Storage Models

By Neelam Singh

Introduction

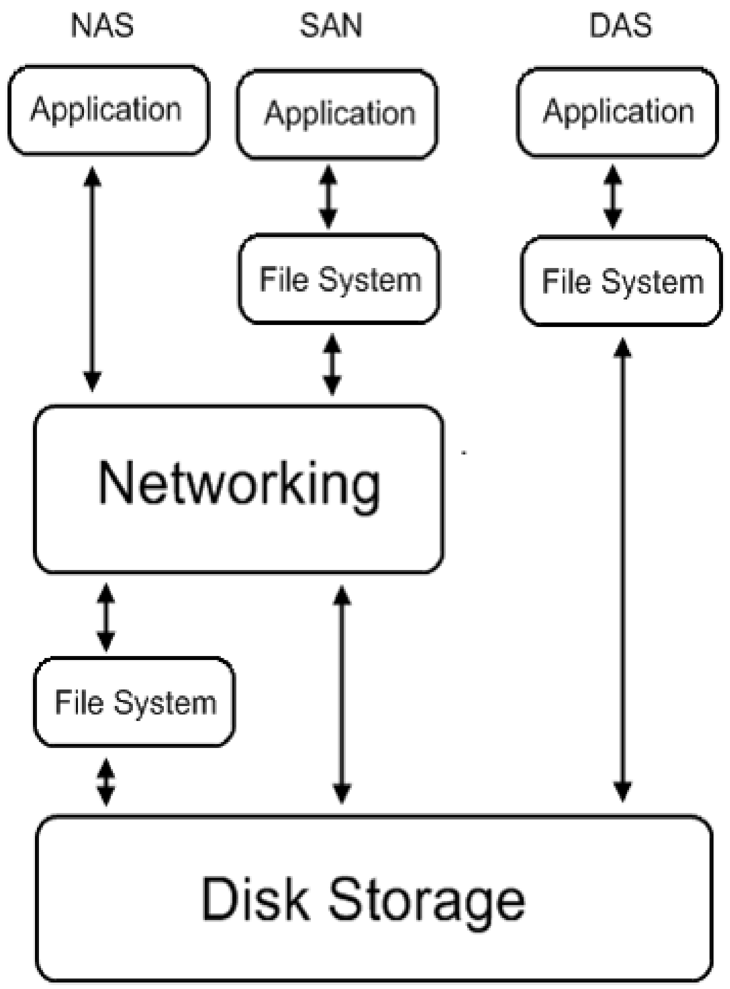
- Cloud storage models are models of cloud computing that stores data on the internet via cloud computing providers. These providers manage and operate data storage as a service.
- Cloud storage is basically online storage of data. Data that is stored can be accessed from multiple connected devices, which constitute a cloud.
- Cloud storage can provide various benefits like greater accessibility and reliability, strong protection of data backup, rapid deployment, and disaster recovery purposes.
- Moving to the cloud also decreases overall storage costs due to cutting costs incurred on the purchase of storage devices and their maintenance.
- Commonly used models are : Block Storage, File Storage and Object Storage.

Storage Models

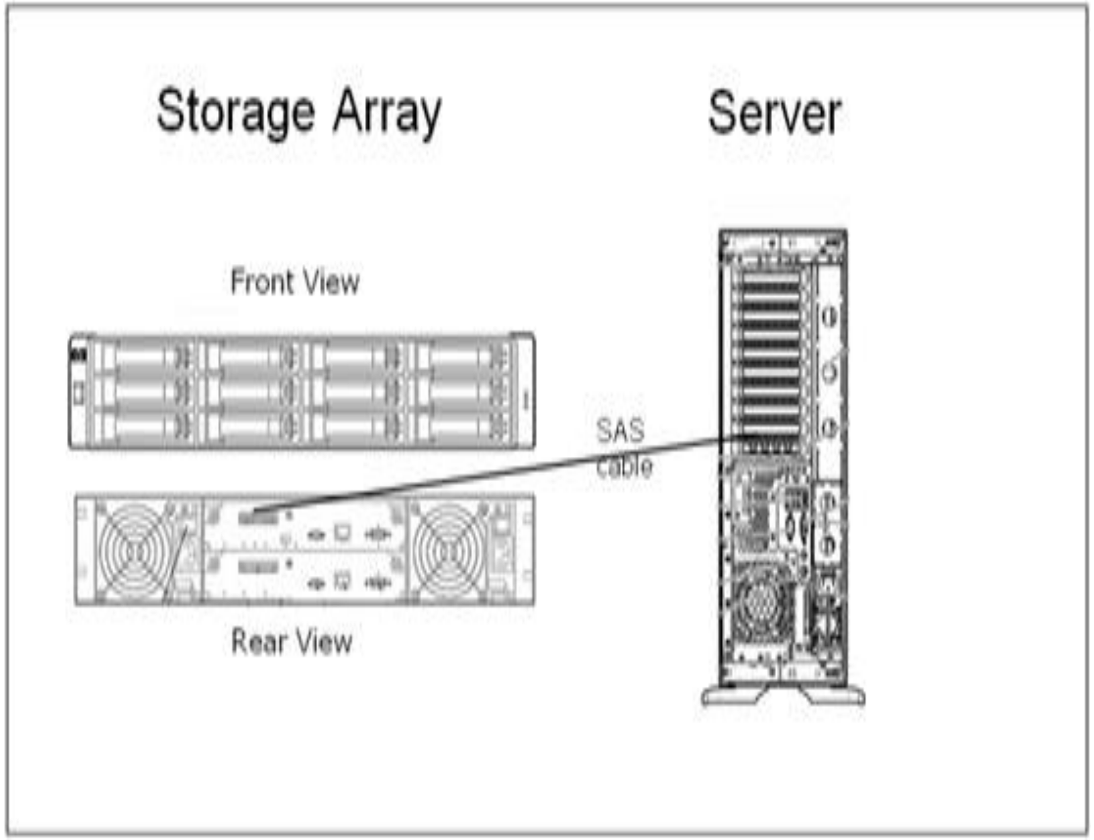
- **Block Storage** focuses on addressing performance centric and transactional application by providing chunks of block data that can be controlled for a particular application through a block interface. In Block storage there is no metadata model for the stored data. Block is used for application models that require low latency between application run time and local storage, example database server.
- **File Storage** uses hierarchical file system to organize file by location and path, attributes can be associated with file system including ownership and access control. Using network attached storage (NAS) Files can be shared among users of the network. File storage is often used for local file and distributed file service and can visible as a network drive
- **Object Storage** uses an object data model which is comprised of the object file and meta data associated with the object. Each object has an identifier and is treated as a whole entity, example: a video, image, or document is treated as an object entity. Object Storage can be used for storing content for user applications to support cloud use cases.

