Data Protection

What is RAID?

- RAID (redundant array of independent disks) is a way of storing the same data in different places on multiple hard disks or solid-state drives (SSDs) to protect data in the case of a drive failure.
- There are different RAID levels, however, and not all have the goal of providing redundancy.

How RAID works

- RAID works by placing data on multiple disks and allowing input/output (I/O) operations to overlap in a balanced way, improving performance.
- Because using multiple disks increases the mean time between failures, storing data redundantly also increases fault tolerance
- RAID arrays appear to the operating system (OS) as a single logical drive.

Implementation of RAID

.

- There are two types of RAID implementation, hardware and software
- Software RAID
- *Software RAID* uses host-based software to provide RAID functions.
- It is implemented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.
- Software RAID implementations offer cost and simplicity benefits when compared with hardware RAID.

limitations

- **Performance**
- **Supported features**
- **Operating system compatibility**

Hardware RAID

- In *hardware RAID* implementations, a specialized hardware controller is implemented either on the host or on the array.
- These implementations vary in the way the storage array interacts with the host.
- *Controller card RAID* is host-based hardware RAID implementation in which
- a specialized RAID controller is installed in the host and HDDs are connected to it.
- The RAID Controller interacts with the hard disks using a PCI bus.
- Manufacturers also integrate RAID controllers on motherboards.
- This integration reduces the overall cost of the system, but does not provide the flexibility required for high-end storage systems

- The external RAID controller is an array-based hardware RAID.
- It acts as an interface between the host and disks.
- It presents storage volumes to the host,which manage the drives using the supported protocol.
- Key functions of RAID controllers are:
- Management and control of disk aggregations
- Translation of I/O requests between logical disks and physical disks
- Data regeneration in the event of disk failures

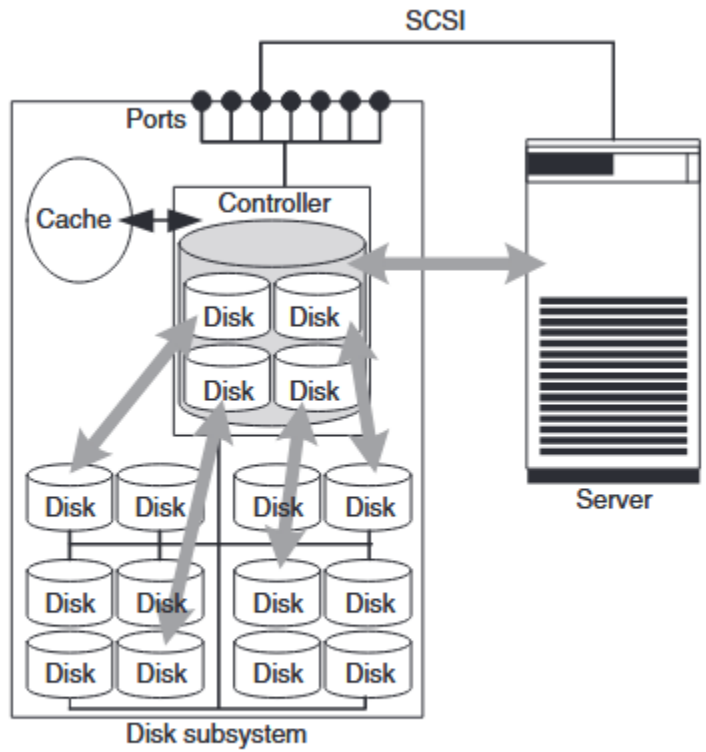
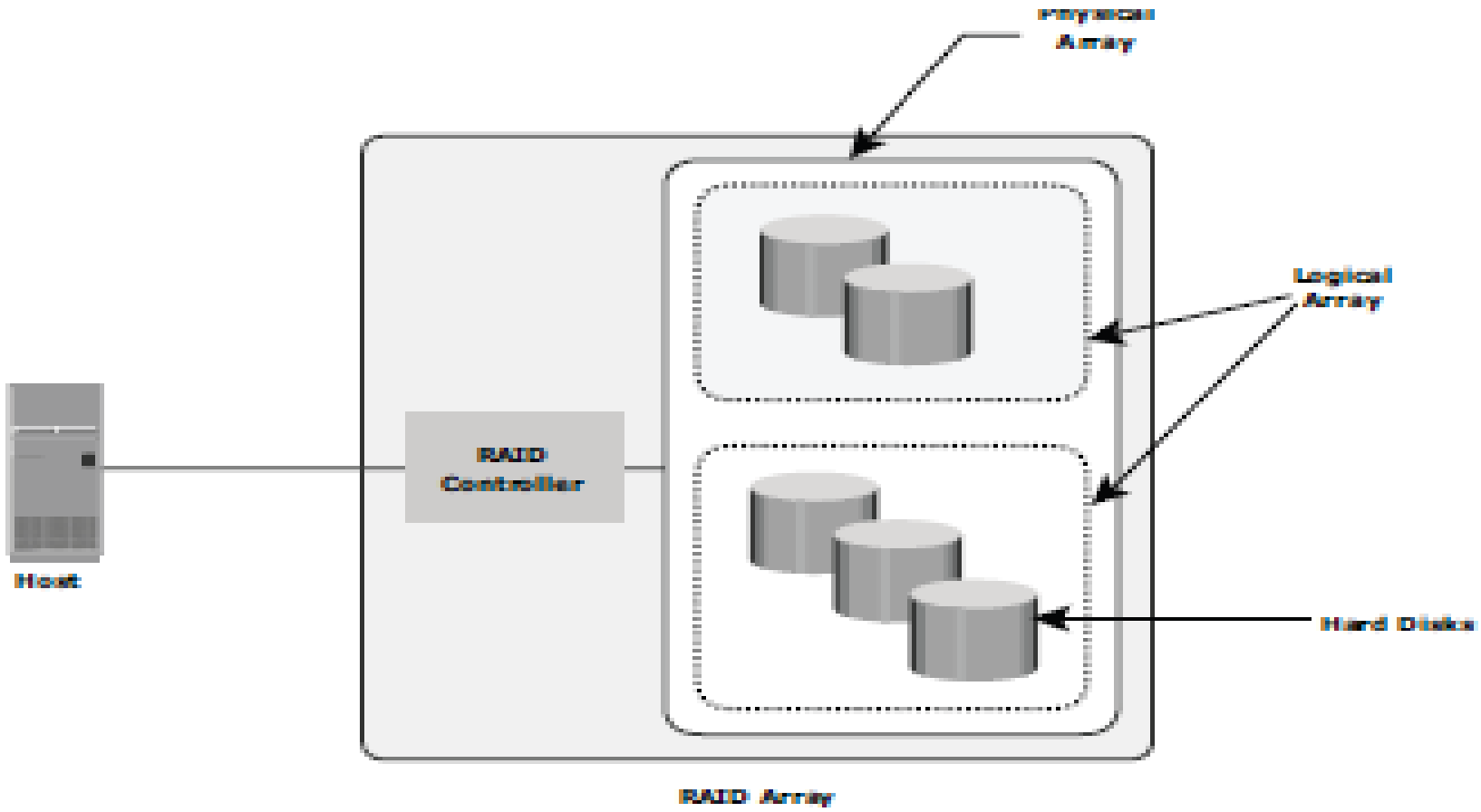


