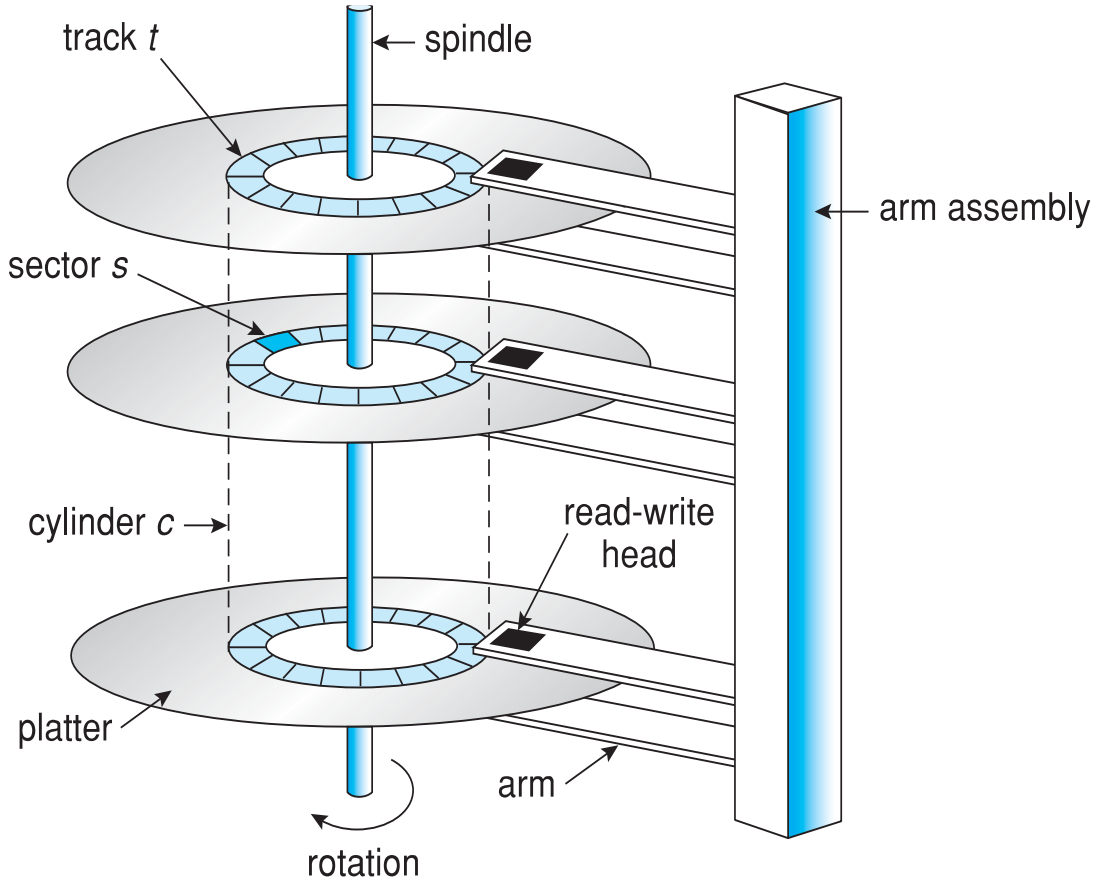
Mass Storage Management

Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 250 times per second
- Disks can be removable
- Drive attached to computer via I/O bus
  - Busses vary, including EIDE, ATA, SATA, USB, Fibre Channel, SCSI, SAS, Firewire
  - Host controller in computer uses bus to talk to disk controller built into drive or storage array

Moving-head Disk Mechanism



Disk Structure

- 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.

The read/write data is a three-stage process:

- Position the head/arm over the proper track (into proper cylinder). The time required to move the heads from one cylinder to another, and for the heads to settle down after the move is called as Seek time/Positioning time (random-access time).
- Wait for the desired sector to rotate under the read/write head. The amount of time required for the desired sector to rotate around and come under the read-write head is referred as Rotational latency.
- Transfer a block of bits (sector) under the read-write head. The time required to move the data electronically from the disk to the computer is called Transfer time.

The disk access time is computed as follows:

Seek Time: **$T_S = m^* n + s$**

where *n* is the number of tracks traversed, *m* is the track traversal time and *s* is the startup time

Rotational Latency: **$T_R = 1/ (2 * r)$**

where *r* is the number of revolutions per time unit

Transfer Time: **$T_T = b/ (r * N)$**

where *b* is the number of bytes to be transferred and N is the number of bytes on track

Disk Access Time: **$T = T_S + T_R + T_T$**

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.

- **Magnetic Tape Storage:**

It is sequential access storage and it is both persistent and rewritable. This device is not suitable for transaction processing where random (direct) access is used for accessing data. Also the access time is very slow. These are used mostly for back up purposes.

- **Disk Storage:**

These are random access devices and the storage capacity varies from megabytes to terabytes. These devices exhibit variable access speed that depends on the relative positions of the read-write head and the requested data. Some examples of disks are magnetic disks, magneto-optical disks, floppies and CDs.