Types

- **Most storage devices share the same physical and logical structure, in order to be able to locate the data Hard Disk drives have sectors (or simply "blocks") to check where data reides, in many cases this reflects the layout of the data written into the physical medium. But accessing your data by addressing the sector number while not very complex, it's an error prone method and you have to keep track yourself of the data you write and the sectors you have written to. So this is where a *file-system* comes to the rescue, a simple file-system will help you by addressing the blocks used and providing you a common interface to retrieve your data, the most common paradigm used is the *folder/file* structure (this is why is it called a *file*-system).**
- **The storage architectures are broadly classified into three types:**
  - **DAS Direct Attached Storage**
  - **NAS Network Attached Storage**
  - **SAN Storage Area Network**



DAS: Direct Attached Storage



Direct attached Storage (DAS) is digital storage that is directly connected to the system (i.e., a PC or a server) through an internal cable.

The DAS system holds multiple hard disk drives in a single enclosure, which is directly connected to a machine through an HBA (Host Bus Adapter). Between these disk drives there is no network device (i.e., a switch, a hub, or a router, or network cable).

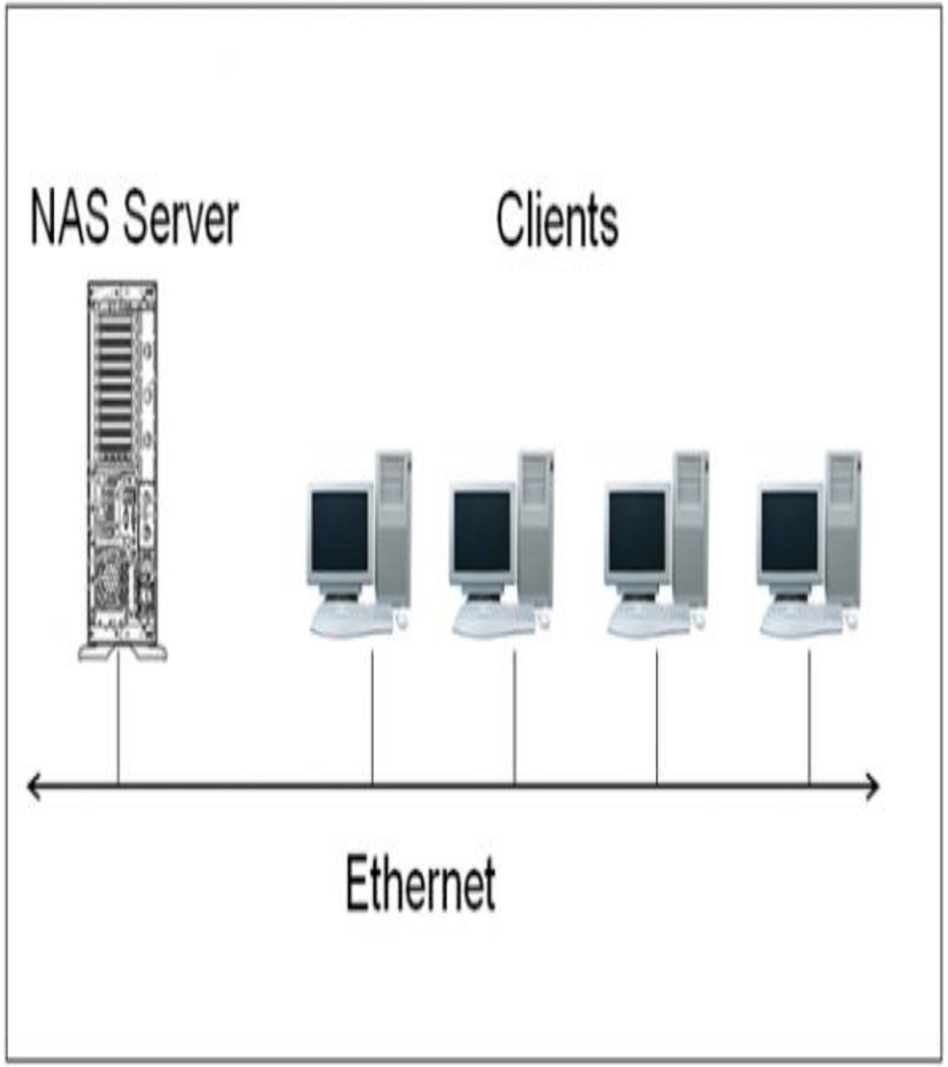
For an individual PC user, the system's hard disk is the standard form of DAS.