Figure 2.7 The RAID controller combines several physical hard disks to create a virtual hard disk. The server sees only a single virtual hard disk. The controller hides the assignment of

RAID Array Components

- A *RAID array* is an enclosure that contains a number of HDDs and the supporting hardware and software to implement RAID.
- HDDs inside a RAID array are usually contained in smaller sub-enclosures. These sub-enclosures, or *physical arrays*, hold a fixed number of HDDs, and may also include other supporting hardware, such as power supplies.
- A subset of disks within a RAID array can be grouped to form logical associations called *logical arrays*, also known as a *RAID set* or a *RAID group*

RAID Array Components



igure 3-1: Components of RAID array

- Logical arrays are comprised of logical volumes (LV).
- The operating system recognizes the LVs as if they are physical HDDs managed by the RAID controller.
- The number of HDDs in a logical array depends on the RAID level used.
- Configurations could have a logical array with multiple physical arrays or a physical array with multiple logical arrays

RAID Levels

- RAID levels are defined on the basis of
- striping,
- mirroring,
- parity techniques.
- These techniques determine the data availability and performance characteristics of an array.
- Some RAID arrays use one technique, whereas others use a combination of techniques.
- Application performance and data availability requirements determine the RAID level selection.

Striping

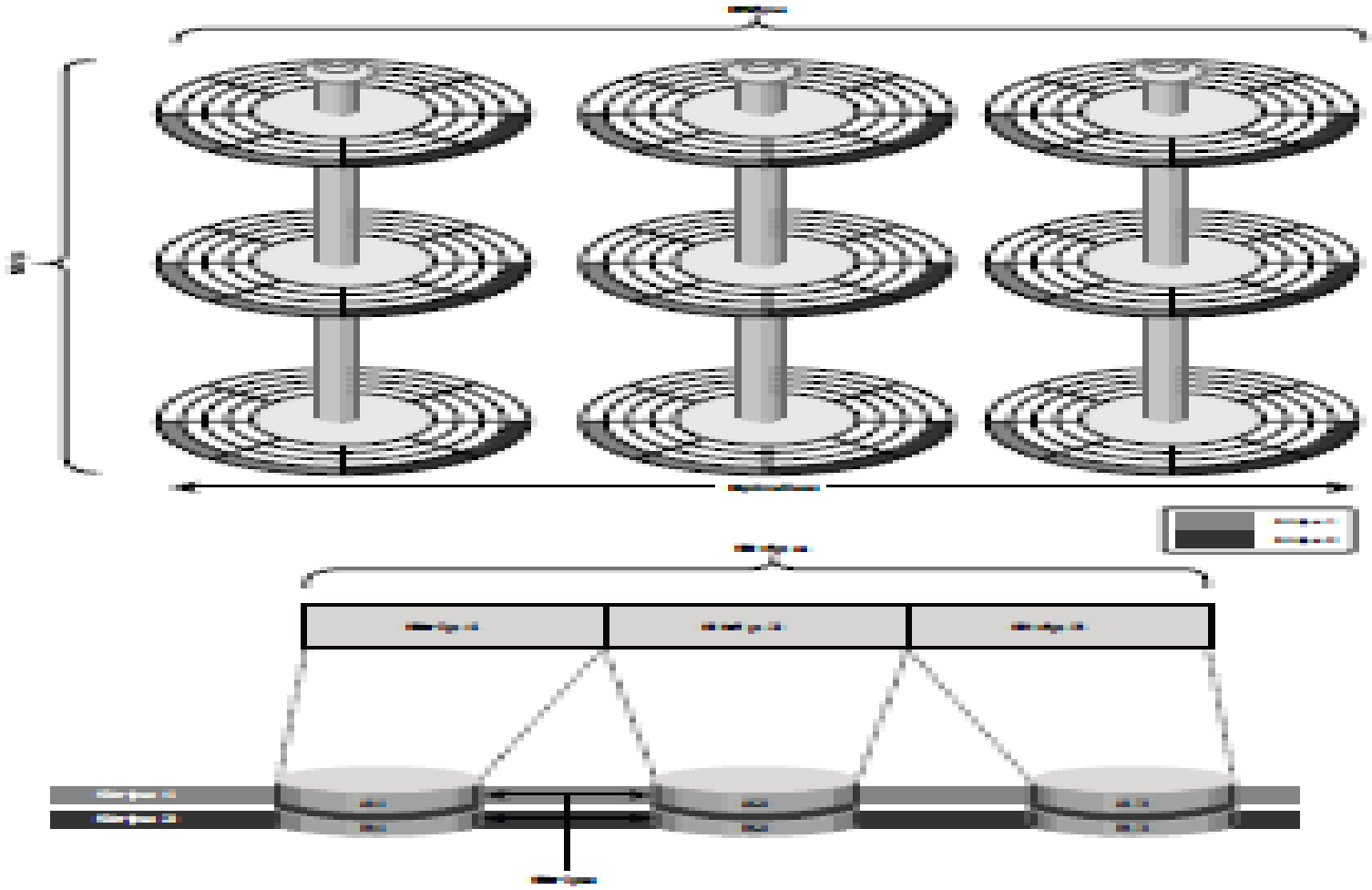


Figure 3-2: Striped RAID set

- A RAID set is a group of disks.
- Within each disk, a predefined number of contiguously addressable disk blocks are defined as *strips.*
- *The set of aligned strips* that spans across all the disks within the RAID set is called a *stripe.*
- *Above* shows physical and logical representations of a striped RAID set.

- *Strip size (also called stripe depth) describes the number of blocks in a strip,*
- and is the maximum amount of data that can be written to or read from a single HDD in the set before the next HDD is accessed, assuming that the accessed data starts at the beginning of the strip.
- Note that all strips in a stripe have the same number of blocks, and decreasing strip size means that data is broken into smaller pieces when spread across the disks.

- Stripe size is a multiple of strip size by the number of HDDs in the RAID set.
- *Stripe width refers to the number of data strips in a stripe.*
- Striped RAID does not protect data unless parity or mirroring is used.
- However, striping may significantly improve I/O performance. Depending on the type of RAID implementation, the RAID controller can be configured to access data across multiple HDDs simultaneously.

Mirroring

- Mirroring is a technique where by data is stored on two different HDDs, yield-ing two copies of data.
- In the event of one HDD failure, the data is intact on the surviving HDD and the controller continues to service the host's data requests from the surviving disk of a mirrored pair
- When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair.
- This activity is transparent to the host.
- In addition to providing complete data redundancy, mirroring enables faster recovery from disk failure.
- However, disk mirroring provides only data protection and is not a substitute for data backup.
- Mirroring constantly captures changes in the data, whereas a backup captures point-in-time images of data.

