BITS Pilani
Hyderabad Campus

BITS Pilani presentation

D. Powar
Lecturer,
BITS-Pilani, Hyderabad Campus

innovate achieve lead

BITS Pilani
Hyderabad Campus

SSZG527

Cloud Computing

Storage as Service

(RAID)

RAID

Data center

RAID (Redundant Array of Independent Disks)

- RAID, originally Redundant Array of Inexpensive Disks, was proposed by Patterson, Gibson, and Katz at UC Berkeley in 1987
- Today, RAID - Redundant Array of Disks that operate independently and in parallel.
- Is a storage technology that provides increased reliability and functions through redundancy.
- The technique combines multiple disk drive components into a logical unit, where data is distributed across the drives in one of several ways called "RAID levels"
- Separate I/O request can be handled in parallel.
- 7 levels (0-6).

Characteristics:

1. Set of physical disks viewed as single logical drive by O/S.
2. Data distributed across physical drives.
3. Can use redundant capacity to store parity information. So that data recovery can be easy in case of disk failure.

RAID provides:

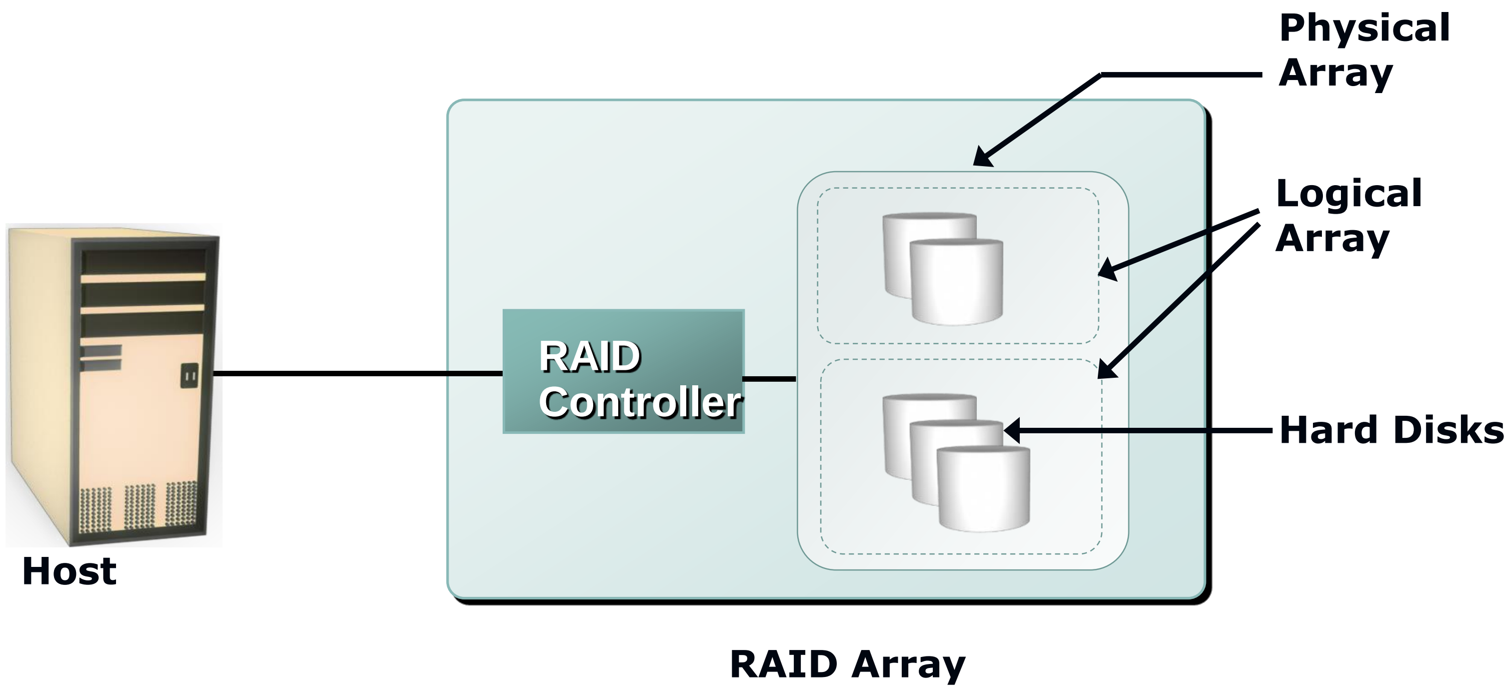
– Increased capacity
– Increased performance
– High availability

RAID Implementations

- Hardware (usually a specialized disk controller card)
  - Controls all drives attached to it
  - Array(s) appear to host operating system as a regular disk drive
  - Provided with administrative software

- Software
  - Runs as part of the operating system
  - Performance is dependent on CPU workload
  - Does not support all RAID levels

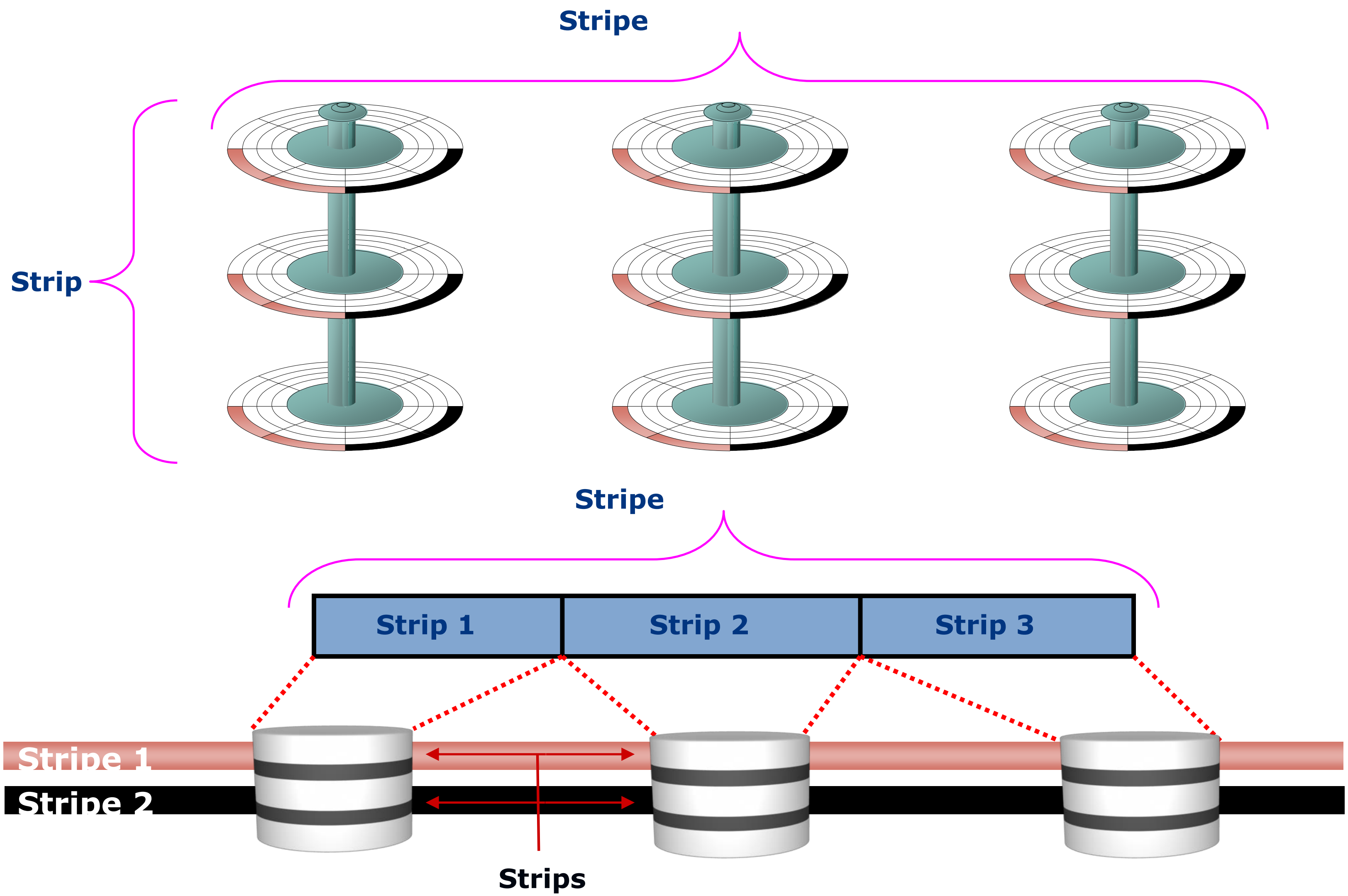
RAID Array Components



RAID terminology

- Mirroring
- Striping
- Fault tolerance

Data Organization: Strip Vs Stripe



Strip size Vs Stripe size

- Stripe size is specified at host level for software RAID and it is vendor specific for hardware RAID.
- When no:of drives in array increases, performance improves.