Advantages and Disadvantages of DAS

- **Advantages**
- DAS systems are simple to use, and the technology is widely available. It provides great performance compared to SAN or NAS.
- DAS is faster
- DAS is more cost effective
- DAS is manageable

- **Disadvantages**
- The storage capacity cannot be expanded i.e. less scalable
- The disadvantages are that different user groups cannot access the data; it is only directly accessible from the applications running in the individual server or desktop machine. Furthermore, DAS does not incorporate any network hardware nor a related operating environment to provide a facility to share storage resources independently.

NAS: Network Attached Storage



- Network-attached Storage (NAS) is a file-level computer data storage server connected to a computer network.
- It provides storage and access to data from a central location to several authorized network users and other groups of clients.
- These systems are commonly used to support shared applications, including engineering software builds, data logging, email systems, video recording and editing, business analytics, financial records, genomics data sets and much more.
- The file systems are contained in one or more storage drives often arranged into logical, redundant storage containers.
- The NAS sizes are dependent on speed, scale, and budget requirements.
- NAS uses one or more file access protocols that are exposed to an internal network. These are then presented by protocols such as NFS (network file system) or SMB (server message block), or a proprietary high-performance protocol that allow clients to attach to the NAS.

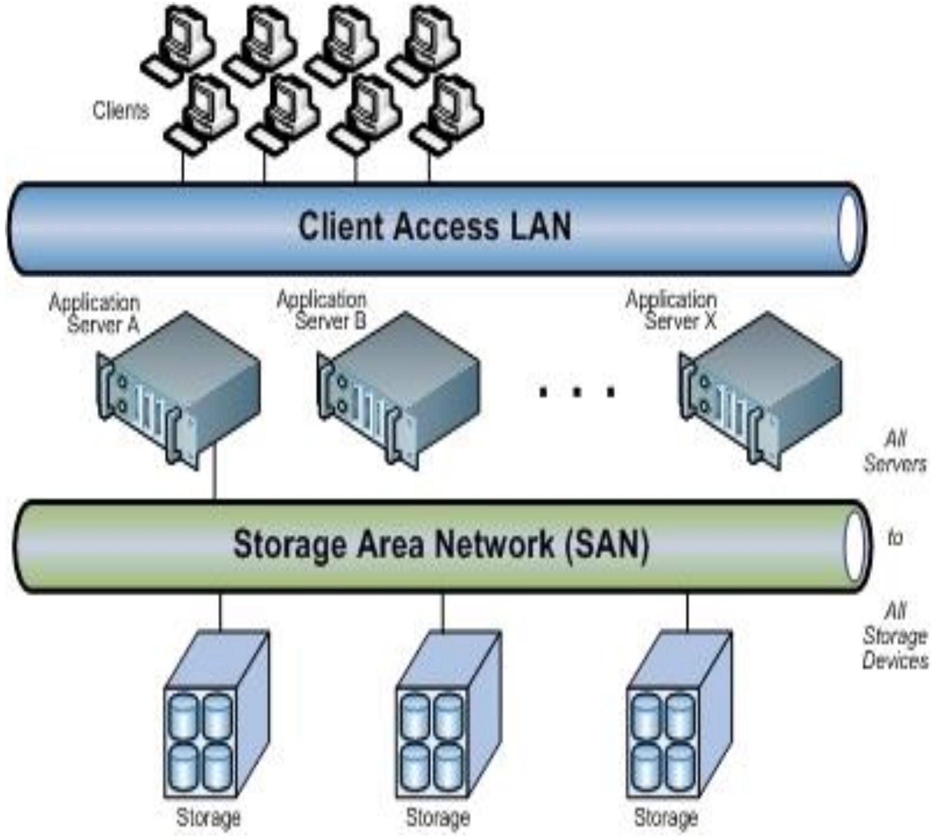
Advantages and Disadvantages of NAS

- **Advantages**
- NAS systems are beneficial for small business owners because they are simple to operate; therefore, an IT professional is often not required. NAS is cost-effective with easy and secure data backup.
- It also significantly reduces wasted space over other storage technologies such as DAS or SAN (storage area network).
- NAS systems are continually accessible, making it easy for employees to collaborate, respond to customers, and support joint development projects.