Mirroring

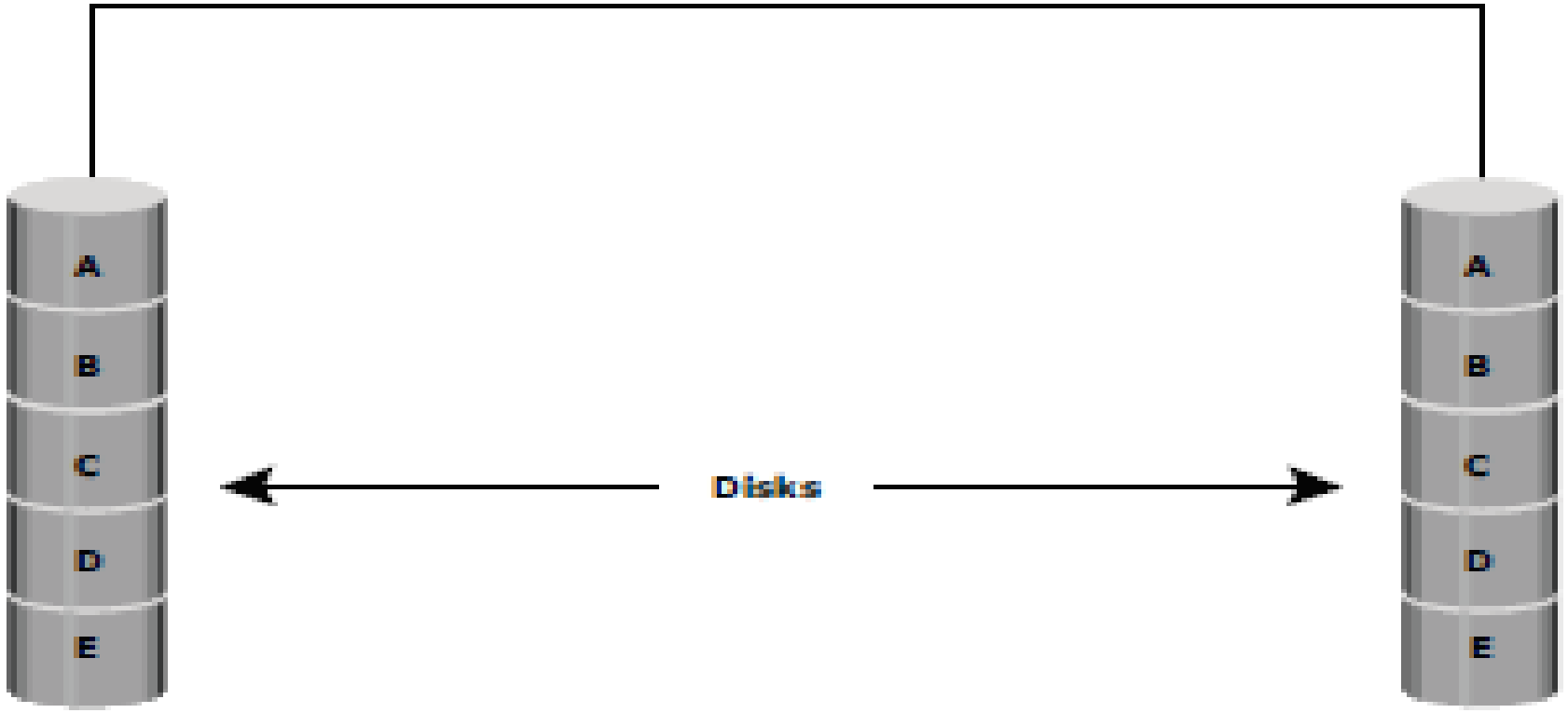


Figure 3-3: Mirrored disks in an array

- Mirroring involves duplication of data — the amount of storage capacity needed is twice the amount of data being stored.
- Therefore, mirroring is considered expensive and is preferred for mission-critical applications that cannot afford data loss.
- Mirroring improves read performance because read requests can be serviced by both disks.
- However, write performance deteriorates, as each write request manifests as two writes on the HDDs.
- In other words, mirroring does not deliver the same levels of write performance as a striped RAID.

Parity

- *Parity* is a method of protecting striped data from HDD failure without the cost of mirroring
- An additional HDD is added to the stripe width to hold parity, a mathematical construct that allows re-creation of the missing data.
- Parity is a redundancy check that ensures full protection of data without maintaining a full set of duplicate data
- Parity information can be stored on separate, dedicated HDDs or distributed across all the drives in a RAID set.
- Figure 3-4 shows a parity RAID.
- The first four disks, labeled *D,* contain the data.
- The fifth disk, labeled *P*, stores the parity information, which in this case is the sum of the elements in each row.
- Now, if one of the *D*s fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value.

