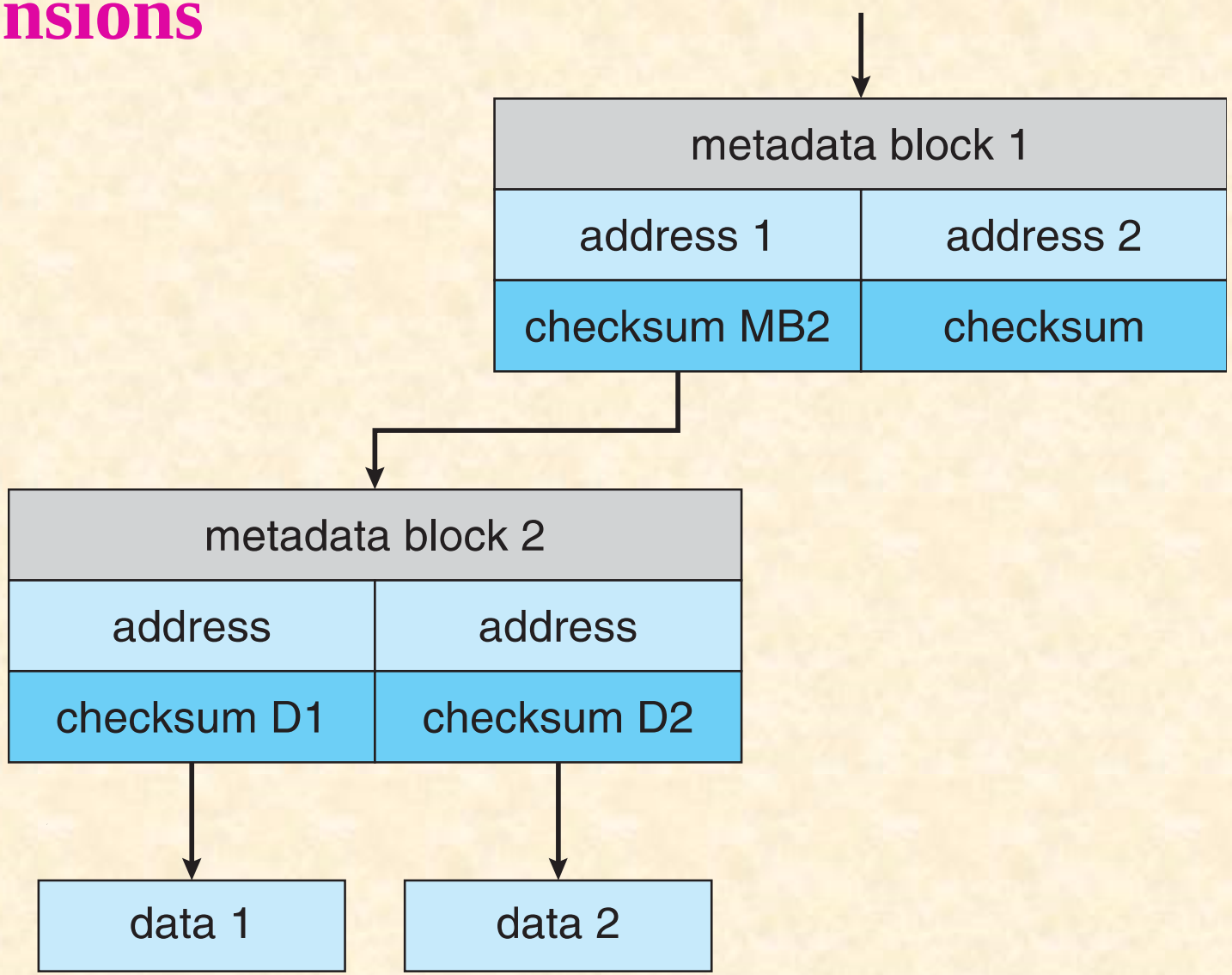
Other Features

- Regardless of where RAID implemented, other useful features can be added
- Snapshot is a view of file system before a set of changes take place (i.e. at a point in time)
- Replication is automatic duplication of writes between separate sites
  - For redundancy and disaster recovery
  - Can be synchronous or asynchronous
- Hot spare disk is unused, automatically used by RAID production if a disk fails to replace the failed disk and rebuild the RAID set if possible
  - Decreases mean time to repair