Disk Attachments

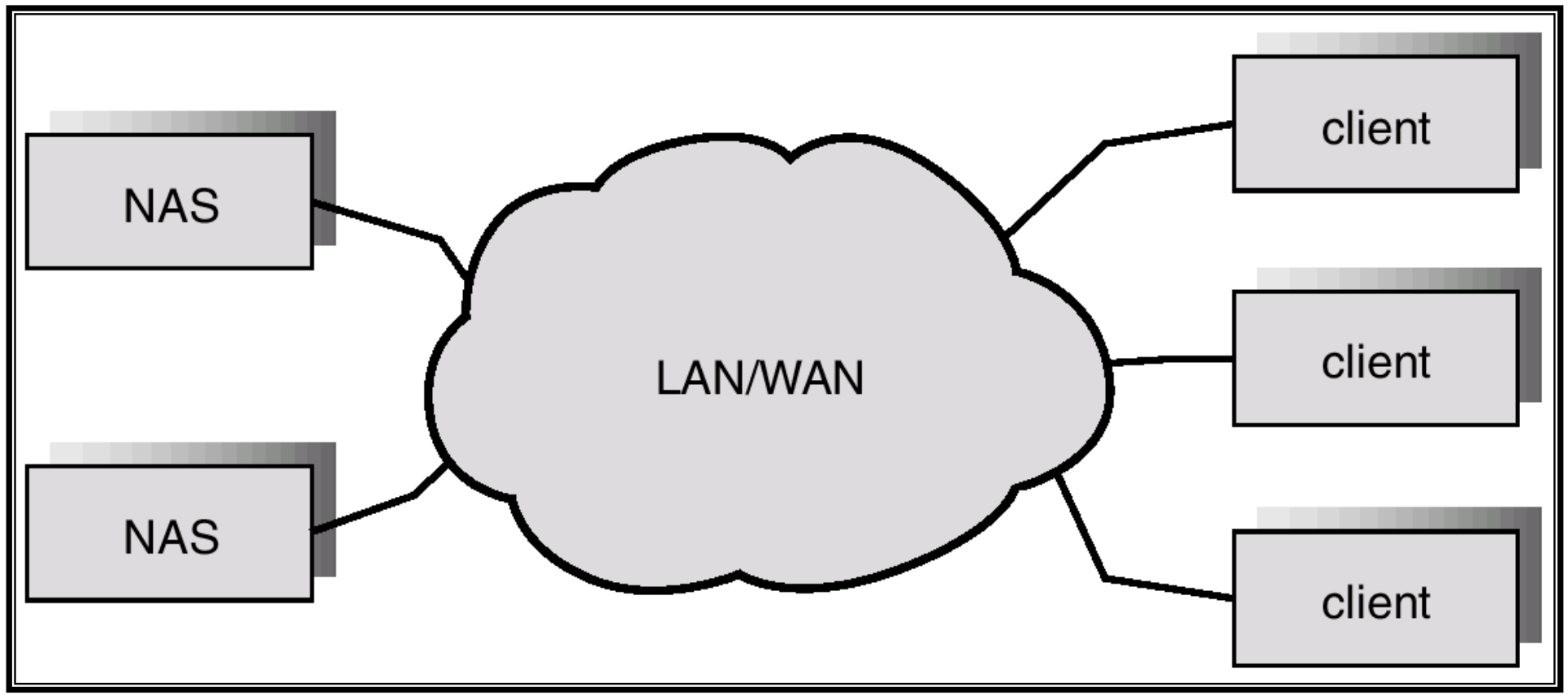
Disk drives can be attached either directly to a particular host machine or to a network.

Host-Attached Storage

The storage is referred as Local disk and accessed through I/O Ports. The most common interfaces are IDE or ATA, each of which allow up to two drives per host controller.

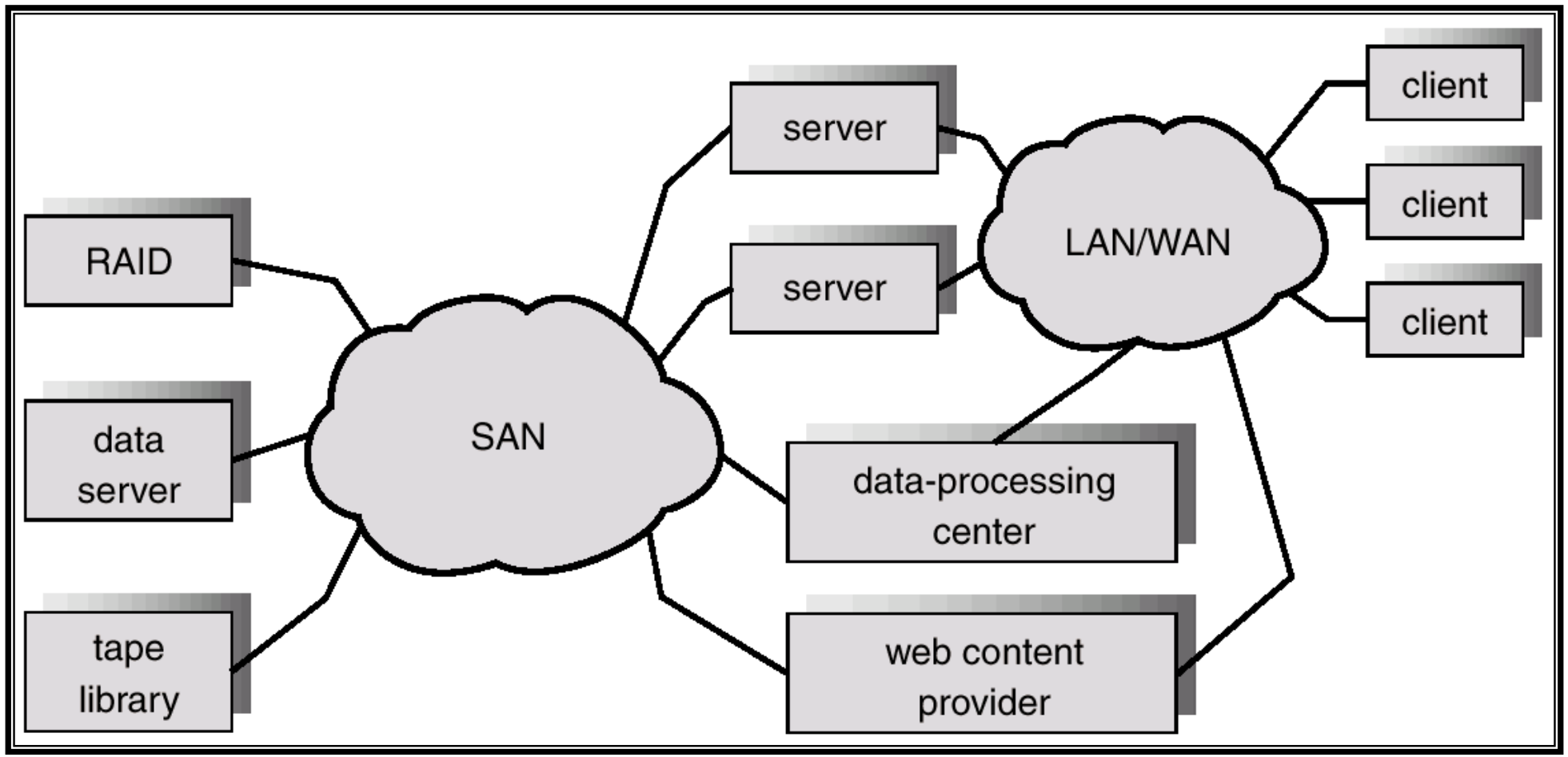
Network-Attached Storage

The storage devices are part of a LAN/WAN and are accesses by the clients using Remote Procedure Call (RPC) typically with NFS file system mounting. These devices form a shared storage by using naming conventions and allow group access by the clients.