Parity

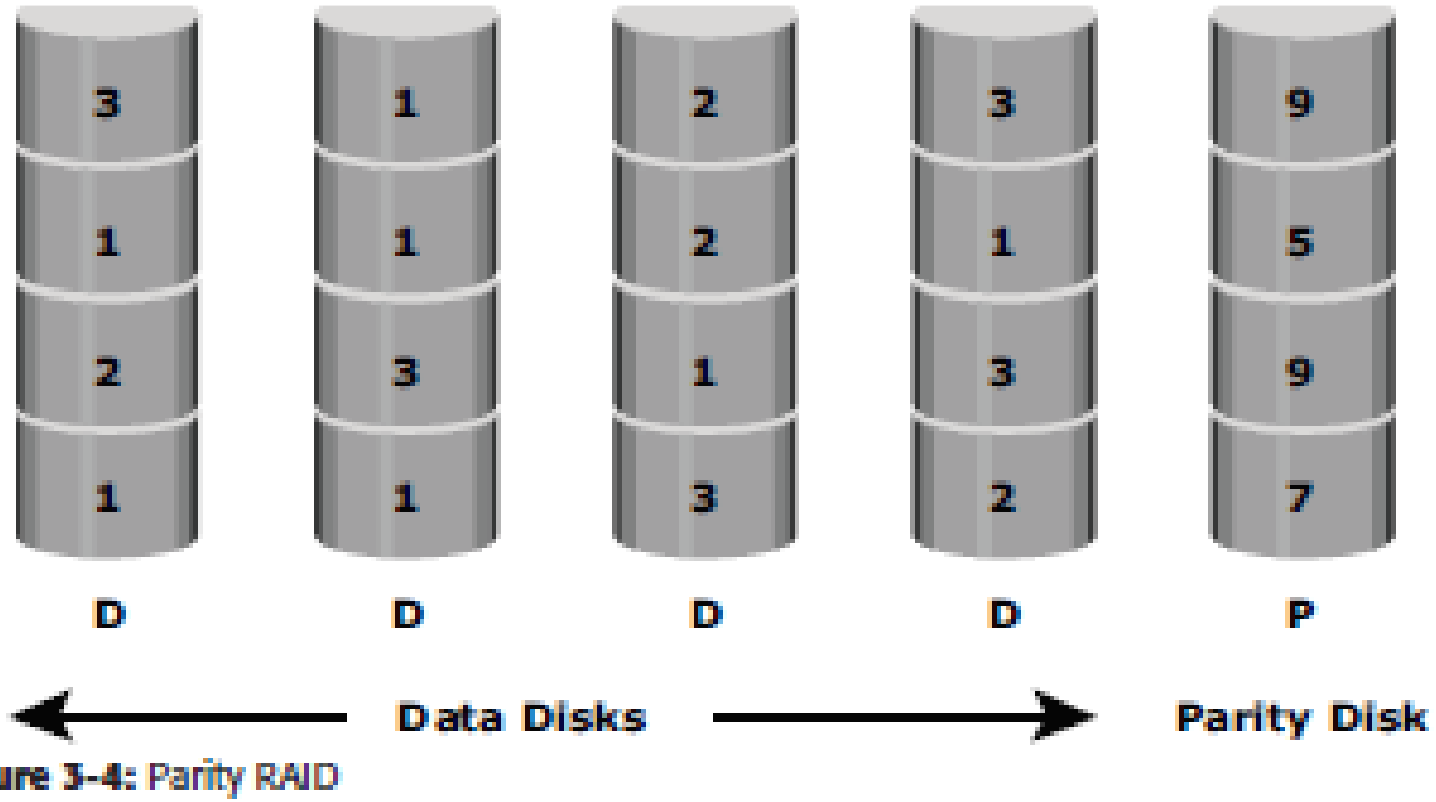


Figure 3-4: Parity RAID

RAID 0

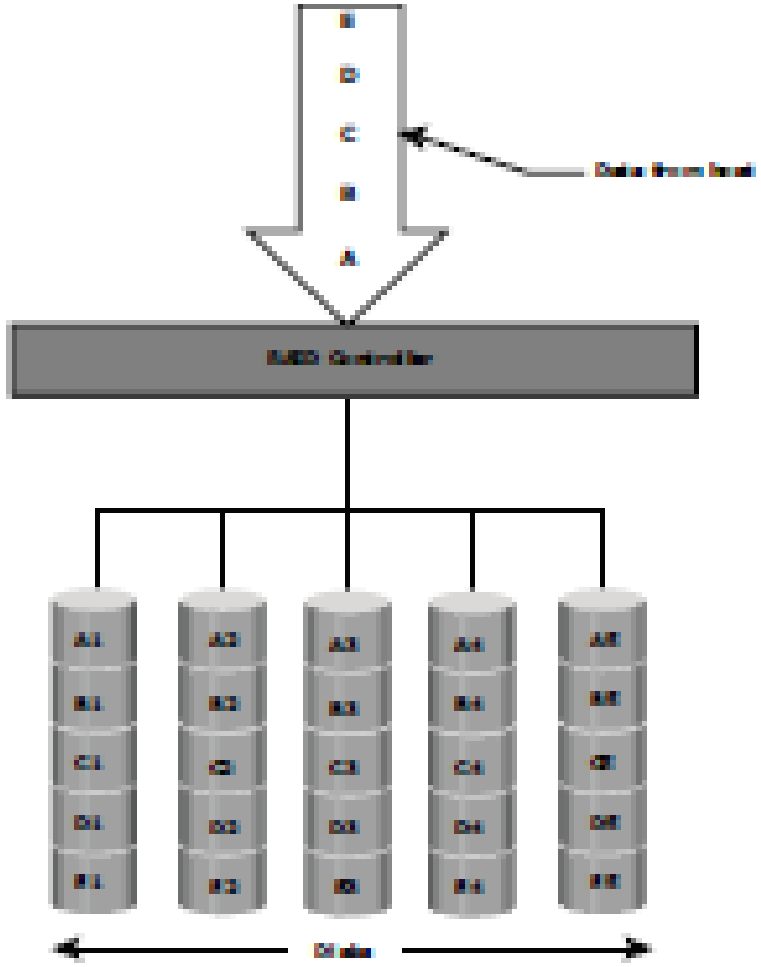


Figure 3-5: RAID 0

RAID 0

- In a RAID 0 configuration, data is striped across the HDDs in a RAID set.
- It utilizes the full storage capacity by distributing strips of data over multiple HDDs in a RAID set.
- To read data, all the strips are put back together by the controller.
- The stripe size is specified at a host level for software RAID and is vendor specific for hardware RAID.
- Figure shows RAID 0 on a storage array in which data is striped across 5 disks. When the number of drives in the array increases, performance improves because more data can be read or written simultaneously.
- RAID 0 is used in applications that need high I/O throughput.
- However, if these applications require high availability, RAID 0 does not provide data protection and availability in the event of drive failures