Extensions

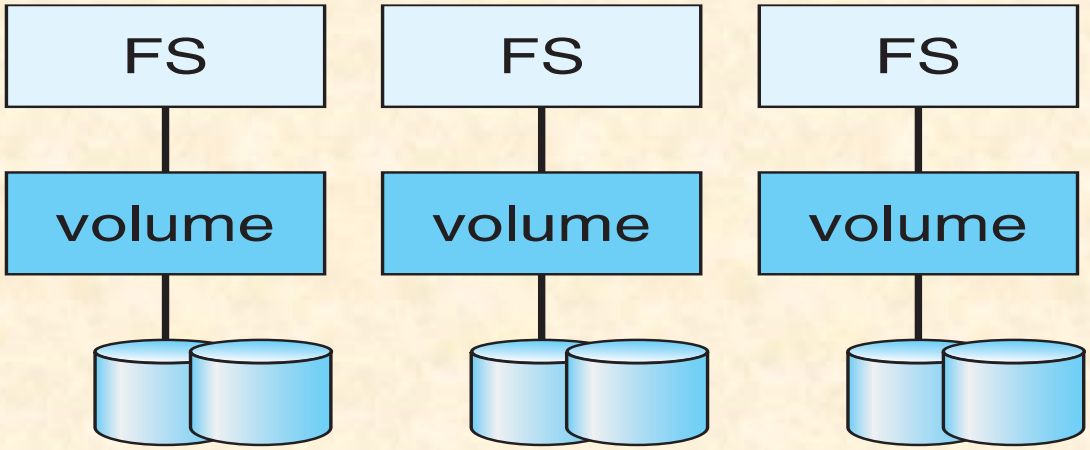
- RAID alone does not prevent or detect data corruption or other errors, just disk failures
- Solaris ZFS(Zettabyte File System) adds checksums of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partitions
  - Disks allocated in pools
  - Filesystems with a pool share that pool, use and release space like `malloc()` and `free()` memory allocate / release calls

Extensions

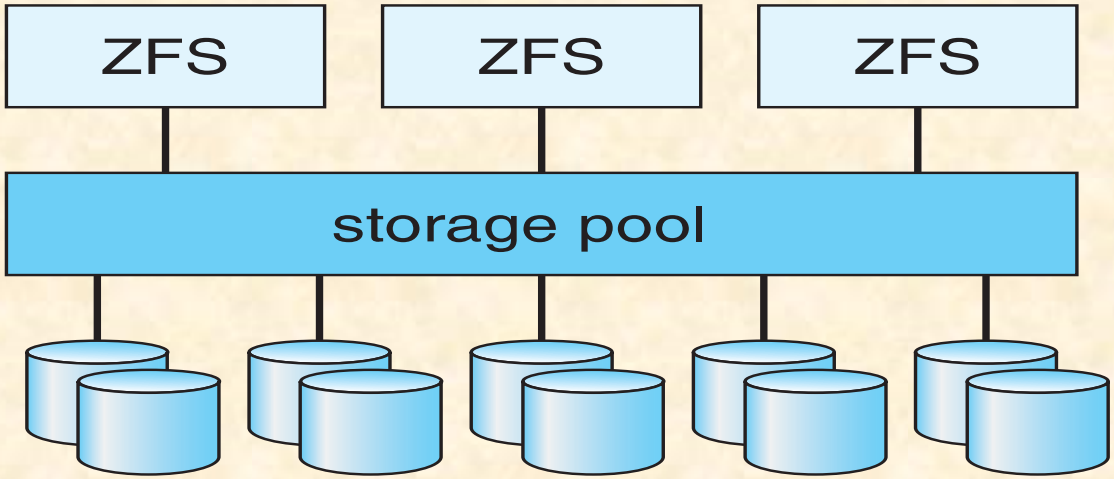


ZFS checksums all metadata and data

Traditional and Pooled Storage



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

Object Storage

- General-purpose computing, file systems not sufficient for very large scale
- Another approach – start with a storage pool and place objects in it
  - Object just a container of data
  - No way to navigate the pool to find objects (no directory structures, few services
  - Computer-oriented, not user-oriented
- Typical sequence
  - Create an object within the pool, receive an object ID
  - Access object via that ID
  - Delete object via that ID

Object Storage

- **Object storage management software like Hadoop file system (HDFS) and Ceph determine where to store objects, manages protection**
  - **Typically by storing N copies, across N systems, in the object storage cluster**
  - **Horizontally scalable**
  - **Content addressable, unstructured**

Ceph is an open source software put together to facilitate highly scalable object, block and file-based storage under one whole system. The clusters of Ceph are designed in order to run commodity hardware with the help of an algorithm called CRUSH (Controlled Replication Under Scalable Hashing).

Stable-Storage Implementation

- **Write-ahead log scheme requires stable storage**
- **Information residing in stable storage is never lost**
- **To implement stable storage:**
  - **Replicate information on more than one nonvolatile storage media with independent failure modes.**
  - **Update information in a controlled manner to ensure that we can recover the stable data after any failure during data transfer or recovery.**

End