NAS
NAS
LAN/WAN
client
client
client

Storage-Area Network

- A **Storage-Area Network (SAN)** connects computers and storage devices in a network, using storage protocols instead of network protocols.
- It is very flexible and dynamic, allowing hosts and devices to attach and detach on the fly and is also controllable allowing restricted access to certain hosts and devices.

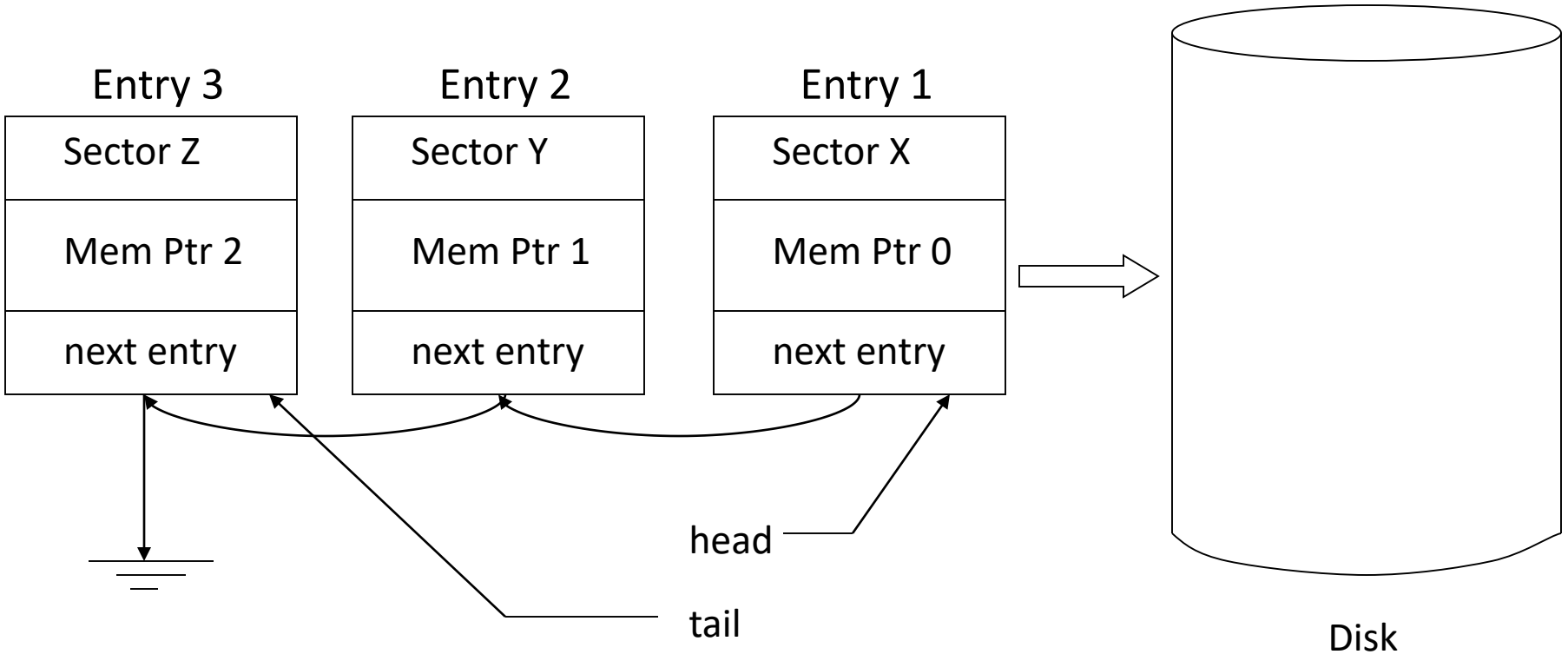

client
server
client
LAN/WAN
RAID
server
client
data
server
SAN
data-processing
center
tape
library
web content
provider

Disk Scheduling

Disk Queues

- Each disk has a queue of jobs waiting to access disk
  - read jobs
  - write jobs
- Each entry in queue contains the following
  - pointer to memory location to read/write from/to
  - sector number to access
  - pointer to next job in the queue
- OS usually maintains this queue

Disk Queues


Entry 3
Sector Z
Mem Ptr 2
next entry
Entry 2
Sector Y
Mem Ptr 1
next entry
Entry 1
Sector X
Mem Ptr 0
next entry
head
tail
Disk

Disk Scheduling

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

Example

- A request queue is (0-199).