RAID 1

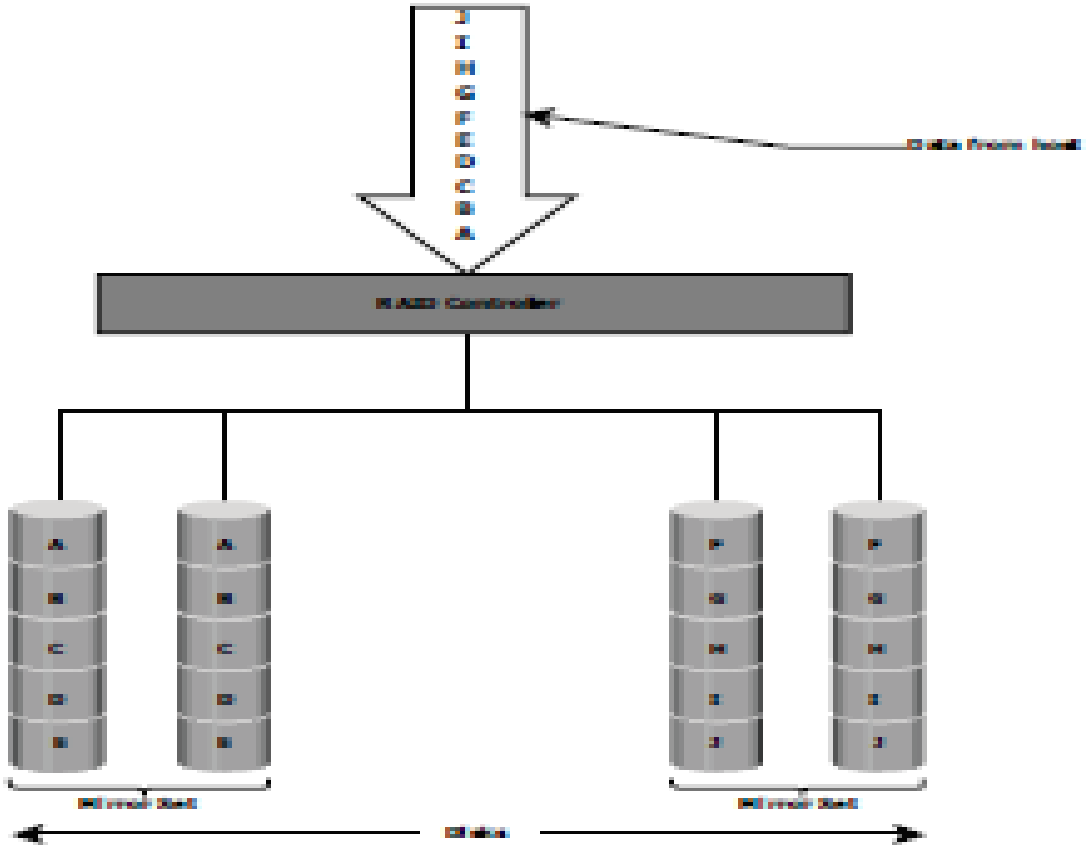


Figure 3-6: RAID 1

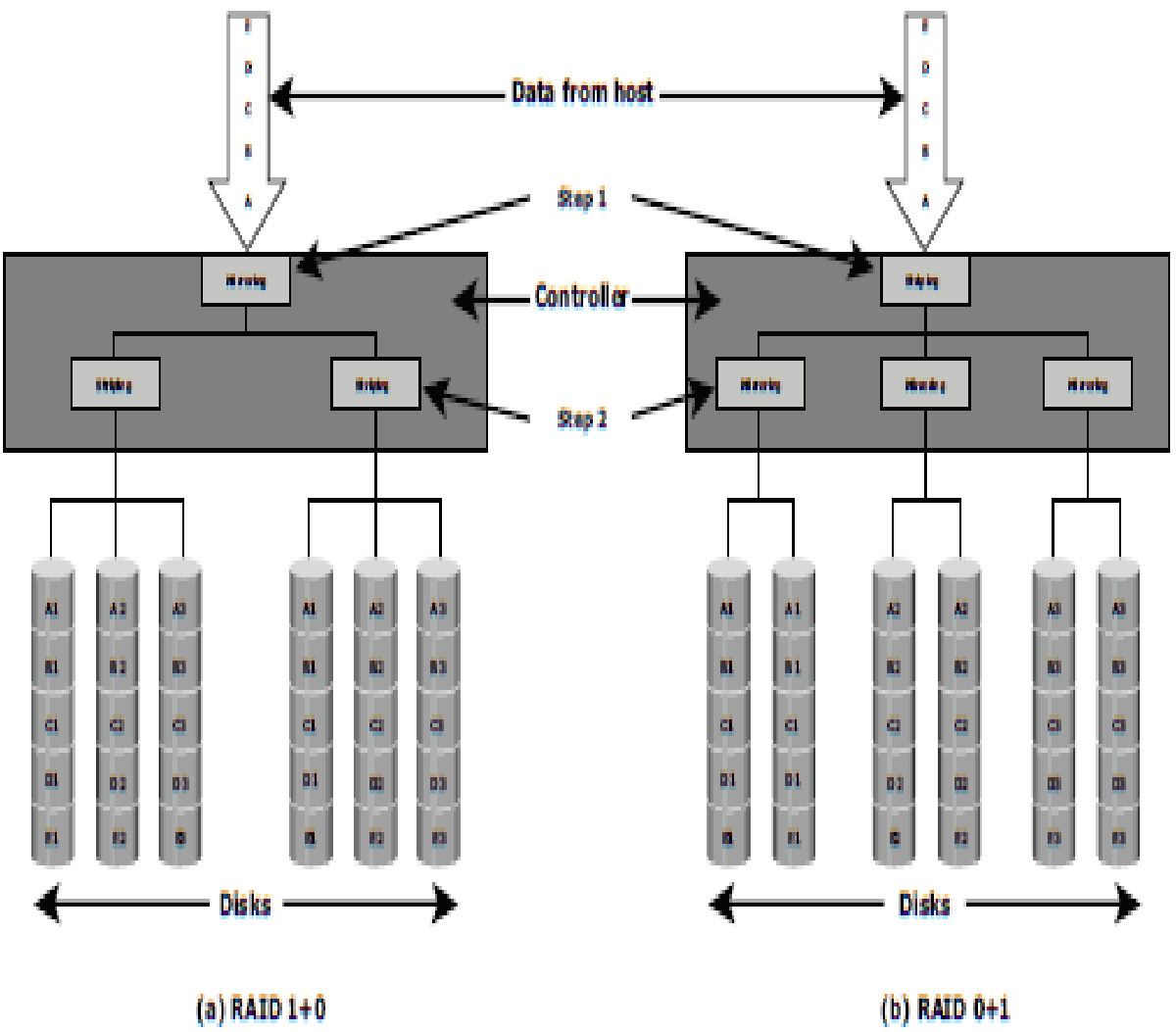
RAID 1

- In a RAID 1 configuration, data is mirrored to improve fault tolerance
- A RAID 1 group consists of at least two HDDs.
- As explained in mirroring, every write is written to both disks, which is transparent to the host in a hardware RAID implementation.
- In the event of disk failure, the impact on data recovery is the least among all RAID implementations.
- This is because the RAID controller uses the mirror drive for data recovery and continuous operation.
- RAID 1 is suitable for applications that require high availability.

Nested RAID

- Most data centers require data redundancy and performance from their RAID arrays.
- RAID 0+1 and RAID 1+0 combine the performance benefits of RAID 0 with the redundancy benefits of RAID 1.
- They use striping and mirroring techniques and combine their benefits.
- These types of RAID require an even number of disks, the minimum being four
- RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0.
- Similarly, RAID 0+1 is also known as RAID 01 or RAID 0/1.
- RAID 1+0 performs well for workloads that use small, random, write-intensive I/O.

Figure 3-7: Nested RAID



(a) RAID 1+0

(b) RAID 0+1

Some applications that benefit from RAID 1+0 include the following:

- High transaction rate Online Transaction Processing (OLTP)
- Large messaging installations
- Database applications that require high I/O rate, random access, and
- high availability

- A common misconception is that RAID 1+0 and RAID 0+1 are the same.
- Under normal conditions, RAID levels 1+0 and 0+1 offer identical benefits.
- However, rebuild operations in the case of disk failure differ between the two.
- RAID 1+0 is also called *striped mirror*.
- The basic element of RAID 1+0 is a mirrored pair, which means that data is first mirrored and then both copies of
- data are striped across multiple HDDs in a RAID set.
- When replacing a failed drive, only the mirror is rebuilt.
- In other words, the disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation.
- Data from the surviving disk is copied to the replacement disk.

- RAID 0+1 is also called *mirrored stripe*.
- The basic element of RAID 0+1 is a stripe.
- This means that the process of striping data across HDDs is performed
- initially and then the entire stripe is mirrored.
- If one drive fails, then the entire stripe is faulted.
- A rebuild operation copies the entire stripe, copying data from each disk in the healthy stripe to an equivalent disk in the failed stripe.
- This causes increased and unnecessary I/O load on the surviving disks and makes the RAID set more vulnerable to a second disk failure.

RAID 3

- RAID 3 stripes data for high performance and uses parity for improved fault
- tolerance.
- Parity information is stored on a dedicated drive so that data can be
- reconstructed if a drive fails.
- For example, of five disks, four are used for data and one is used for parity. Therefore, the total disk space required is 1.25 times the size of the data disks.
- RAID 3 **always** reads and writes complete stripes of data across all disks, as the drives operate in parallel.
- There are no partial writes that update one out of many strips in a stripe..