RAID 0

- No redundancy.
- Striped array with no fault tolerance
- Data striped across all disks.
- The data is broken into fragments called blocks. The number of blocks is dictated by the **stripe size**
- Round Robin striping.

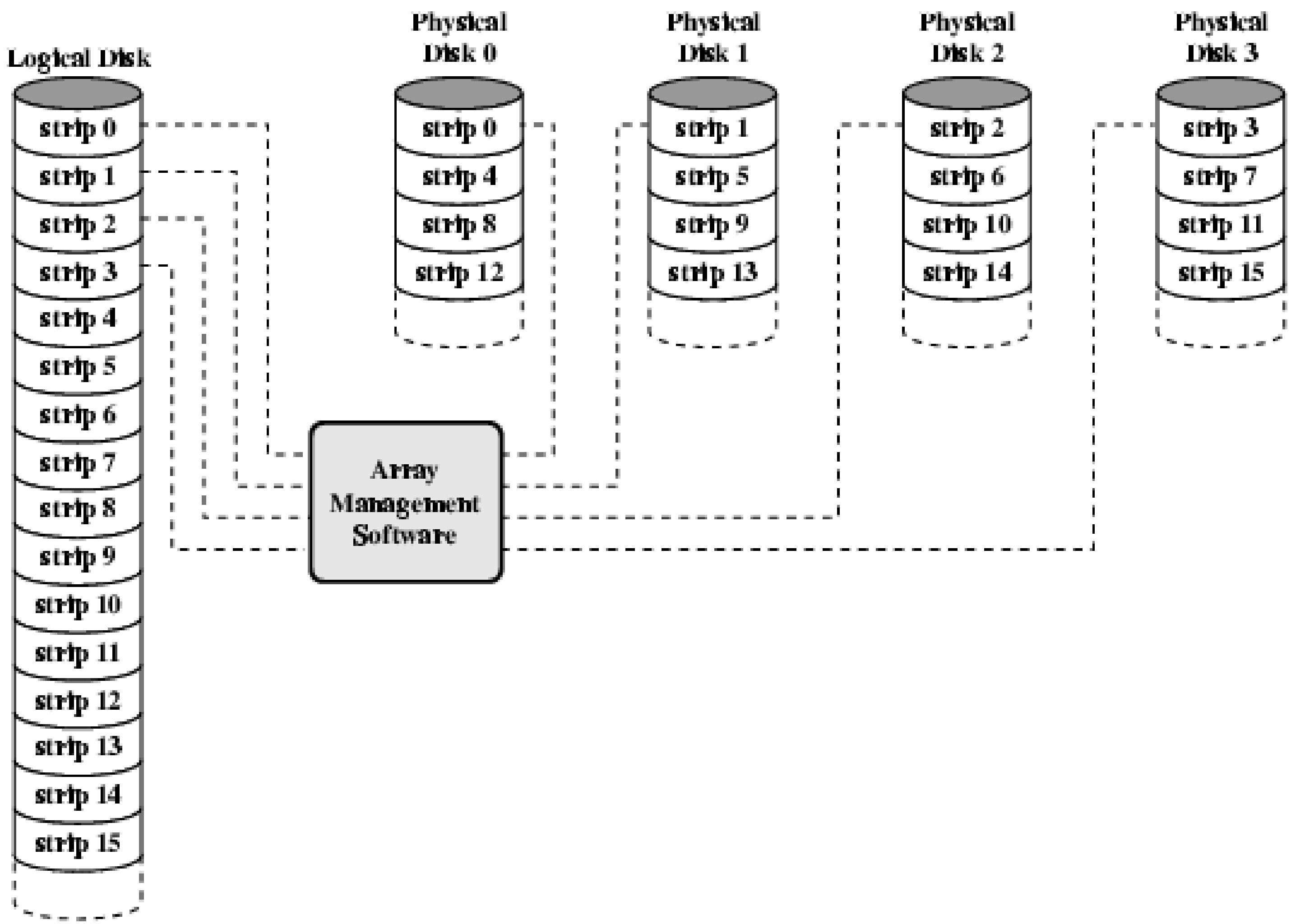
Advantages:

- **High data transfer :**Multiple data requests probably not on same disk
- **High I/O request rate:** Disks seek in parallel.

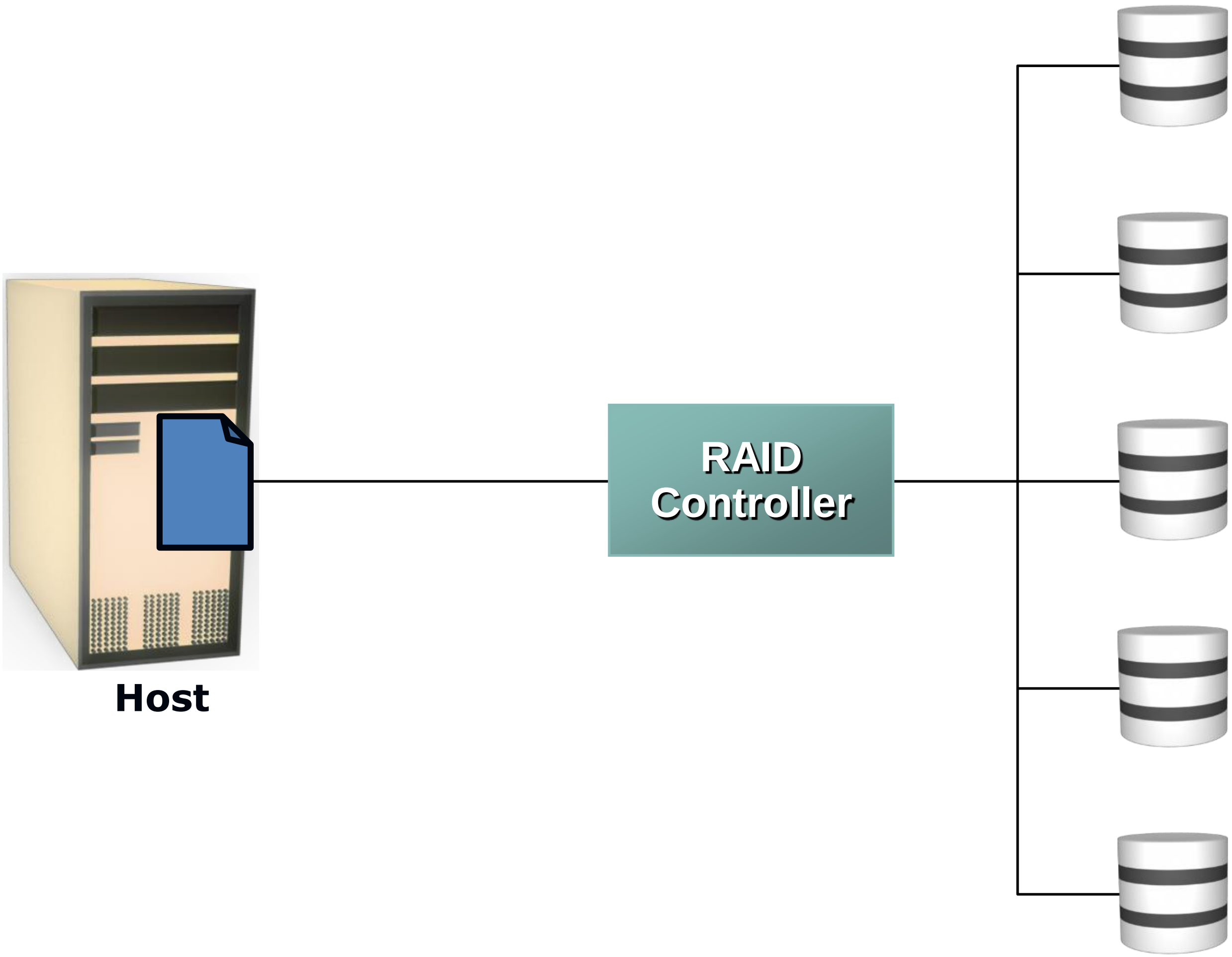
Draw Backs:

- Not a "True" RAID because it is NOT fault-tolerant.
- The failure of just one drive will result in all data in an array being lost.