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53

FCFS

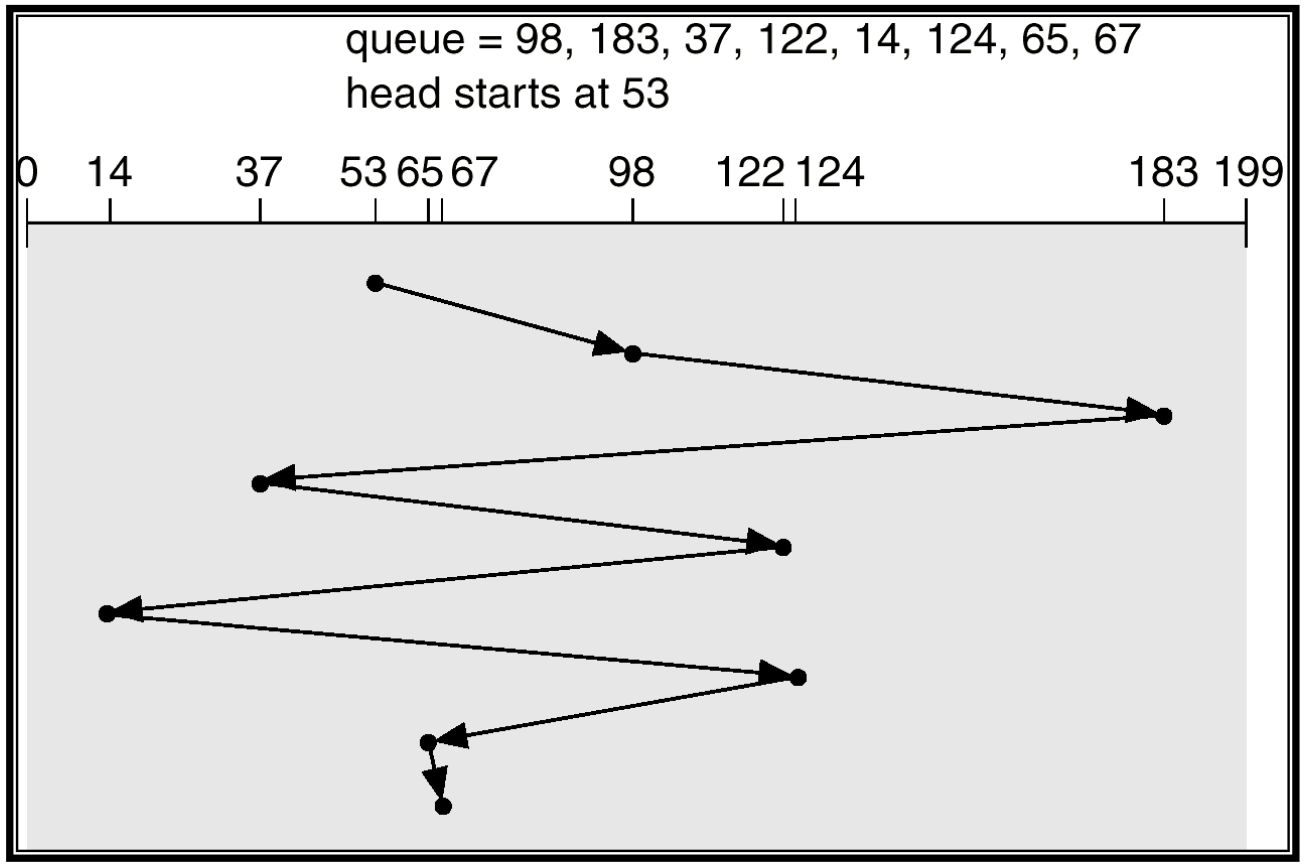
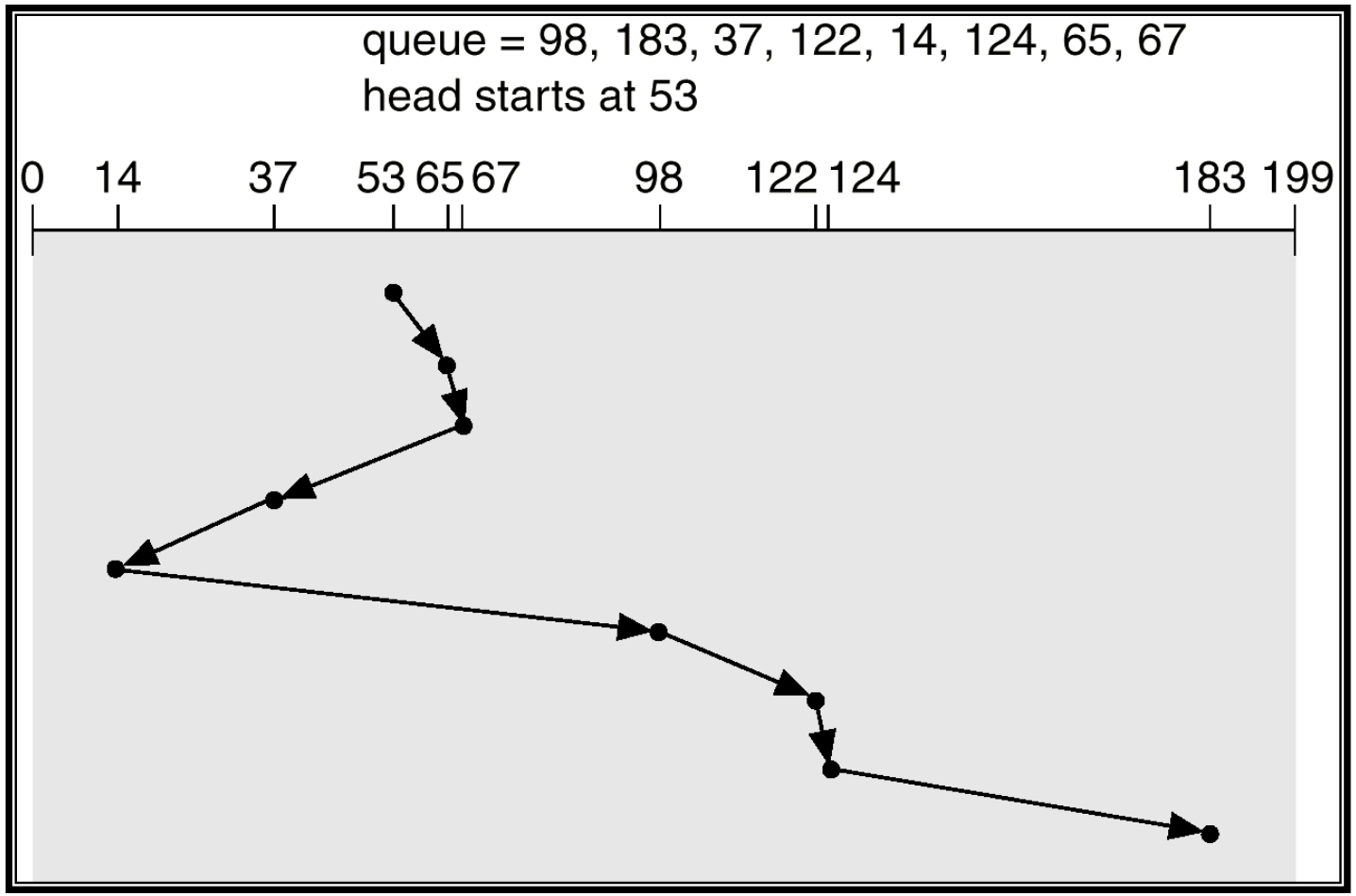


Illustration shows total head movement of 640 cylinders.