RAID 3

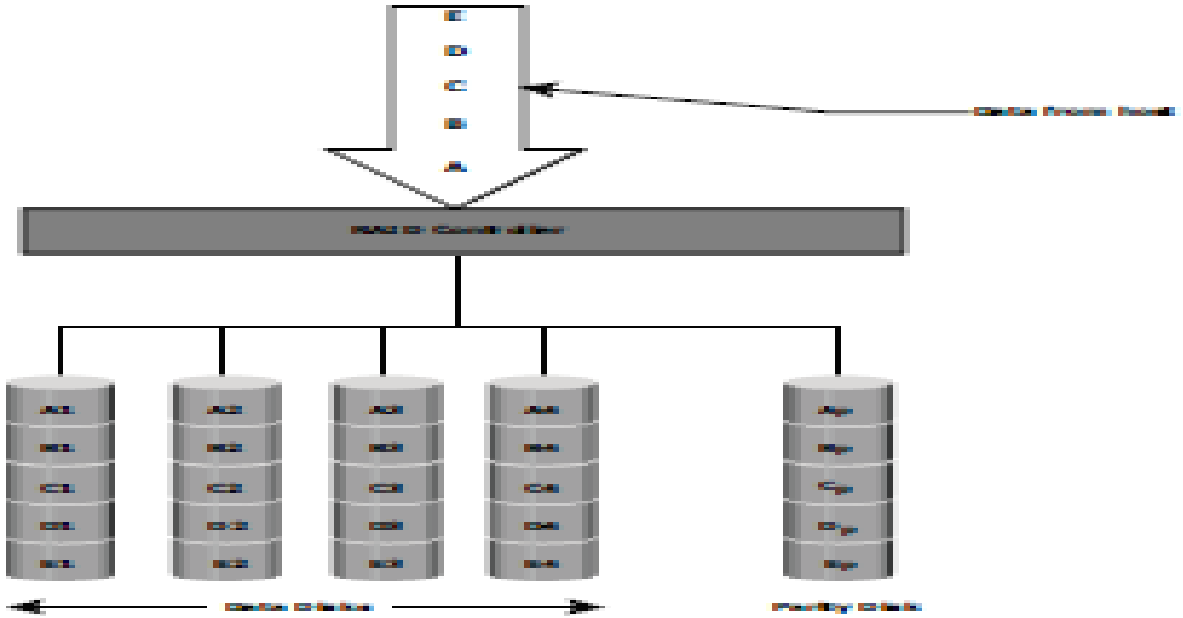


Figure 3–8: RAID 3

RAID 3 provides good bandwidth for the transfer of large volumes of data. RAID 3 is used in applications that involve large sequential data access, such as video streaming.

RAID 4

- Similar to RAID 3, RAID 4 stripes data for high performance and uses parity for improved fault tolerance .
- Data is striped across all disks except the parity disk in the array. Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails.
- Striping is done at the block level.
- Unlike RAID 3, data disks in RAID 4 can be accessed independently so that specific data elements can be read or written on single disk without read or write of an entire stripe.
- RAID 4 provides good read throughput and reasonable write throughput.