- Disadvantages
- The weaknesses of NAS are related to scale and performance.
- The NAS is limited to its resources, and if the number of users requiring access increases, the NAS appliance cannot keep up, leading to slow performance and user frustration
- NAS is network dependent

What Is a Storage Area Network (SAN)?

- A Storage Area Network (SAN) is a specialized, high-speed network that provides block-level network access to storage. SANs are typically composed of hosts, switches, storage elements, and storage devices that are interconnected using a variety of technologies, topologies, and protocols. SANs may also span multiple sites.



Components of SAN

- SANs have their own networking devices, such as SAN switches. To access the SAN, so-called SAN servers are used, which in turn connect to SAN host adapters. Within the SAN, a range of data storage devices may be interconnected, such as SAN-capable disk arrays and tape libraries.
- A SAN generally consists of three core components; therefore, SAN architecture is composed of:
  - Hosts: These are the system/end devices that use the SAN services. This can include servers and computers on the network.
  - Fabric: This consists of the interfaces such as fiber channel and host bus adapter that enable connectivity between the hosts and SAN infrastructure.
  - Storage: This is the physical storage drives.
- Typically, SAN architecture defines:
  - Pool of storage used and how it is shared in between different servers or computers connected via the network
  - Type of network or data transmission connection used between the key SAN infrastructure and all connecting nodes
  - Placement of data depending upon the type of SAN architecture or topology
  - Type of SAN topology being used

Types of SAN

- The most common SAN protocols are:
  - **Fibre Channel Protocol (FCP):** The most widely used SAN or block protocol, deployed in 70% to 80% of the total SAN market. FCP uses Fibre Channel transport protocols with embedded SCSI commands.
  - **Internet Small Computer System Interface (iSCSI):** The next largest SAN or block protocol, with approximately 10% to 15% of the market. iSCSI encapsulates SCSI commands inside an Ethernet frame and then uses an IP Ethernet network for transport.
  - **Fibre Channel over Ethernet (FCoE):** FCoE is less than 5% of the SAN market place. It is similar to iSCSI, since it encapsulates an FC frame inside an Ethernet datagram. Then like iSCSI, it uses an IP Ethernet network for transport.
  - **Non-Volatile Memory Express over Fibre Channel (FC-NVMe):** NVMe is an interface protocol for accessing flash storage via a PCI Express (PCIe) bus. Unlike traditional all-flash architectures, which are limited to a single, serial command queue, NVMe supports tens of thousands of parallel queues, each with the ability to support tens of thousands of concurrent commands.

Use of SAN

- SANs are often used to:
  - Improve application availability (e.g., multiple data paths)
  - Enhance application performance (e.g., off-load storage functions, segregate networks, etc.)
  - Increase storage utilization and effectiveness (e.g., consolidate storage resources, provide tiered storage, etc.), and improve data protection and security.
  - SANs also typically play an important role in an organization's Business Continuity Management (BCM) activities.

RAID (Redundant Array of Independent Disks)

- Also known as Redundant Array of Inexpensive Disks.
- It is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both.
- Data is distributed across the drives in one of several ways, referred to as RAID levels, depending on the required level of redundancy and performance.
- Basic idea is to connect multiple disks together to provide
  - large storage capacity
  - faster access to reading data
  - redundant data