Data mapping for RAID 0



RAID 0



- A RAID 0 can be created with disks of differing sizes, but the storage space added to the array by each disk is limited to the size of the smallest disk.

  For example, if a 500 GB disk is striped together with a 400 GB disk, the size of the array will be 800 GB.

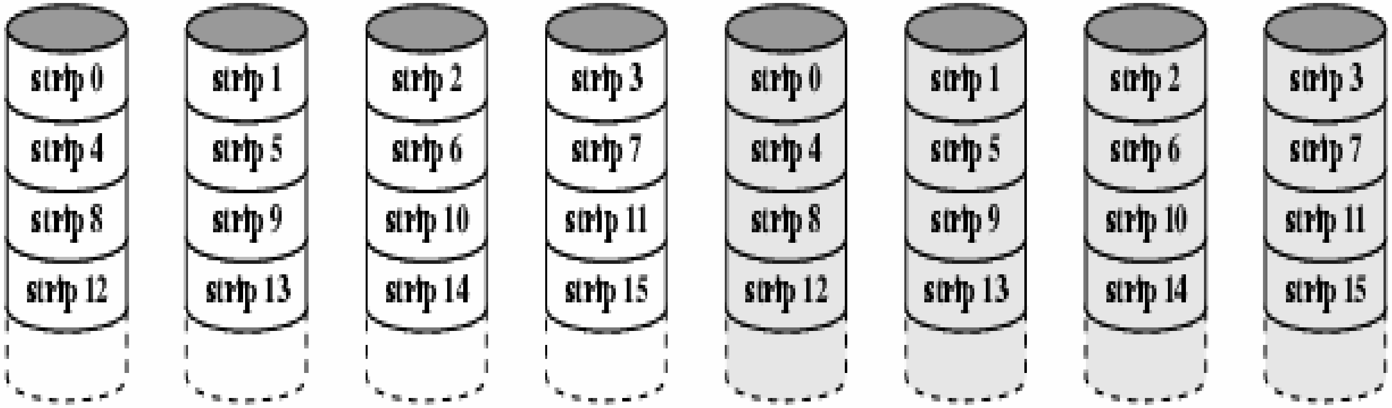
RAID 1

- Mirrored Disks (No parity calculation)
- Data is striped across disks.
- 2 copies of each stripe on separate disks.

Advantages:
- Read from either.
- Recovery is simple as data is stored in mirror disk

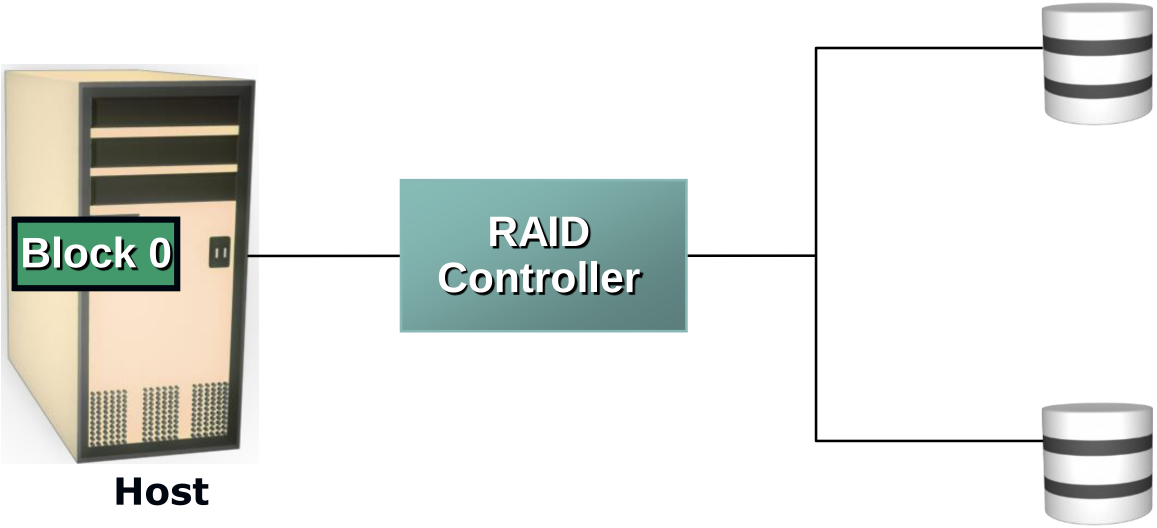
Draw back:
- Expensive (data is stored twice so double capacity than RAID 0)
- Write performance breaks down

Data mapping for RAID 1



(b) RAID 1 (mirrored)

RAID 1

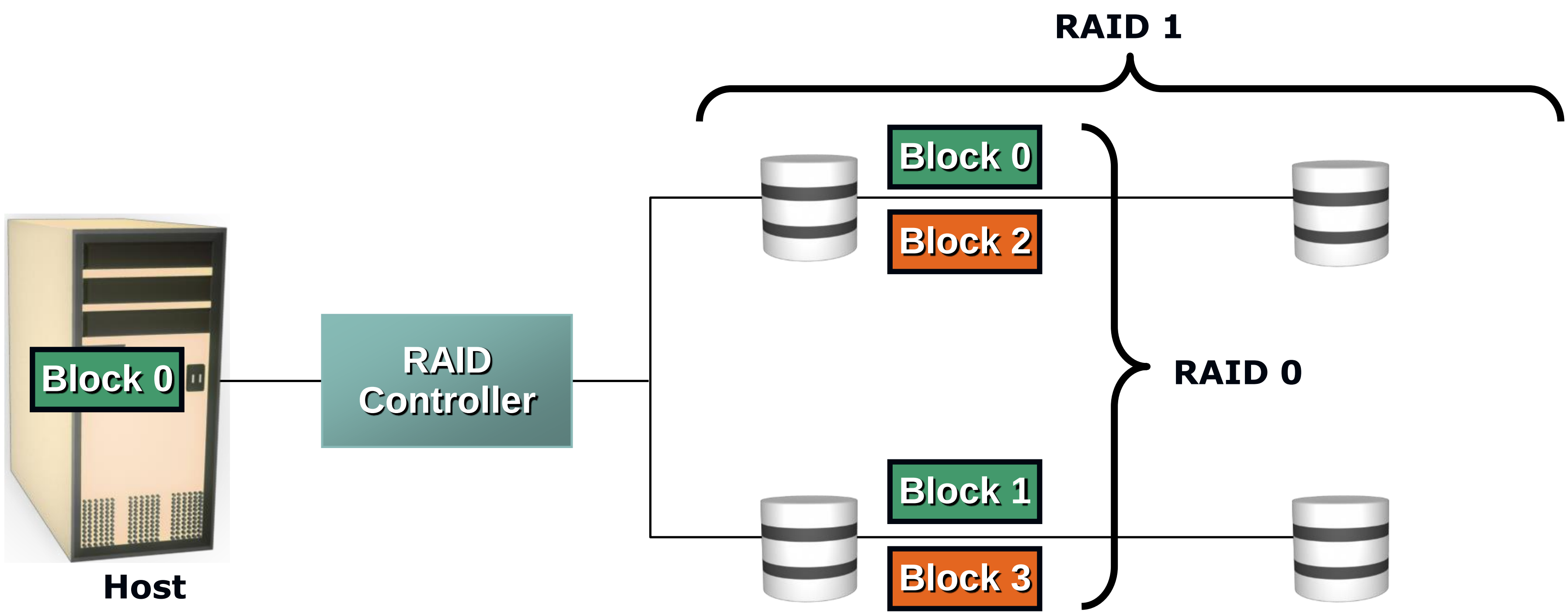


Note:

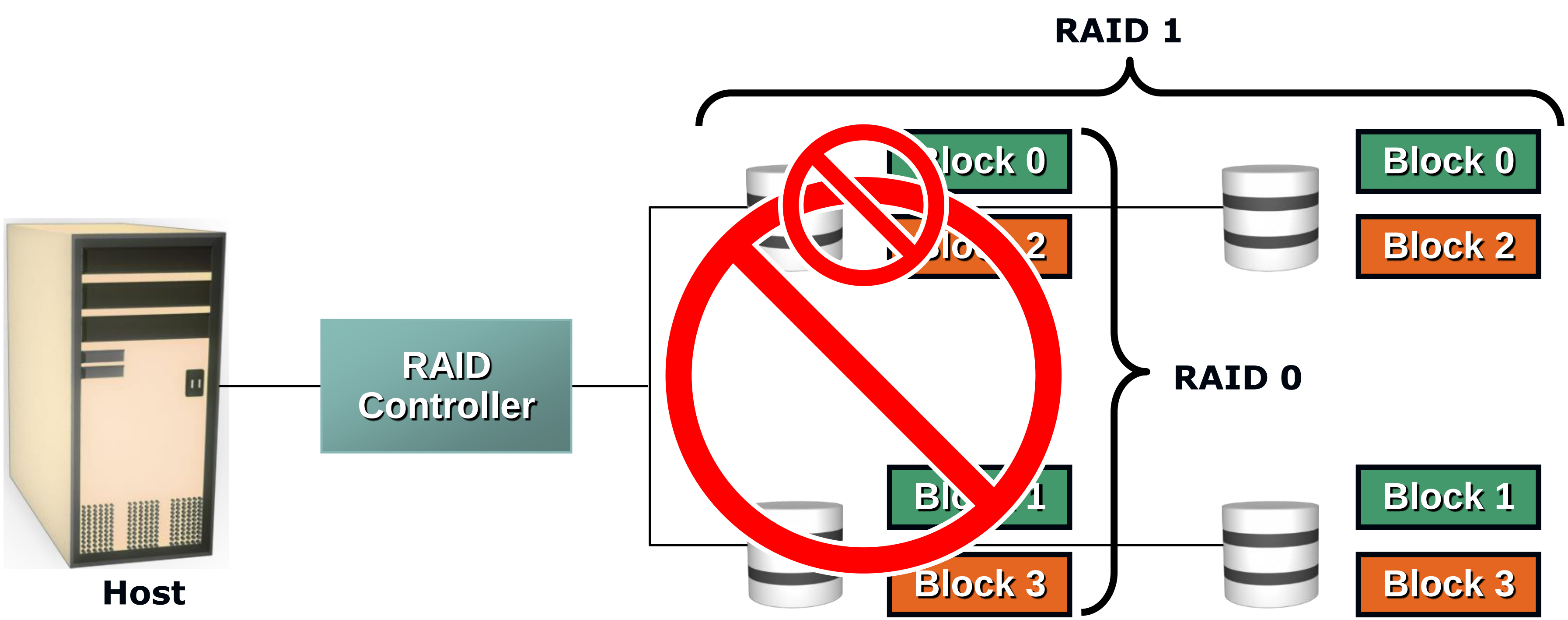
- Mirroring can be implemented with striped RAID by mirroring entire stripes of disks to stripes on other disks.

- This is called as nested RAID
  - RAID 0/1 (mirrored stripe)
  - RAID 1/0 or RAID 10 (striped mirror)

Nested RAID – 0+1 (Striping and Mirroring)



Nested RAID – 0+1 (Striping and Mirroring)



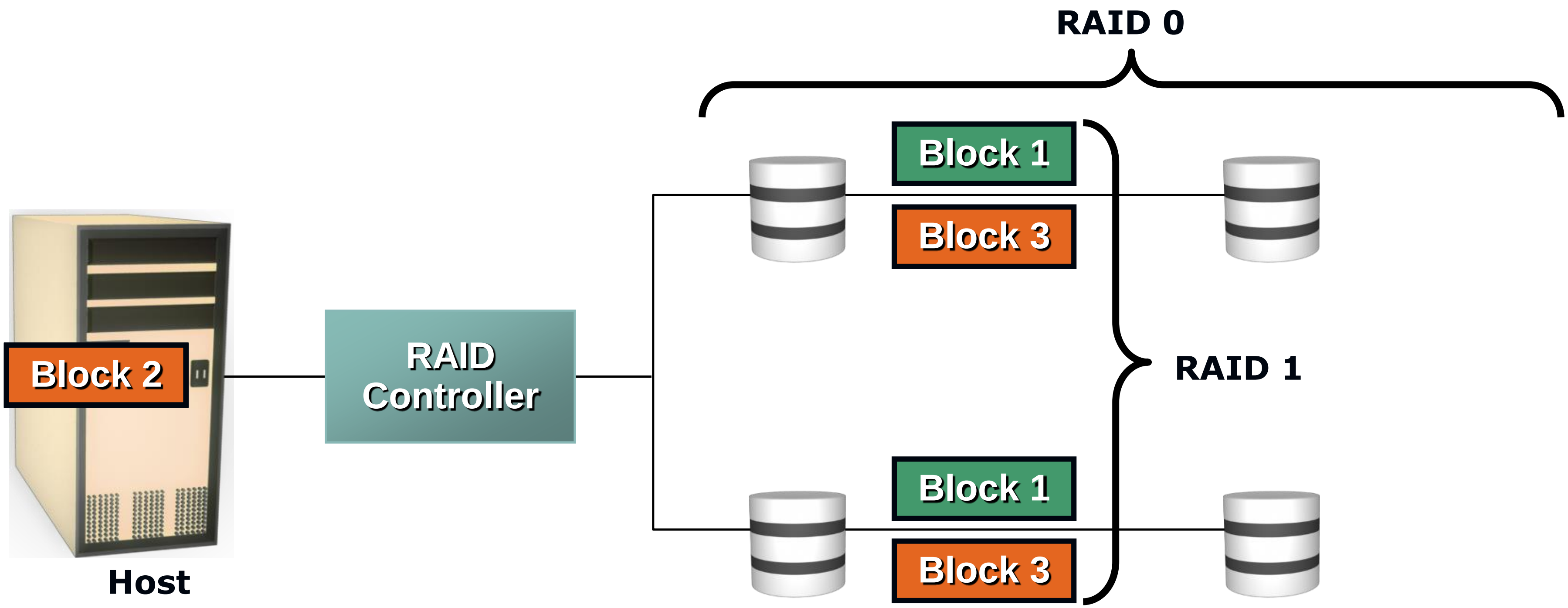
Disadv:

- Unnecessary I/O load on surviving disks
- Vulnerable to a second disk failure

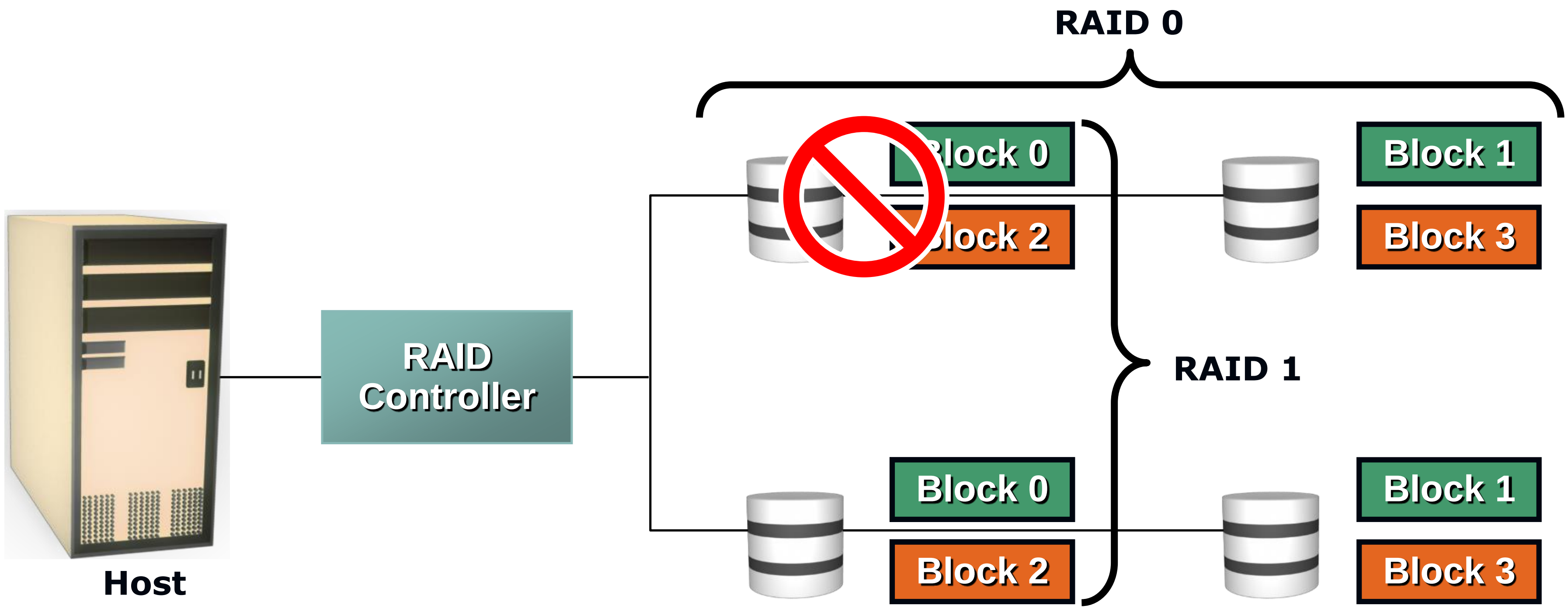
RAID 10

- Both offers identical benefits
- Rebuild operations in the case of disk failure differ between the two
- When replacing a failed drive, only the mirror is rebuilt

Nested RAID – 1+0 (Mirroring and Striping)



Nested RAID – 1+0 (Mirroring and Striping)

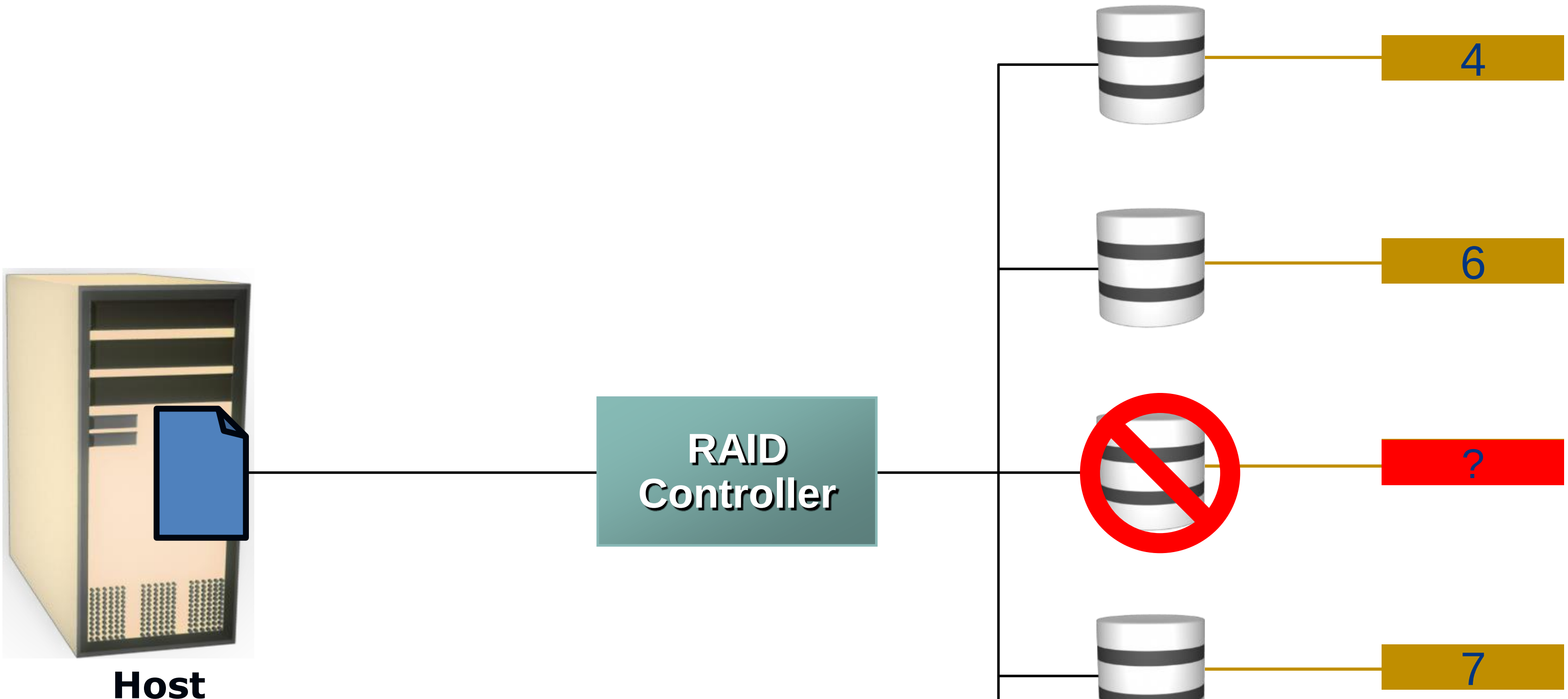


Parity

- It is a method of protecting striped data from hard disk failure without the cost of mirroring
- Parity RAID is less expensive than mirroring (i.e., parity overhead is only a fraction of total capacity)
- Parity info can be stored in dedicated disk or can be distributed over all the disk
- Uses XOR to compute parity

- Disadv: parity computation (data updation)