RAID 5 Implementation

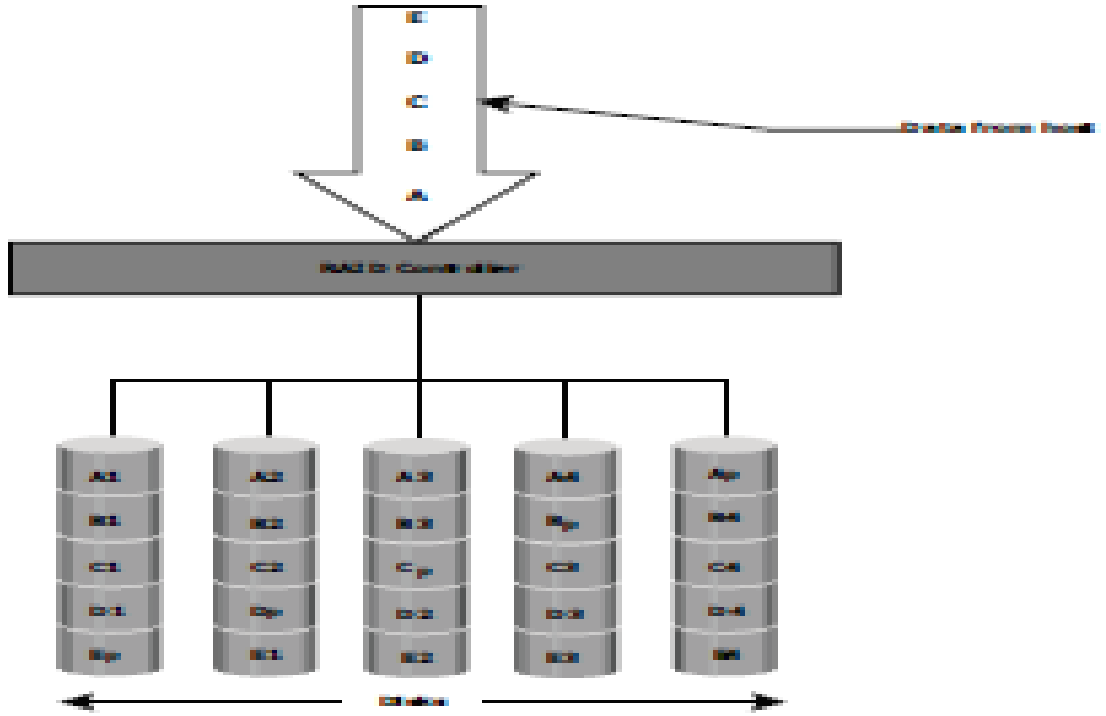


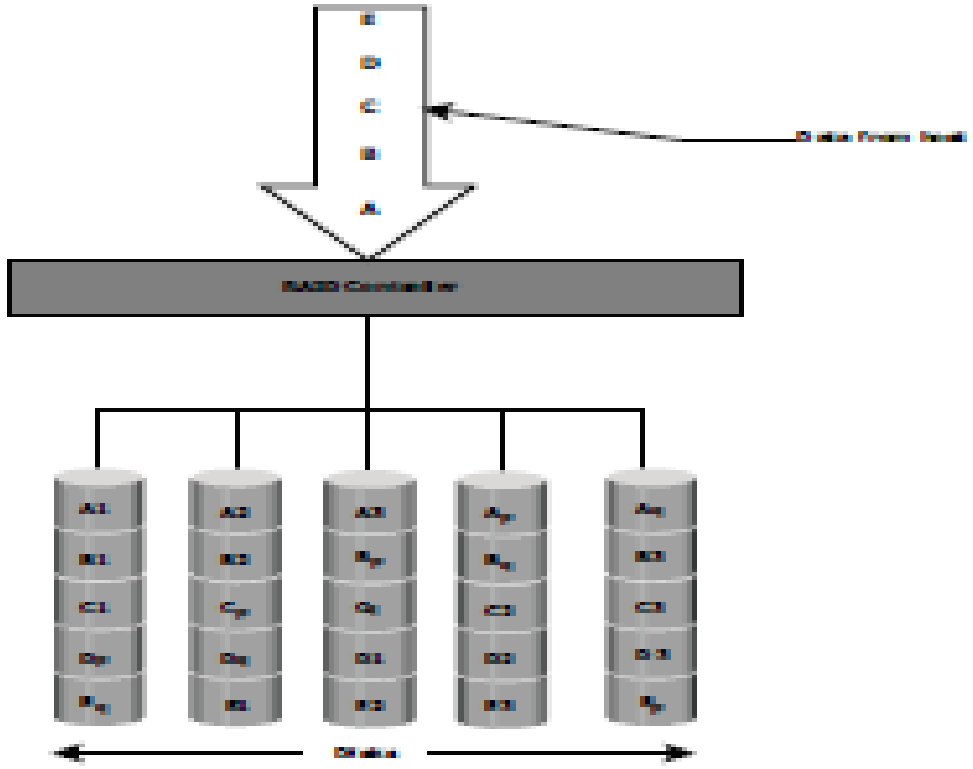
Figure 3-9: RAID 5

RAID 5 is preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations in which database administrators (DBAs) optimize data access.

RAID 5

- RAID 5 is a very versatile RAID implementation.
- It is similar to RAID 4 because it uses striping and the drives (strips) are independently accessible.
- The difference between RAID 4 and RAID 5 is the parity location.
- In RAID 4, parity is written to a dedicated drive, creating a write bottleneck for the parity disk.
- In RAID 5, parity is distributed across all disks.
- The distribution of parity in RAID 5 overcomes the write bottleneck.

RAID 6



igure 3-10: RAID 6

RAID 6

- RAID 6 works the same way as RAID 5 except that RAID 6 includes a second parity element to enable survival in the event of the failure of two disks in a RAID group .
- Therefore, a RAID 6 implementation requires at least four disks.
- RAID 6 distributes the parity across all the disks.
- The write penalty in RAID 6 is more than that in RAID 5; therefore, RAID 5 writes perform better than RAID 6.
- The rebuild operation in RAID 6 may take longer than that in RAID 5 due to the presence of two parity sets.

RAID Comparison

RAID	Min. Disks	Storage Efficiency %	Cost	Read Performance	Write Performance	Write Penalty
0	2	100	low	Very good for both random and sequential read	Very good	no
1	2	50	high	Good better than single disk	Good slower than single disk as every wright must be communicated to all disks	moderate
3	3	(n-1)*100/n where n=no 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.	High
4	3	(n-1)*100/n where n=no of disks	Moderate	Very good for random reads. Good to very good for sequential writes.	Poor to fair for random writes. Fair to good for sequential writes.	high

Table 2.2 The table compares the theoretical basic forms of different RAID levels. In practice, huge differences exist in the quality of the implementation of RAID controllers.

RAID level	Fault-tolerance	Read performance	Write performance	Space requirement
RAID 0	None	Good	Very good	Minimal
RAID 1	High	Poor	Poor	High
RAID 10	Very high	Very good	Good	High
RAID 4	High	Good	Very very poor	Low
RAID 5	High	Good	Very poor	Low
RAID 6	Very high	Good	Very very poor	Low

RAID Impact on Disk Performance

- When choosing a RAID type, it is imperative to consider the impact to disk performance and application IOPS.
- In both mirrored and parity RAID configurations, every write operation
- translates into more I/O overhead for the disks which is referred to as *write penalty*.
- In a RAID 1 implementation, every write operation must be performed on two disks configured as a mirrored pair while in a RAID 5 implementation, a write operation may manifest as four I/O operations.
- When performing small I/Os to a disk configured with RAID 5, the controller has to read, calculate, and write a parity segment for every data write operation.
- Figure illustrates a single write operation on RAID 5 that contains a group of five disks.
- Four of these disks are used for data and one is used for parity.