RAID flavors

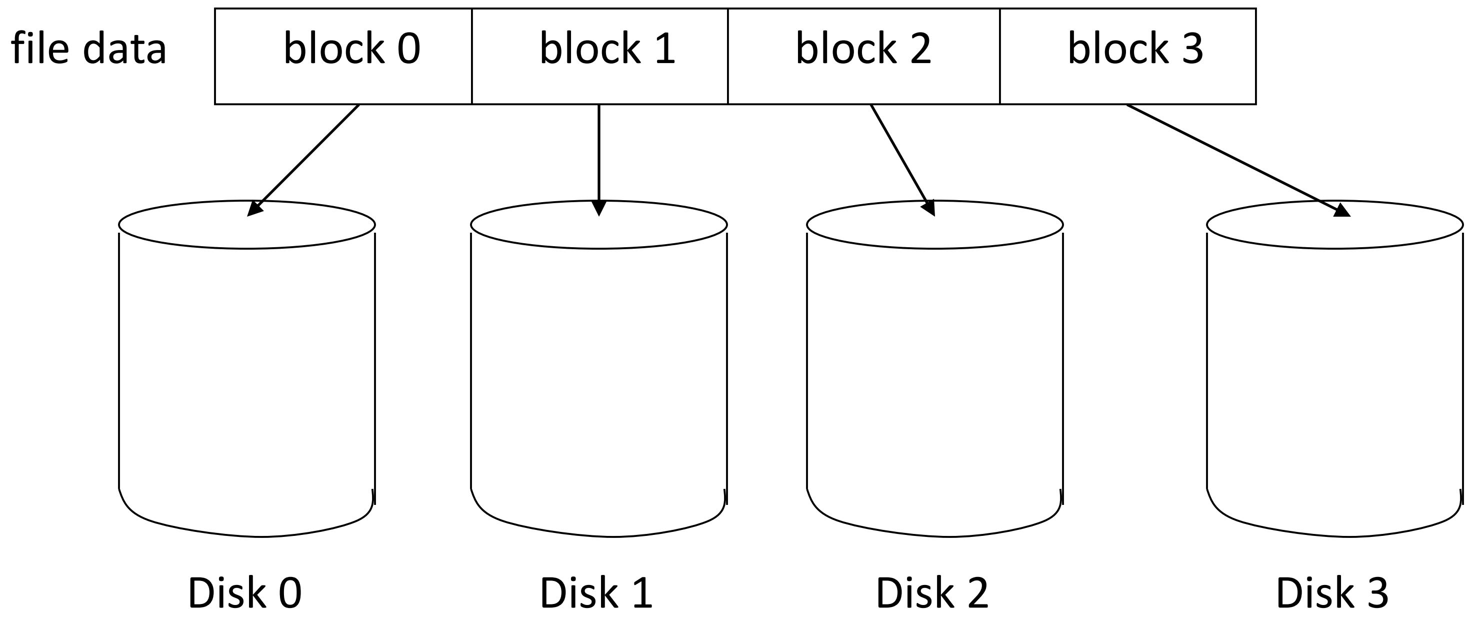
- Commonly used ones:

1. RAID 0
2. RAID 1
3. RAID 5
4. RAID 10

- Other types used…but rarely: RAID 2,3,4,6,50

Striping

- Take file data and map it to different disks
- Allows for reading data in parallel



Parity

- Way to do error checking and correction
- Add up all the bits that are 1
  - if even number, set parity bit to 0
  - if odd number, set parity bit to 1
- To actually implement this, do an exclusive OR of all the bits being considered
- Consider the following 2 bytes