SSTF

- 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.
- total head movement of 236 cylinders.

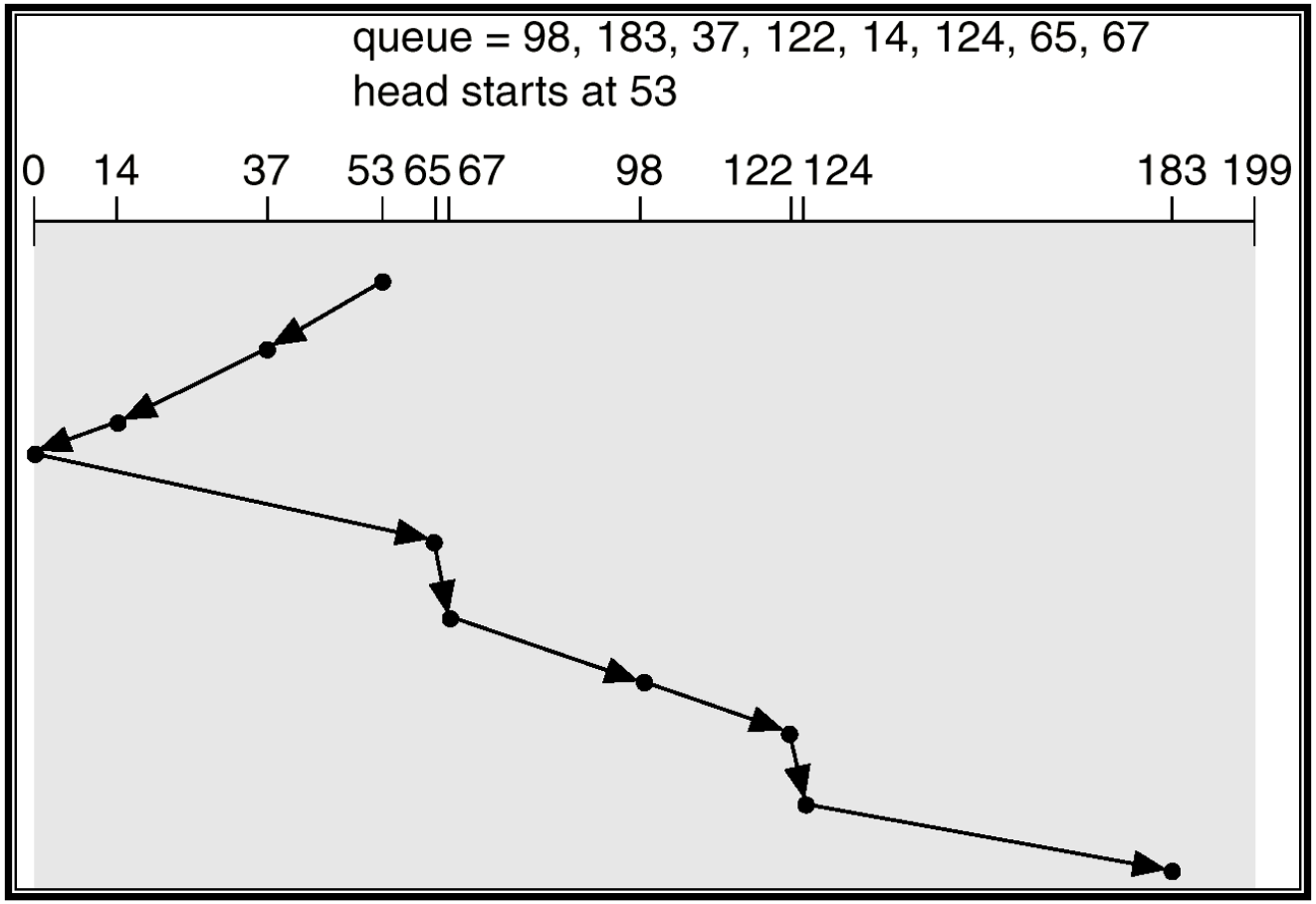
SSTF


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

SCAN

- The 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, where the head movement is reversed and servicing continues.
- Sometimes called the *elevator algorithm*.
- total head movement of 208 cylinders.