RAID with Parity



The middle drive fails:
Parity calculation 4 + 6 + 1 + 7 = 18

4 + 6 + ? + 7 = 18

? = 18 – 4 – 6 – 7

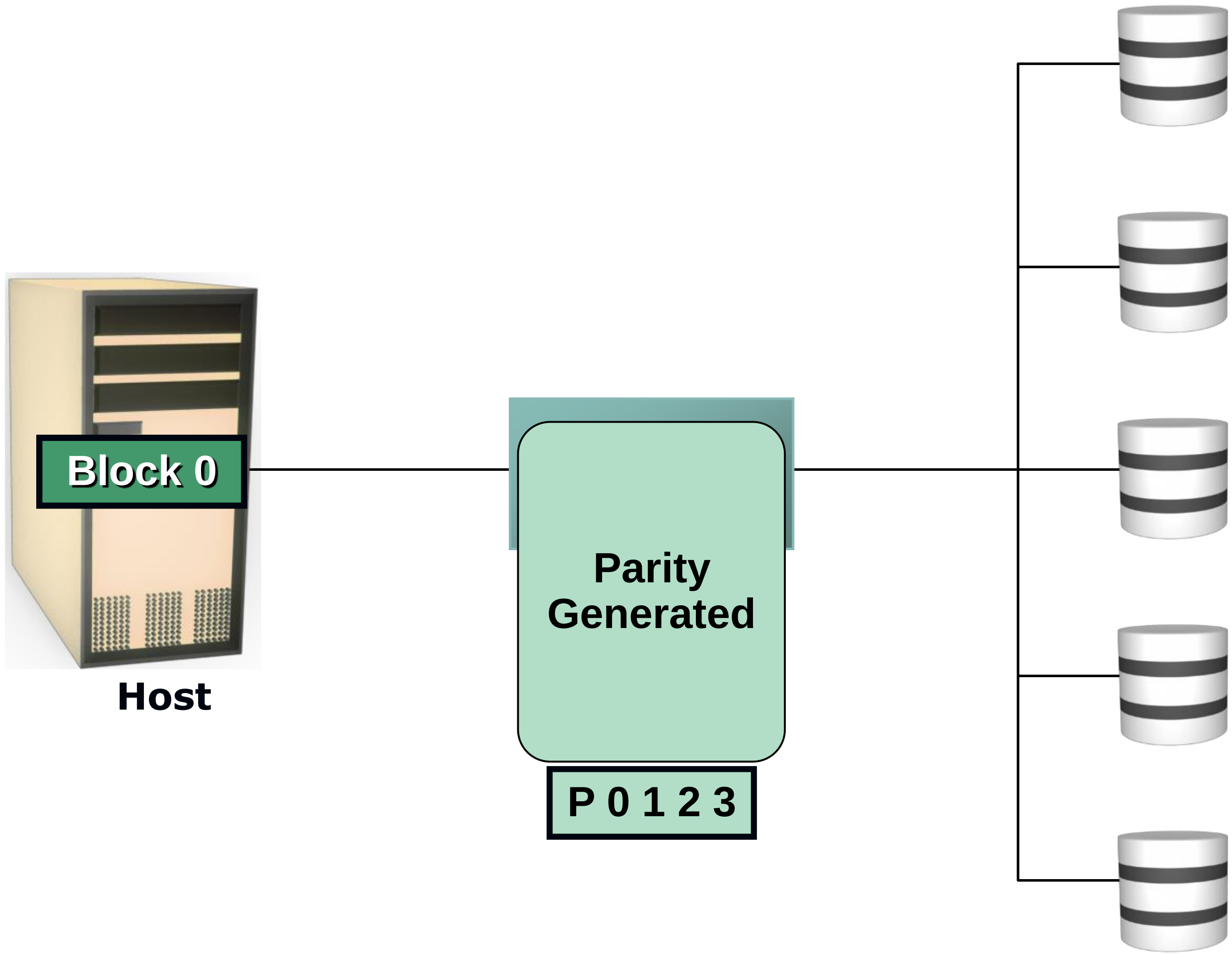
? = 1

Parity Disk

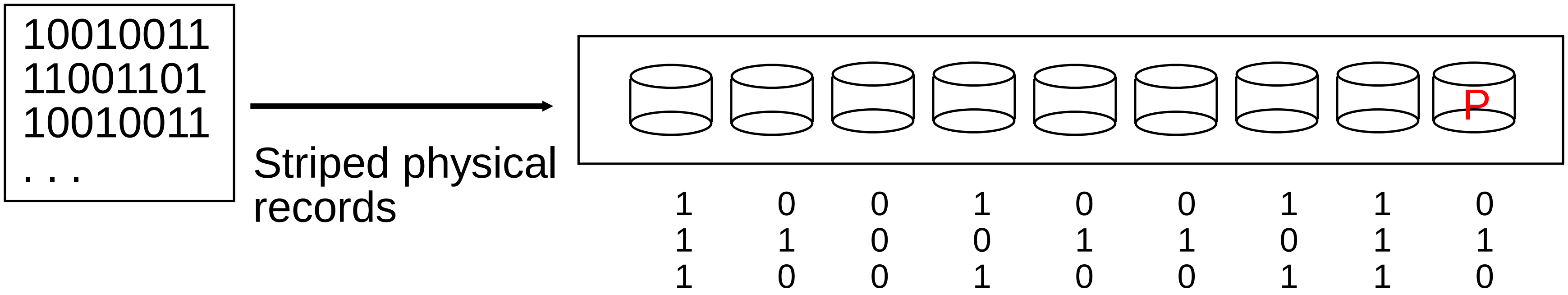
RAID 3

- Only one parity disk, no matter how large the array.
- Simple parity bit for each set of corresponding bits.
- Data on failed drive can be reconstructed from surviving data and parity information.
- Very high transfer rates.
- Drawback
  - Controller design is fairly complex

RAID 3

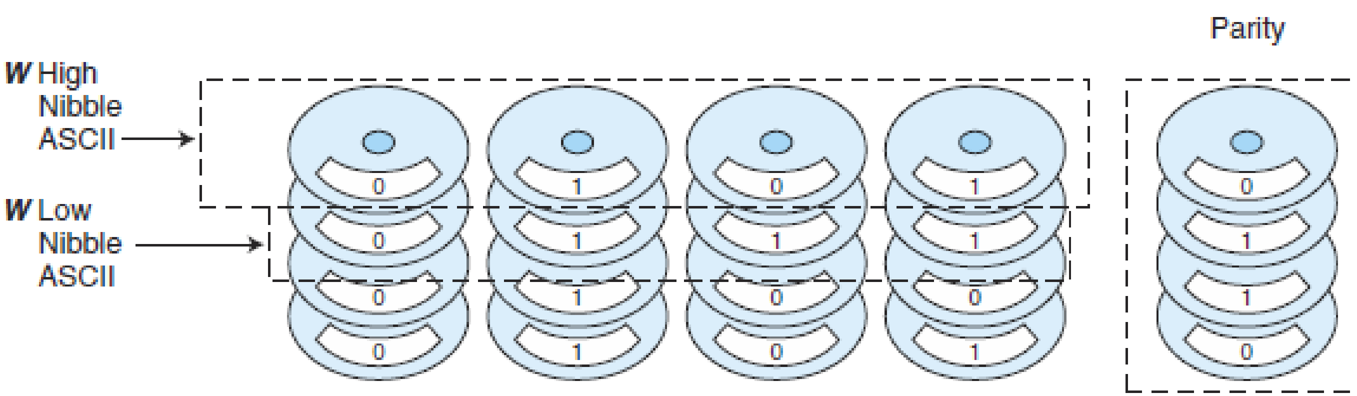


Example



Logical record

Physical record

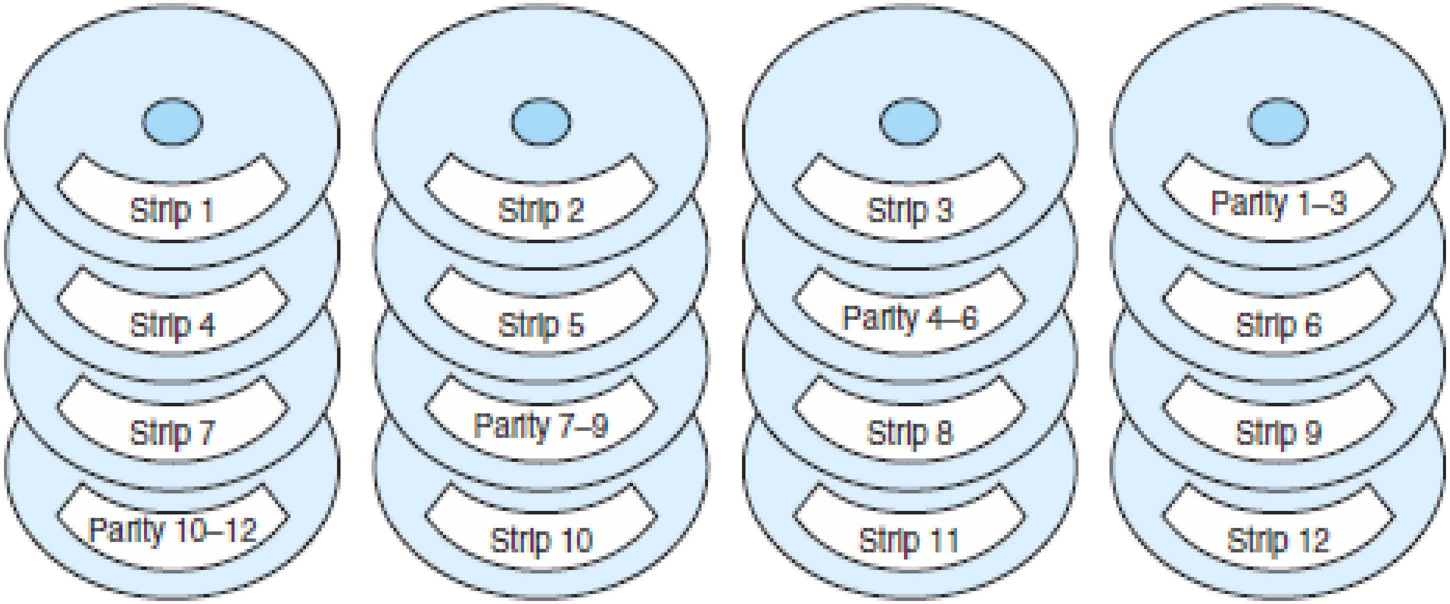


- RAID-3 is most useful for environments where large blocks of data would be read or written, such as with image or video processing.