byte	parity
10110011	1
01101010	0

- If a single bit is bad, it is possible to correct it

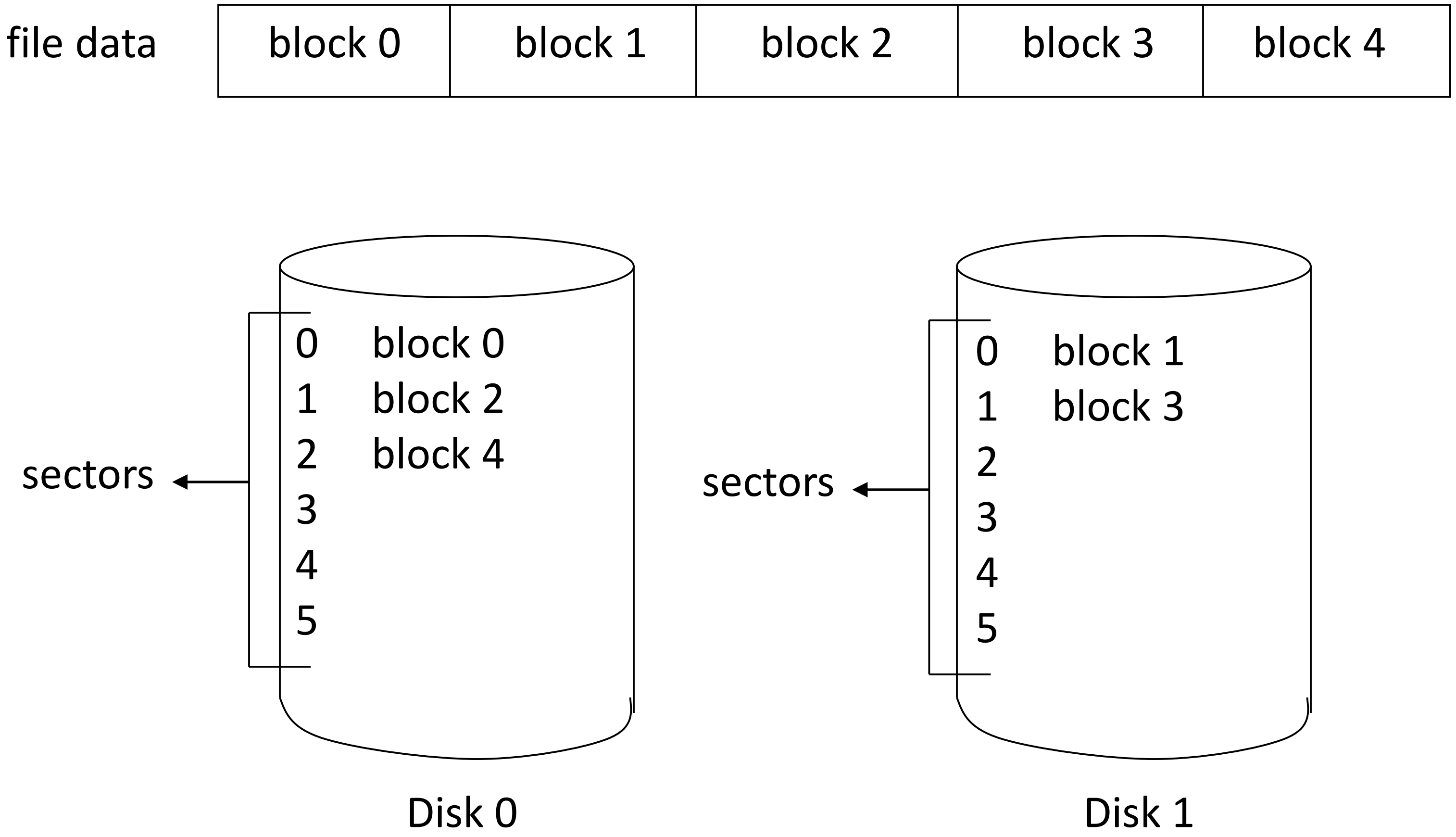
Mirroring

- Keep to copies of data on two separate disks
- Gives good error recovery
  - if some data is lost, get it from the other source
- Expensive
  - requires twice as many disks
- Write performance can be slow
  - have to write data to two different spots
- Read performance is enhanced
  - can read data from file in parallel

RAID Level-0

- Often called striping
- Break a file into blocks of data
- Stripe the blocks across disks in the system
- Simple to implement
  - disk = file block % number of disks
  - sector = file block / number of disks
- provides no redundancy or error detection
  - important to consider because lots of disks means low Mean Time To Failure (MTTF)

RAID Level-0

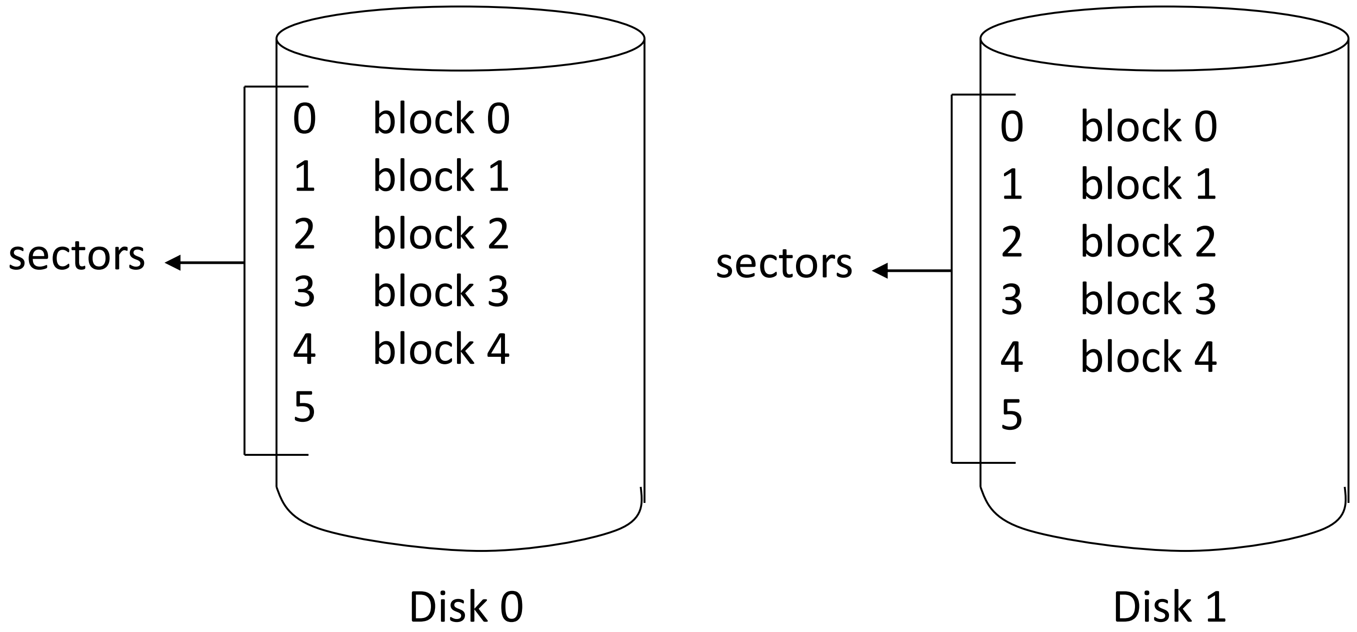

file data
block 0
block 1
block 2
block 3
block 4
sectors
0
1
2
3
4
5
block 0
block 2
block 4
Disk 0
sectors
0
1
2
3
4
5
block 1
block 3
Disk 1

RAID Level-1

- A complete file is stored on a single disk
- A second disk contains an exact copy of the file
- Provides complete redundancy of data
- Read performance can be improved
  - file data can be read in parallel
- Write performance suffers
  - must write the data out twice
- Most expensive RAID implementation
  - requires twice as much storage space