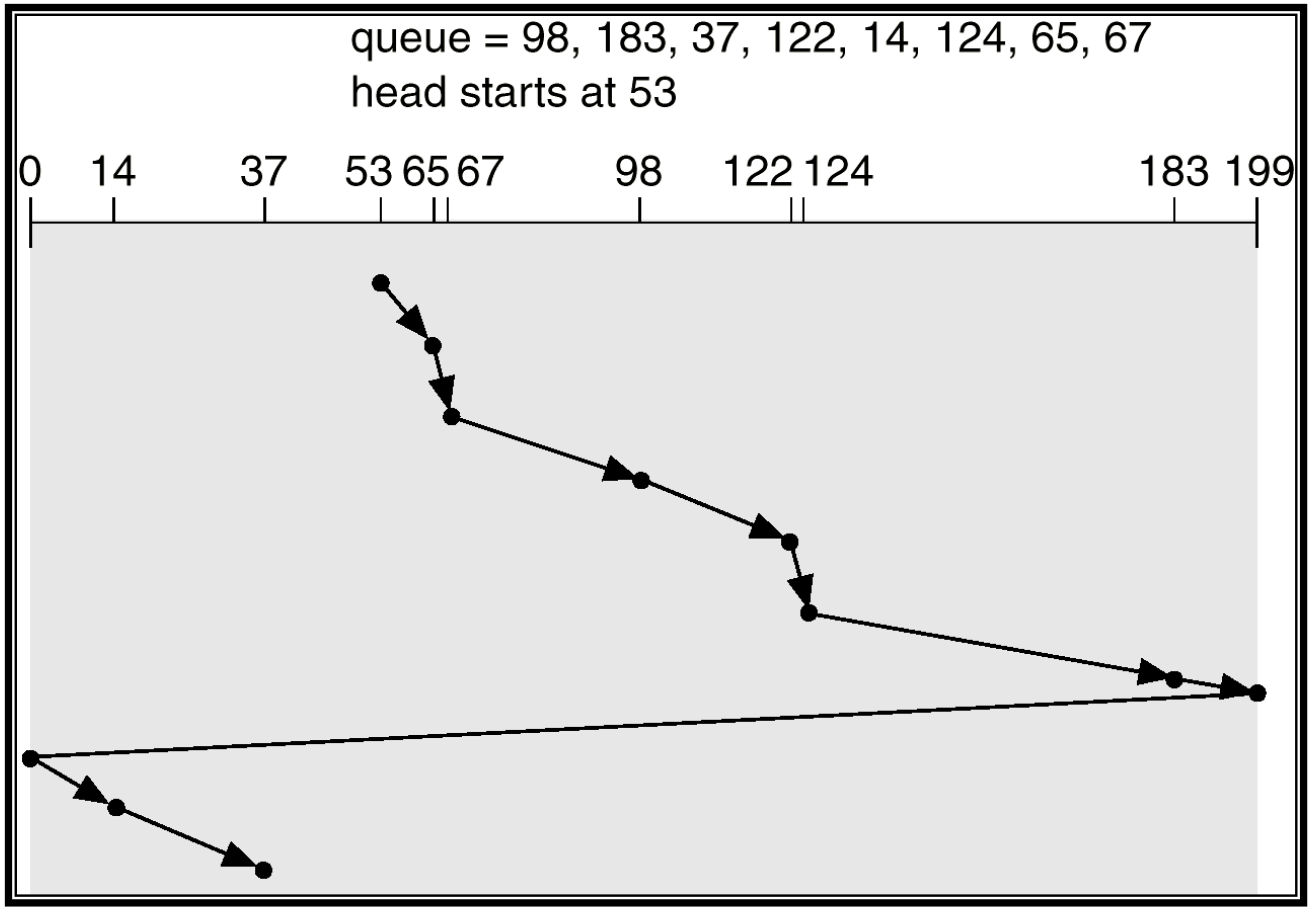
SCAN


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Circular SCAN (C-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


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

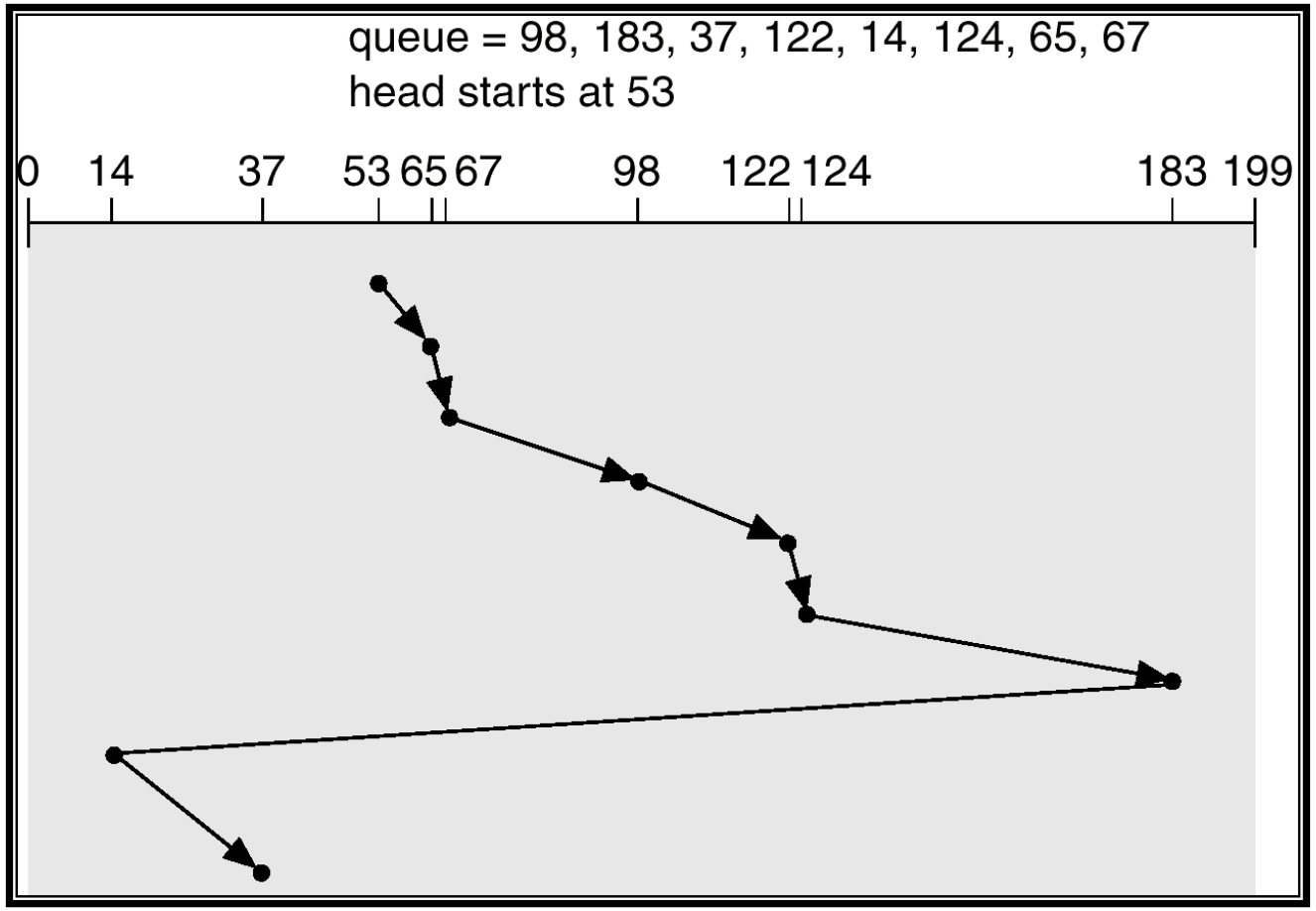
LOOK

- The head moves in one direction and satisfies the request in that direction, if there is no request in that direction, it reverses its direction and serves request

C-LOOK

- Version of C-SCAN
- 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


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Selecting a Disk-Scheduling 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.