RAID 5

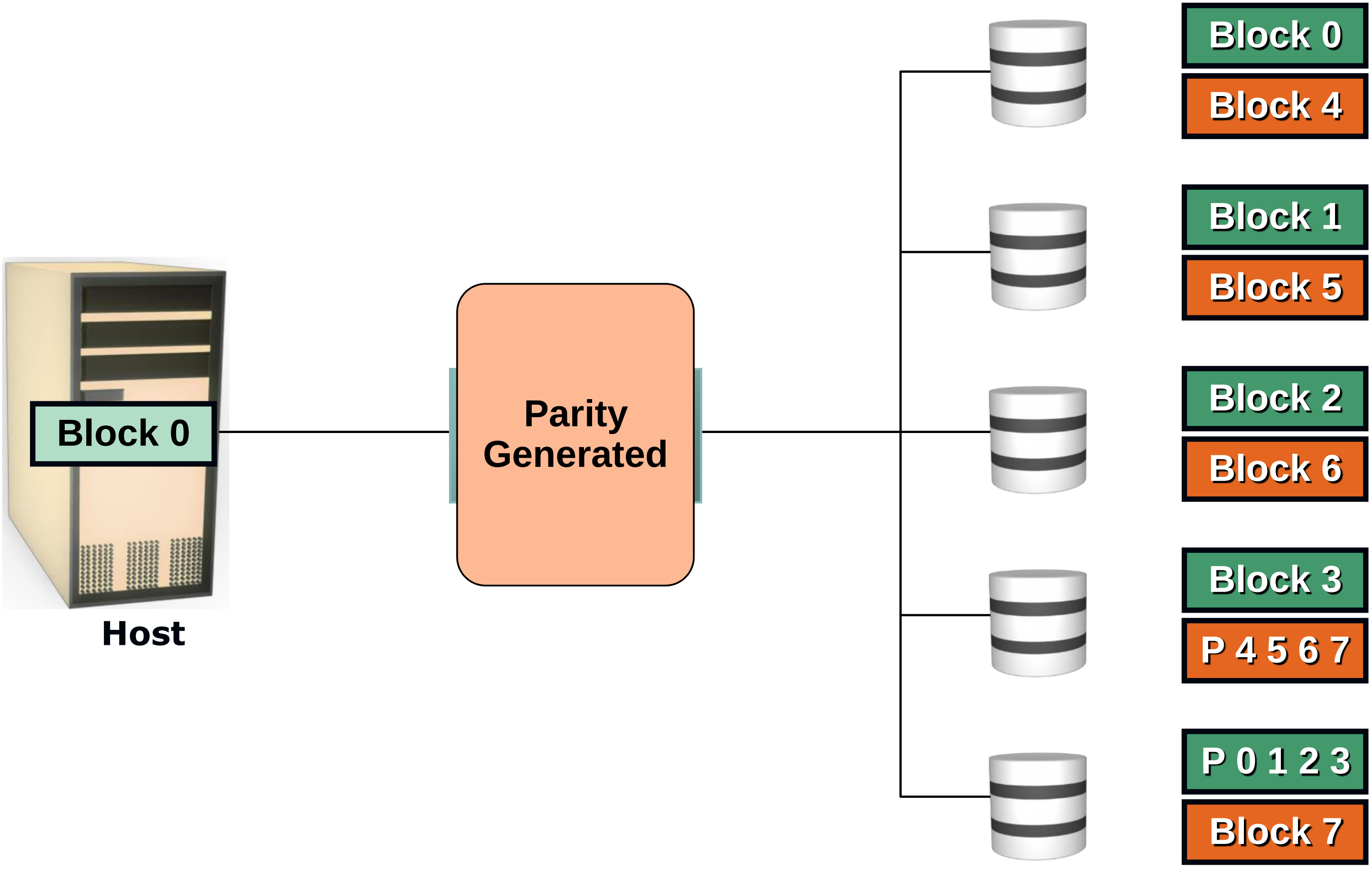
- Eliminate the parity disk bottleneck of previous RAID’s
- Parity across all disks (for load balancing)
- Round robin allocation for parity strips.
- Commonly used in network servers.
- Because some requests can be serviced concurrently, RAID-5 provides the best read throughput of all of the parity models and gives acceptable throughput on write operations

- Recommended applications include file and application servers, email and news servers, database and Web servers.



RAID-5, Data Striping with Distributed Parity

RAID 5, Data updation



- Drawback
  - Disk failure has a medium impact on throughput.
  - – Most complex controller design.
  - – Difficult to rebuild in the event of a disk failure (as compared to RAID level 1).
  - – Every strip updating involves 2 read & 2 write operation

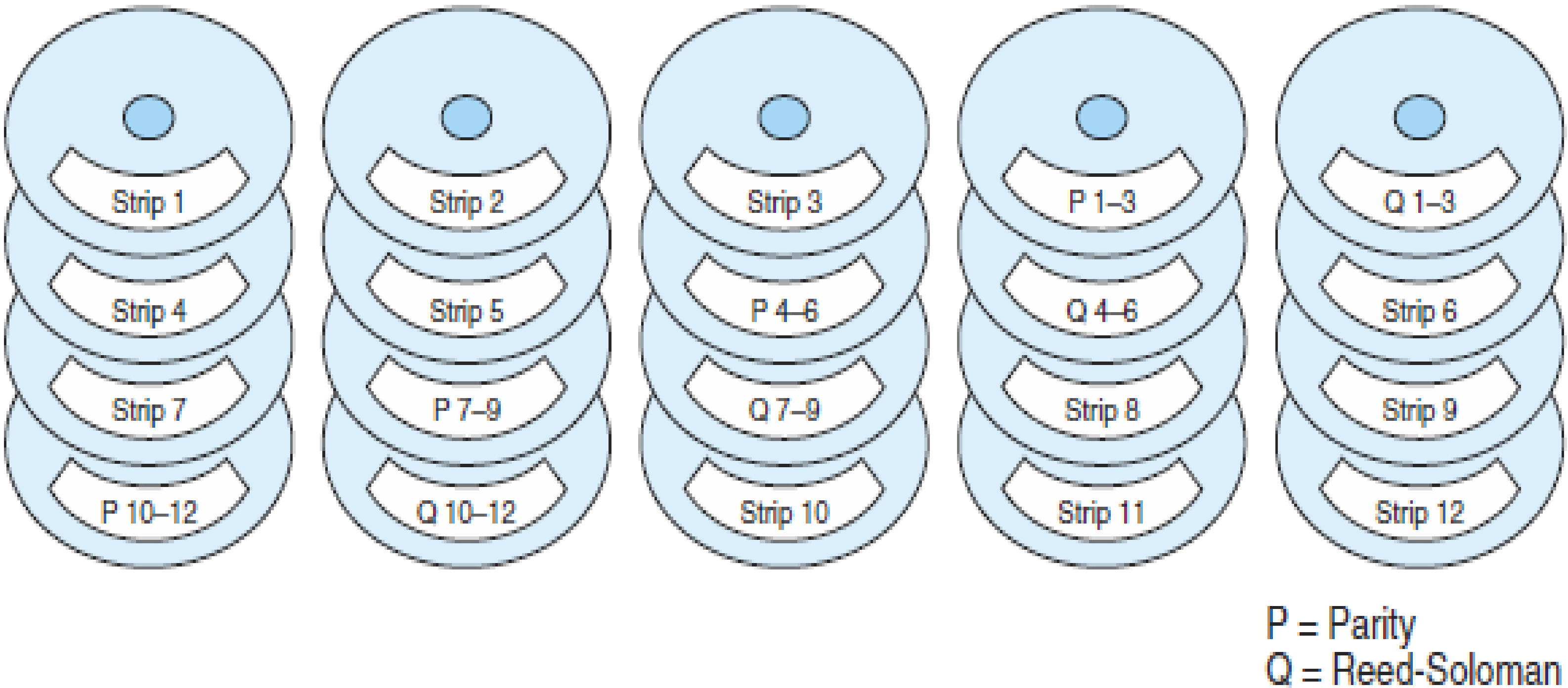
RAID 6

Limitations of RAID 0-5:

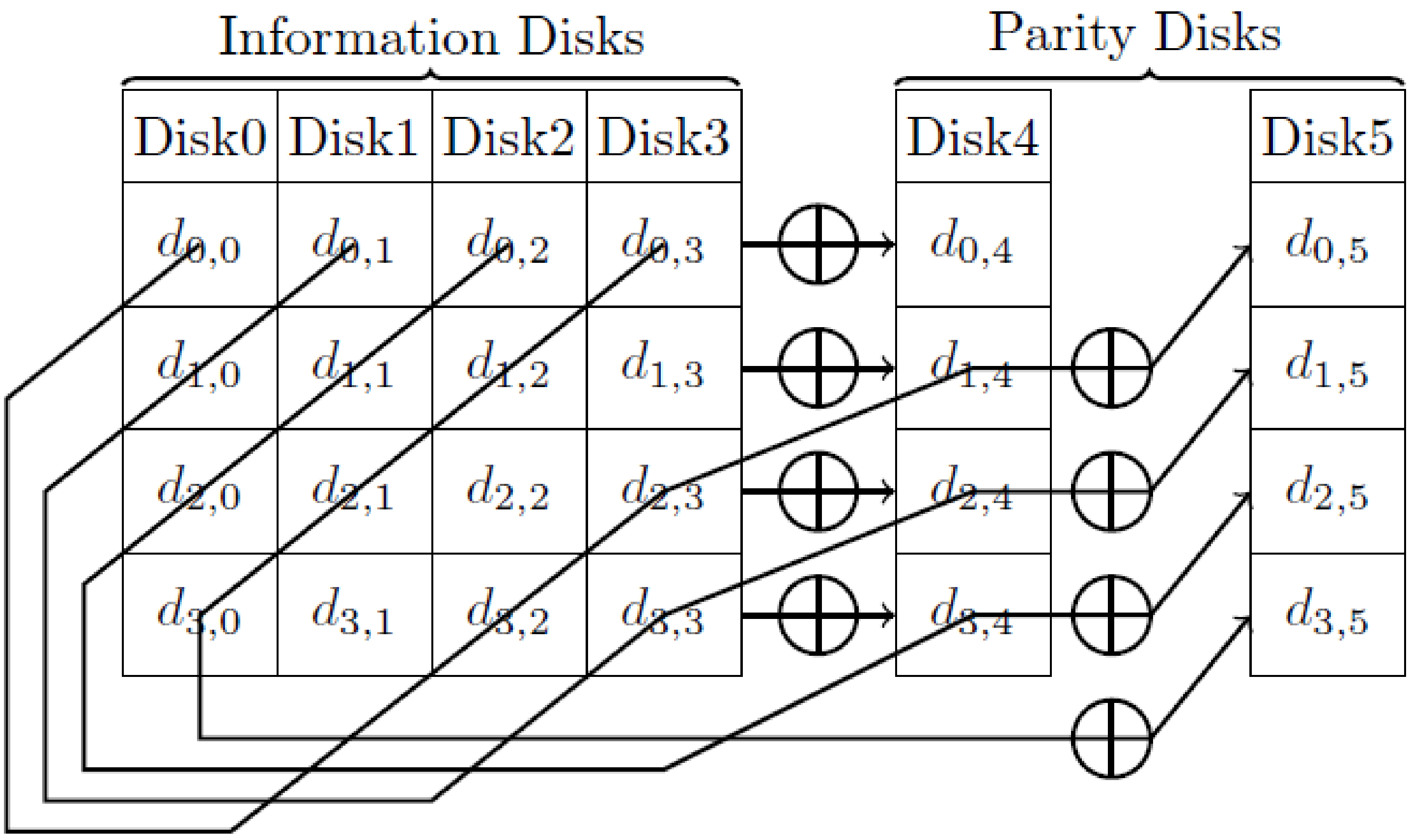
- All of the RAID systems discussed can tolerate at most one disk failure at a time.
- Disk drive failures in large systems tend to come in clusters.
- Systems that require high availability must be able to tolerate more than one concurrent drive failure

Solution is RAID 6:

- Two parity calculations, second being Reed Soloman error correcting codes (protects against two disk failures).
- Parity stored in separate blocks on different disks.
- User requirement of N disks needs N+2.
- High data availability.
- 3 disks have to fail within the MTTR interval to cause data to be lost.



RAID-6, Data Striping with Dual Error Protection


Information Disks
Parity Disks
Disk0
Disk1
Disk2
Disk3
Disk4
Disk5
$d_{0,0}$
$d_{0,1}$
$d_{0,2}$
$d_{0,3}$
$d_{0,4}$
$d_{0,5}$
$d_{1,0}$
$d_{1,1}$
$d_{1,2}$
$d_{1,3}$
$d_{1,4}$
$d_{1,5}$
$d_{2,0}$
$d_{2,1}$
$d_{2,2}$
$d_{2,3}$
$d_{2,4}$
$d_{2,5}$
$d_{3,0}$
$d_{3,1}$
$d_{3,2}$
$d_{3,3}$
$d_{3,4}$
$d_{3,5}$

RAID 6-RP&DP (Row &Diagonal Parity)

Disk 0	Disk 1	Disk 2	Disk 3	RP	DP
1	0	1	0		
0	0	1	0		
1	1	0	0		
0	1	0	1		

Disk 0	Disk 1	Disk 2	Disk 3	RP	DP
1	0	1	0	**0**	
0	0	1	0	**1**	
1	1	0	0	**0**	
0	1	0	1	**0**	

Disk0	Disk1	Disk2	Disk3	RP	DP
1	0	1	0	0	**0**
0	0	1	0	1	**1**
1	1	0	0	0	**0**
0	1	0	1	0	**0**

Which disks failed??

- Both data disk
- One data disk and one parity disk(row parity)
- One data disk and one parity disk(diagonal parity)
- Both parity disks

How to recover??

Disk0	Disk1	Disk2	Disk3	RP	DP
1		1		0	**0**
0		1		1	**1**
1		0		0	**0**
0		0		0	**0**

- Drawback
  - More complex controller design.
  - Because of the two-dimensional parity, RAID-6 offers very poor write performance..

RAID Comparison

RAID	Min Disks	Storage Efficiency %	Cost	Read Performance	Write Performance
0	2	100	Low	Very good for both random and sequential read	Very good
1	2	50	High	Good Better than a single disk	Good Slower than a single disk, as every write must be committed to two disks
3	3	(n-1)*100/n where n= number of disks	Moderate	Good for random reads and very good for sequential reads	Poor to fair for small random writes Good for large, sequential writes
5	3	(n-1)*100/n where n= number of disks	Moderate	Very good for random reads Good for sequential reads	Fair for random write Slower due to parity overhead Fair to good for sequential writes
6	4	(n-2)*100/n where n= number of disks	Moderate but more than RAID 5	Very good for random reads Good for sequential reads	Good for small, random writes (has write penalty)
1+0 and 0+1	4	50	High	Very good	Good

RAID Exercise

Total IOPS at peak workload is 1200

Read/Write ratio 2:1

Calculate IOPS requirement at peak activity for

- RAID 1/0
- RAID 5

Sol

Total IOPS = 1200

Read / Writ ratio 2:1

- For RAID 1/0:

(1200x2/3) + (1200x(1/3)x2 = 800 + 800 = 1600 IOPS

- For RAID 5:

(1200x2/3)x2 + (1200x(1/3)x2 = 1600 + 800 = 2400 IOPS

Summary

- Storage as Service
  - ✓ **RAID**
    - ❖ **RAID 0**
    - ❖ **RAID 1**
    - ❖ **RAID 0/1**
    - ❖ **RAID 1/0**
    - ❖ **RAID 3**
    - ❖ **RAID 5**
    - ❖ **RAID 6**