RAID Level-1

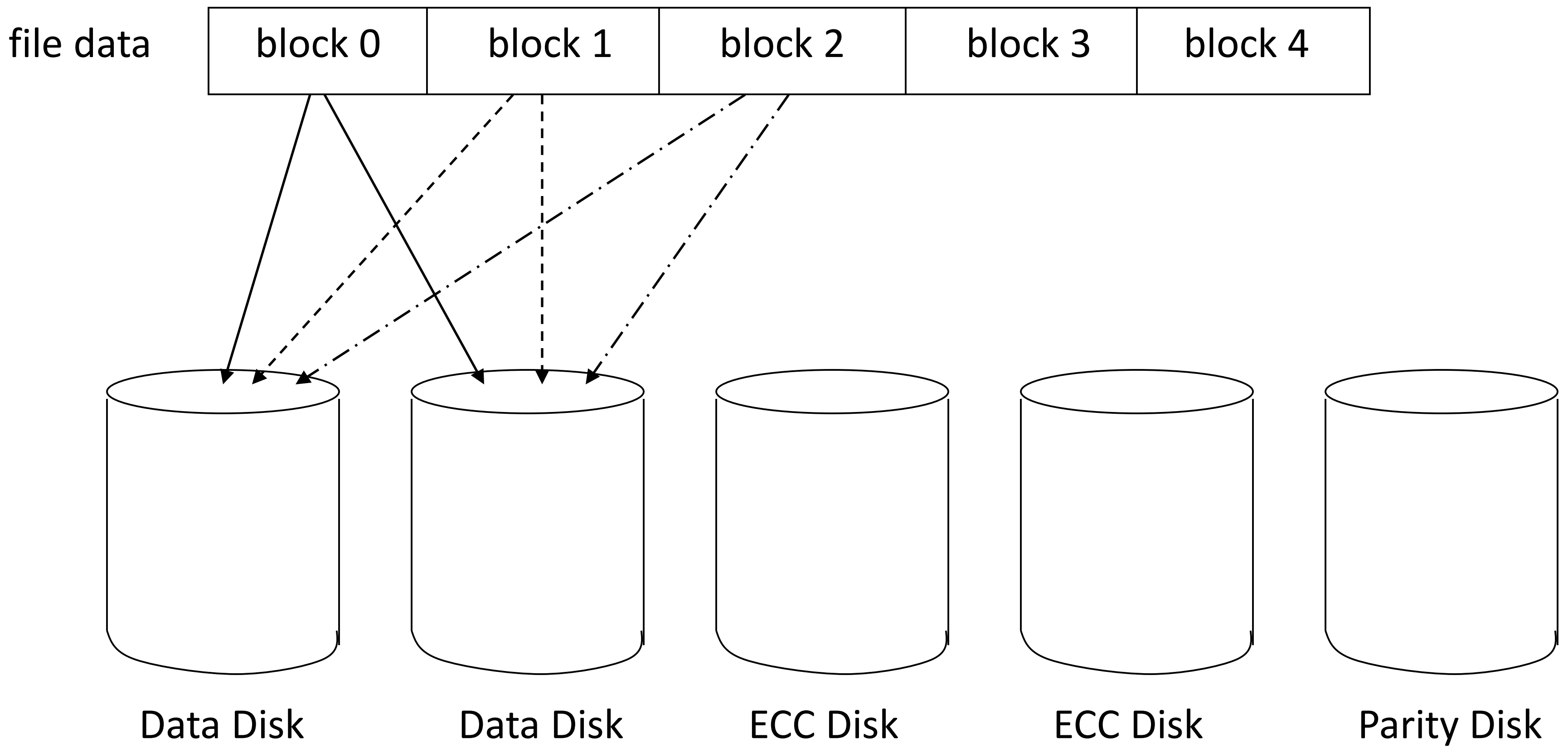
file data	block 0	block 1	block 2	block 3	block 4



RAID Level-2

- Stripes data across disks similar to Level-0
  - difference is data is bit interleaved instead of block interleaved
- Uses ECC to monitor correctness of information on disk
- Multiple disks record the ECC information to determine which disk is in fault
- A parity disk is then used to reconstruct corrupted or lost data

RAID Level-2


file data
block 0
block 1
block 2
block 3
block 4
Data Disk
Data Disk
ECC Disk
ECC Disk
Parity Disk

RAID Level-2

- Reconstructing data
  - assume data striped across eight disks
  - correct data: 10011010
  - parity: 0
  - data read: 10011110
  - if we can determine that disk 2 is in error
  - just use read data and parity to know which bit to flip

RAID Level-2

- Requires fewer disks than Level-1 to provide redundancy
- Still needs quite a few more disks
  - for 10 data disks need 4 check disks plus parity disk
- Big problem is performance
  - must read data plus ECC code from other disks
  - for a write, have to modify data, ECC, and parity disks
- Another big problem is only one read at a time
  - while a read of a single block can be done in parallel
  - multiple blocks from multiple files can't be read because of the bit-interleaved placement of data