Redundant Array of Inexpensive/Independent Disks (RAID)

- Secondary storage devices are slow and to improve their performance multiple devices in parallel such as arrays of disks are used.
- RAID is a group of hard drives together with some form of redundancy is employed to improve the performance and to provide reliability.
- Single large capacity disk is replaced with array of smaller capacity disks.
- Earlier small cheaper disks were used so it was inexpensive but now large expensive disks are used so it is independent.

Common characteristics:

- Array of physical disks are visible as single device to OS.
- Data is distributed across physical drives of array.
- Redundant disk capacity is used for error detection/correction.

Benefits:

- Improved I/O performance; Enables incremental upgrade

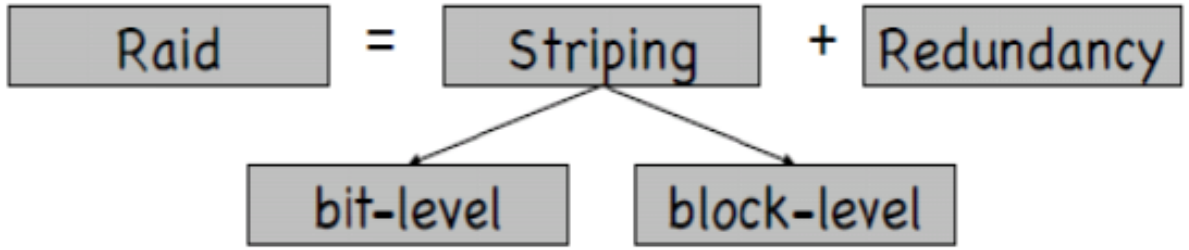
Problem:

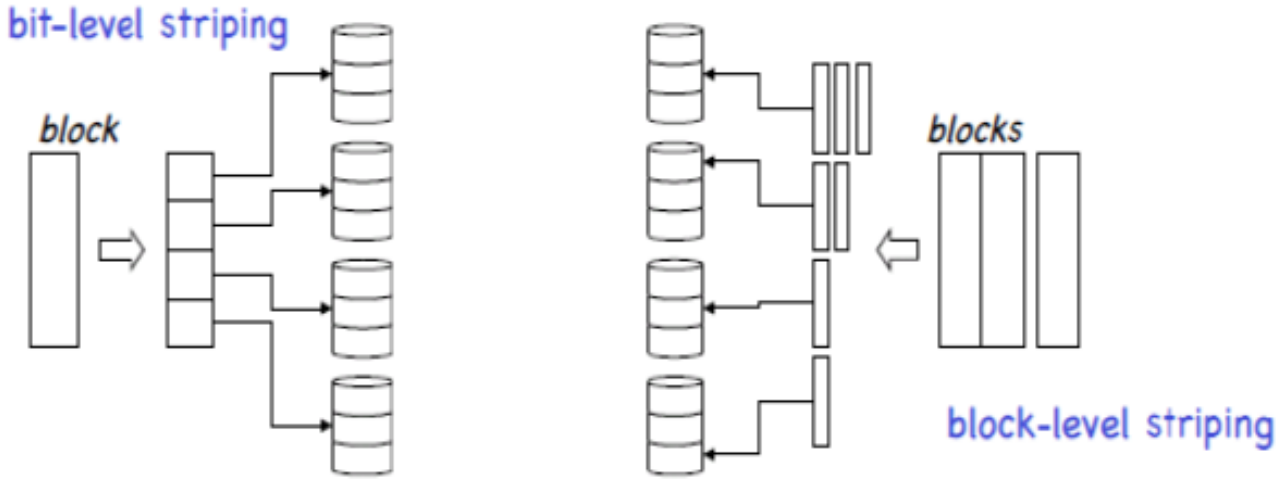
- Reliability is achieved through more devices that increase the probability of failure and the solution is redundancy.

- RAID is arranged into six different levels.
- 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.

- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.
  - *Mirroring* or *shadowing* keeps duplicate of each disk.
  - *Block interleaved parity* uses much less redundancy.

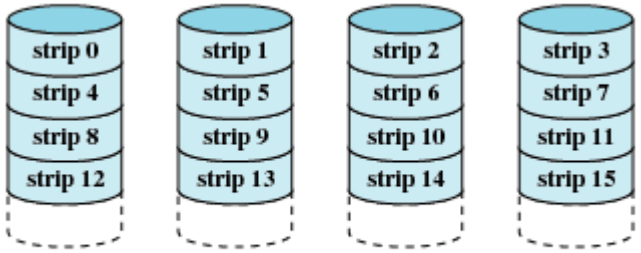
Bit level and block level striping





RAID 0 (non-redundant)

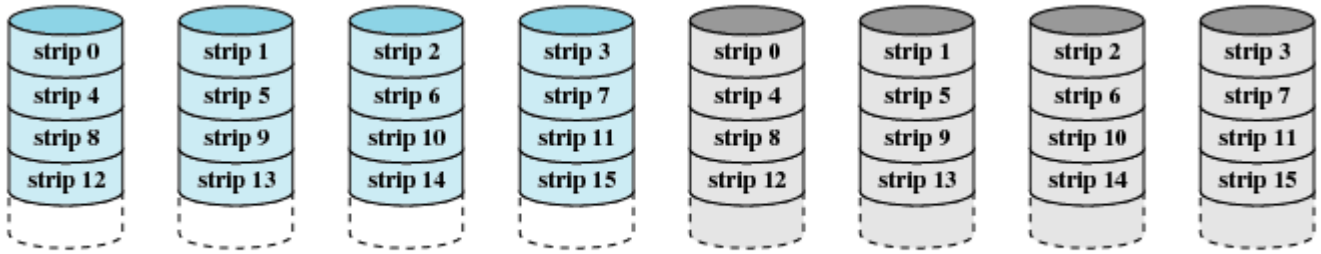
- Block level striped set without Parity
- RAID 0 offers no redundancy, but improves disk access
- Here, files are broken into strips and distributed across disk surfaces (known as disk spanning) so that access to a single file can be done in parallel disk accesses