RAID Impact on Disk Performance

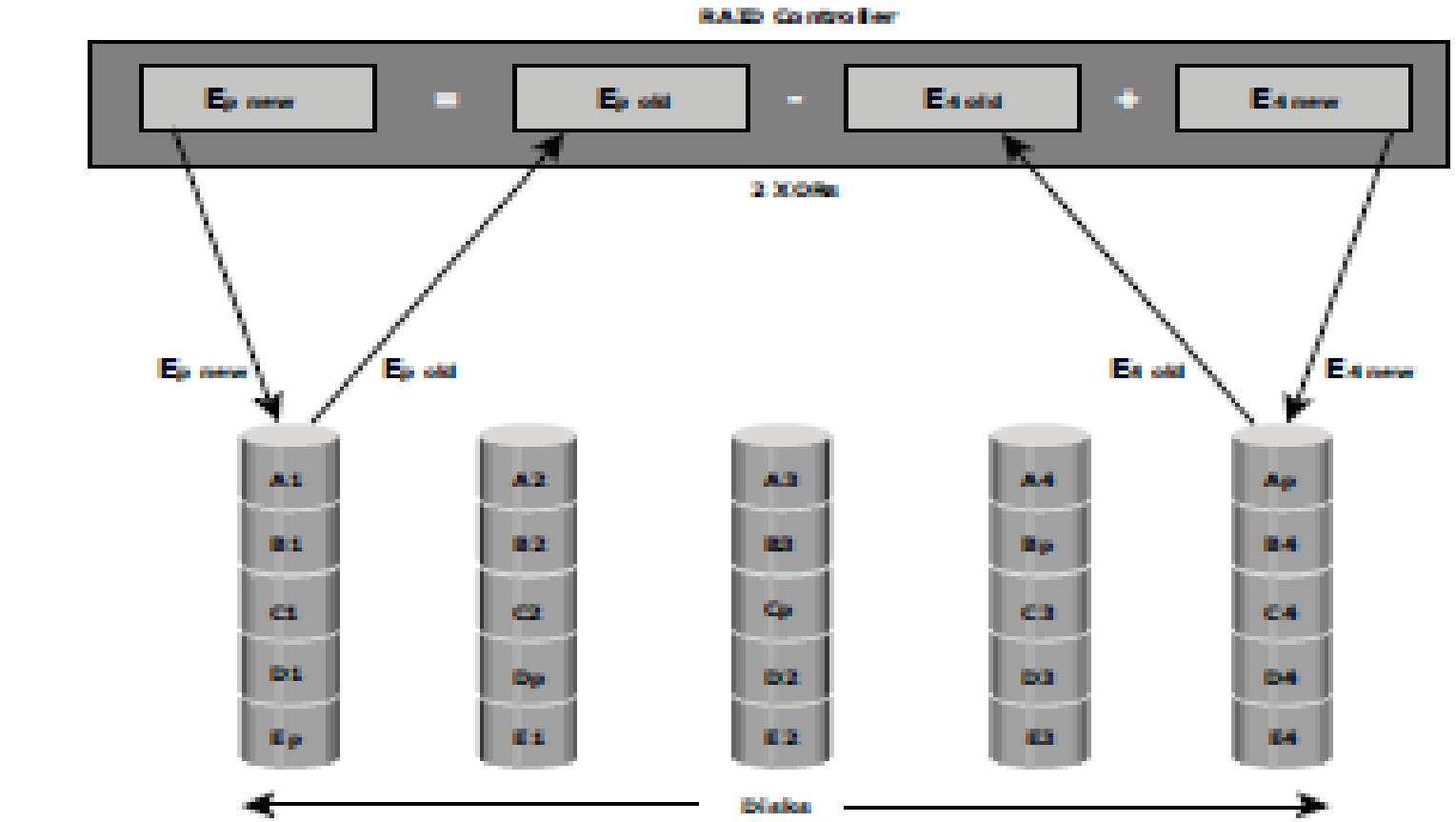


Figure 3-11: Write penalty in RAID 5

- parity.
- The parity (P) at the controller is calculated as follows:
- Ep = E1 + E2 + E3 + E4 (XOR operations)
- Here, D1 to D4 is striped data across the RAID group of five disks.
- Whenever the controller performs a write I/O, parity must be computed by
- reading the old parity (Ep old) and the old data (E4 old) from the disk, which means two read I/Os.
- The new parity (Ep new) is computed as follows:
- Ep new = Ep old – E4 old + E4 new (XOR operations)

- When deciding the number of disks required for an application, it is important to consider the impact of RAID based on IOPS generated by the application.
- The total disk load should be computed by considering the type of RAID configuration and the ratio of read compared to write from the host.
- The following example illustrates the method of computing the disk load in different types of RAID.
- Consider an application that generates 5,200 IOPS, with 60 percent of them being reads.
- The disk load in RAID 5 is calculated as follows:
- RAID 5 disk load = 0.6 × 5,200 + 4 × (0.4 × 5,200)
- [because the write penalty for RAID 5 is 4] = 3,120 + 4 × 2,080 = 3,120 + 8,320 = 11,440 IOPS

- The disk load in RAID 1 is calculated as follows:
- RAID 1 disk load = 0.6 × 5,200 + 2 × (0.4 × 5,200)
- [because every write manifests as two writes to the disks] =
- 3,120 + 2 × 2,080 = 3,120 + 4,160 = 7,280 IOPS
- The computed disk load determines the number of disks required for the application.
- If in this example an HDD with a specification of a maximum 180 IOPS for the application needs to be used, the number of disks required to meet the workload for the RAID configuration would be as follows:
- RAID 5: 11,440 / 180 = 64 disks■n RAID 1: 7,280 / 180 = 42 disks (approximated to the nearest even ■n number)

Hot Spares

- A *hot spare* refers to a spare HDD in a RAID array that temporarily replaces a
- failed HDD of a RAID set. A hot spare takes the identity of the failed HDD in
- the array. One of the following methods of data recovery is performed depending
- on the RAID implementation:
- If parity RAID is used, then the data is rebuilt onto the ■■ hot spare from the
- parity and the data on the surviving HDDs in the RAID set.
- ■■ If mirroring is used, then the data from the surviving mirror is used to
- copy the data

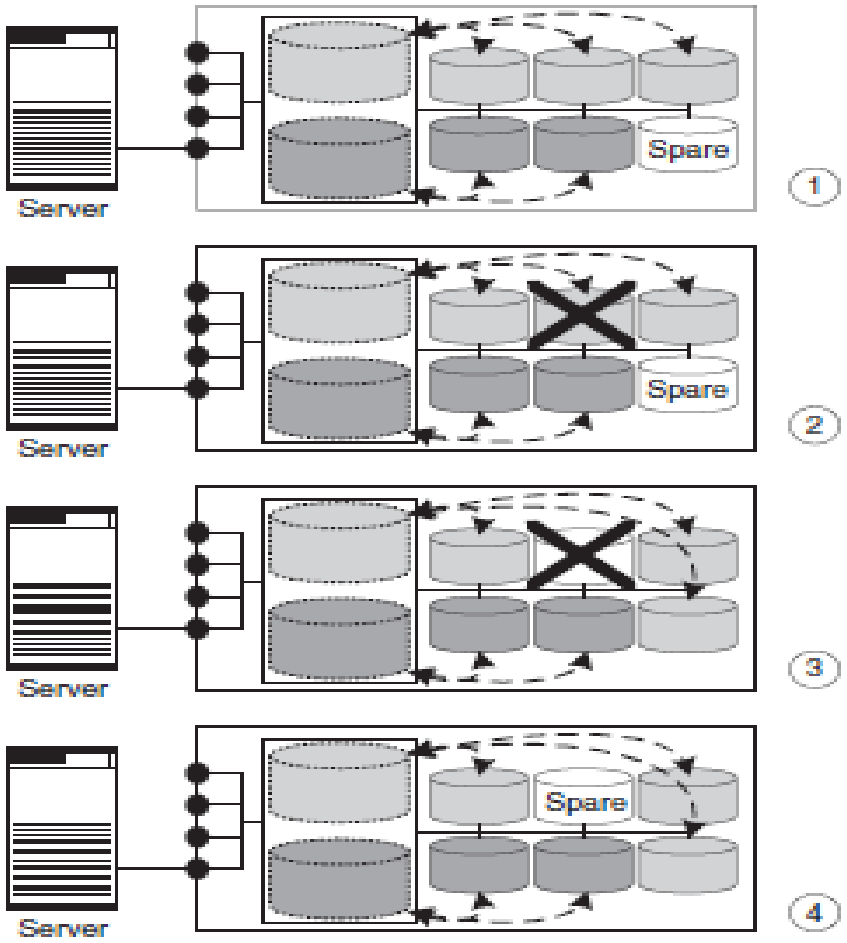


Figure 2.8 Hot spare disk: The disk subsystem provides the server with two virtual disks for which a common hot spare disk is available (1). Due to the redundant data storage the server can continue to process data even though a physical disk has failed, at the expense of a reduction in performance (2). The RAID controller recreates the data from the defective disk on the hot spare disk (3). After the defective disk has been replaced a hot spare disk is once again available (4).

- When the failed HDD is replaced with a new HDD, one of the following
- takes place:
- ■■ The hot spare replaces the new HDD permanently. This means that it is
- no longer a hot spare, and a new hot spare must be configured on the
- array.
- ■■When a new HDD is added to the system, data from the hot spare is
- copied to it. The hot spare returns to its idle state, ready to replace the
- next failed drive.

- A hot spare should be large enough to accommodate data from a failed drive.
- Some systems implement multiple hot spares to improve data availability.
- A hot spare can be configured as *automatic* or *user initiated,* which specifies how it will be used in the event of disk failure.
- In an automatic configuration, when the recoverable error rates for a disk exceed a predetermined threshold, the disk subsystem tries to copy data from the failing disk to the hot spare automatically.
- If this task is completed before the damaged disk fails, then the subsystem switches to the hot spare and marks the failing disk as unusable.
- Otherwise, it uses parity or the mirrored disk to recover the data. In the case of a user-initiated
- configuration, the administrator has control of the rebuild process.
- For example, the rebuild could occur overnight to prevent any degradation of system performance.
- However, the system is vulnerable to another failure if a hot spare is unavailable.