(a) RAID 0 (non-redundant)

RAID 1 (mirrored)

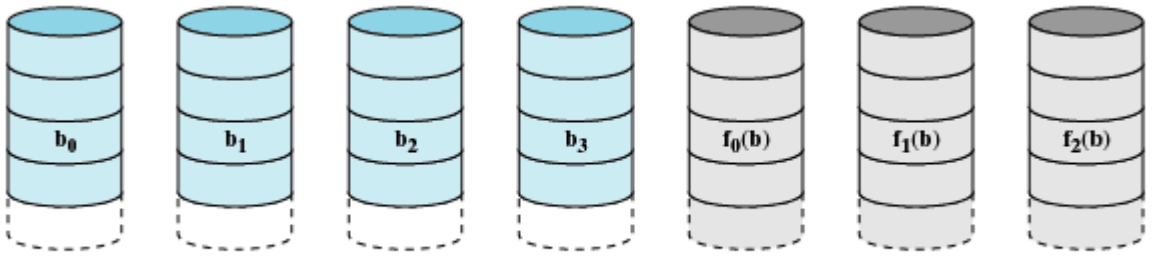
- Mirrored set without parity
- 100 percent redundancy leads to increase in cost.
- Read operation is done with multithreading and split reads. There is small penalty in write operation because of redundancy (writes require saving to both sets of disk).



(b) RAID 1 (mirrored)

RAID 2 (redundancy through Hamming code)

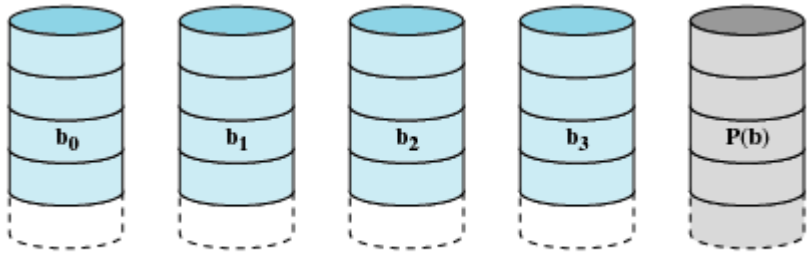
- Memory style error correcting parity
- Head and spindles are synchronized.
- Strips are small.
- It strips each byte into 1 bit per disk and uses additional disks to store Hamming codes for redundancy
- Error correction codes are corrected over bits of data disks.
- It is suitable for system with many failures.



(c) RAID 2 (redundancy through Hamming code)

RAID 3 (bit-interleaved parity)

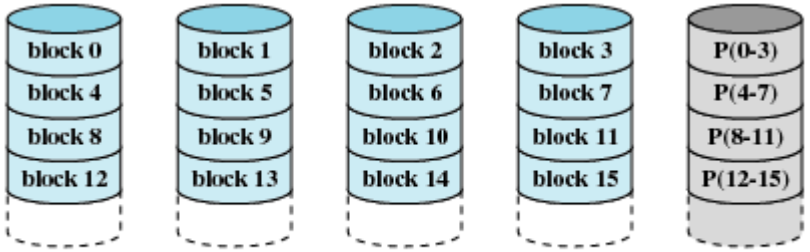
- Bit interleaved Parity Head and spindles are synchronized.
- Strips are small.
- Simple parity bits are used instead of Error correcting codes.
- most suitable for small computer systems that require some but not total redundancy



(d) RAID 3 (bit-interleaved parity)

RAID 4 (block-level parity)

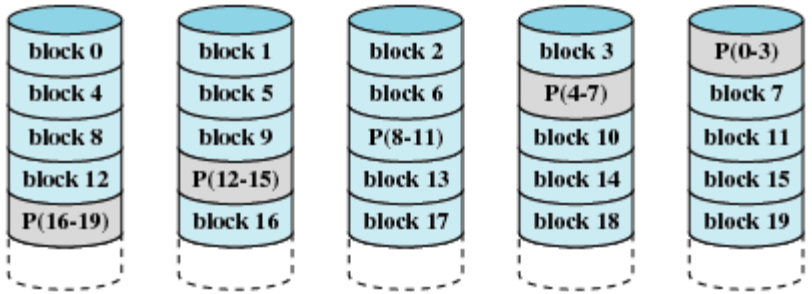
- Block Level Parity ; Same as RAID 3 but with block level striping
- Disks are not synchronized
- Strips are large and each strip contains parity information of all corresponding strips
- All parity information is placed on a single disk which creates a bottleneck and so defeats the advantage of parallel accesses



(e) RAID 4 (block-level parity)

RAID 5 (block-level distributed parity)

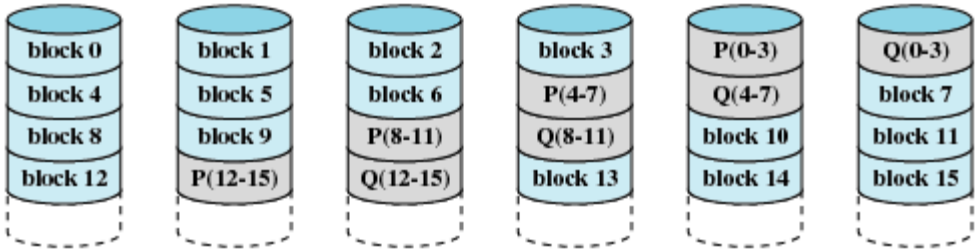
- Striped Set with Interleaved parity
- Same as RAID 4 but parity spread across all disks
- Disks are not synchronized
- Strips are large strips



(f) RAID 5 (block-level distributed parity)

RAID 6 (dual redundancy)

- Striped Set with Dual Interleaved Parity
- Same as RAID 5 but uses 2 bits for storing parity
- Disks are not synchronized
- Strips are large
- ECC is used instead of parity
- Tolerates two failures



(g) RAID 6 (dual redundancy)