RAID Level-3

- One big problem with Level-2 are the disks needed to detect which disk had an error
- Modern disks can already determine if there is an error
  - using ECC codes with each sector
- So just need to include a parity disk
  - if a sector is bad, the disk itself tells us, and use the parity disk to correct it

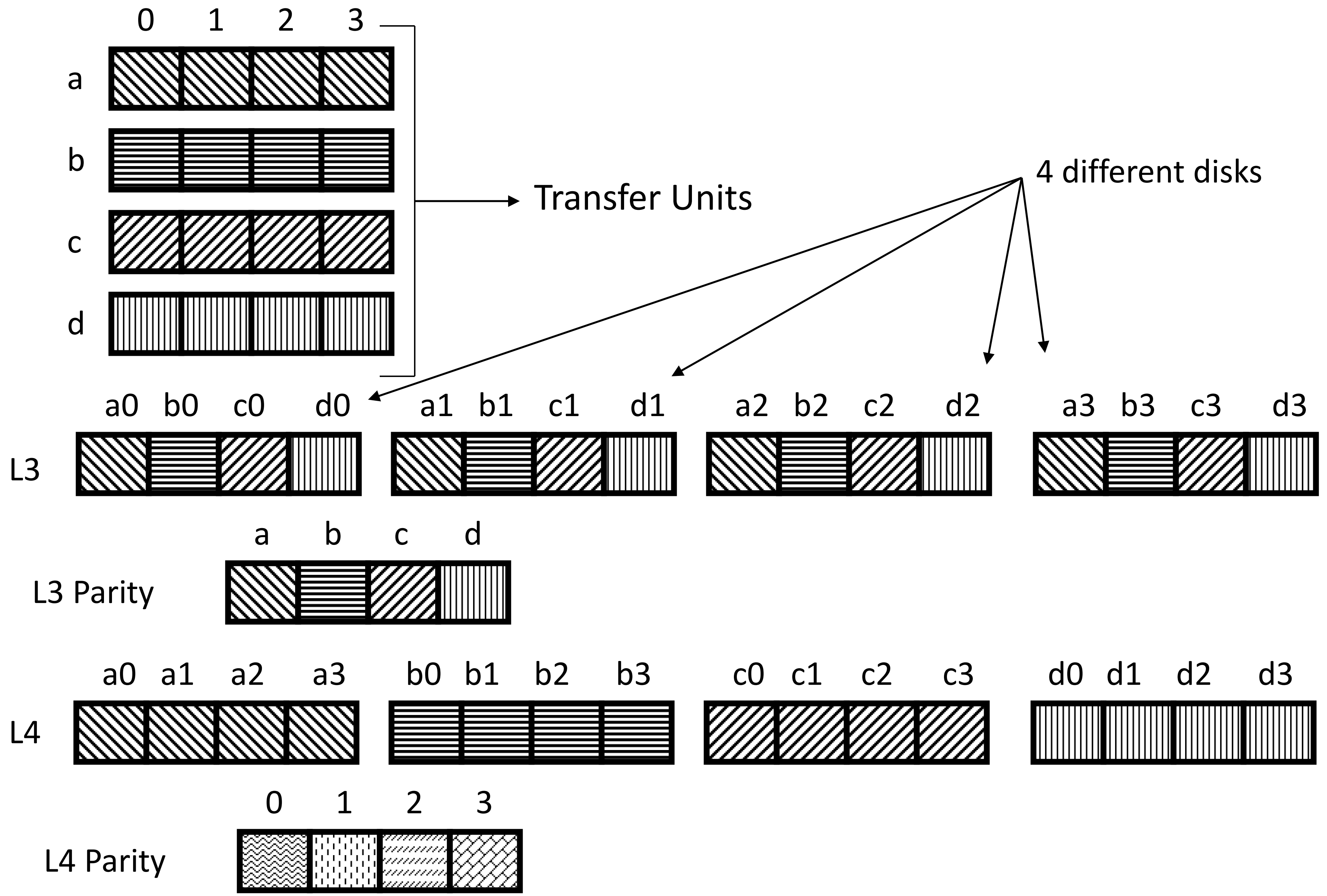
RAID Level-4

- Big problem with Level-2 and Level-3 is the bit interleavening
  - to access a single file block of data, must access all the disks
  - allows good parallelism for a single access but doesn’t allow multiple I/O’s
- Level-4 interleaves file blocks
  - allows multiple small I/O’s to be done at once

RAID Level-4

- Still use a single disk for parity
- Now the parity is calculated over data from multiple blocks
  - Level-2,3 calculate it over a single block
- If an error detected, need to read other blocks on other disks to reconstruct data

Level-4 vs. Level-2,3



RAID Level-4

- Reads are simple to understand
  - want to read block A, read it from disk 0
  - if there is an error, read in blocks B,C, D, and parity block and calculate correct data
- What about writes?
  - it looks like a write still requires access to 4 data disks to recalculate the parity data
  - not true, can use the following formula
    - new parity = (old data xor new data) xor old parity
  - a write requires 2 reads and 2 writes

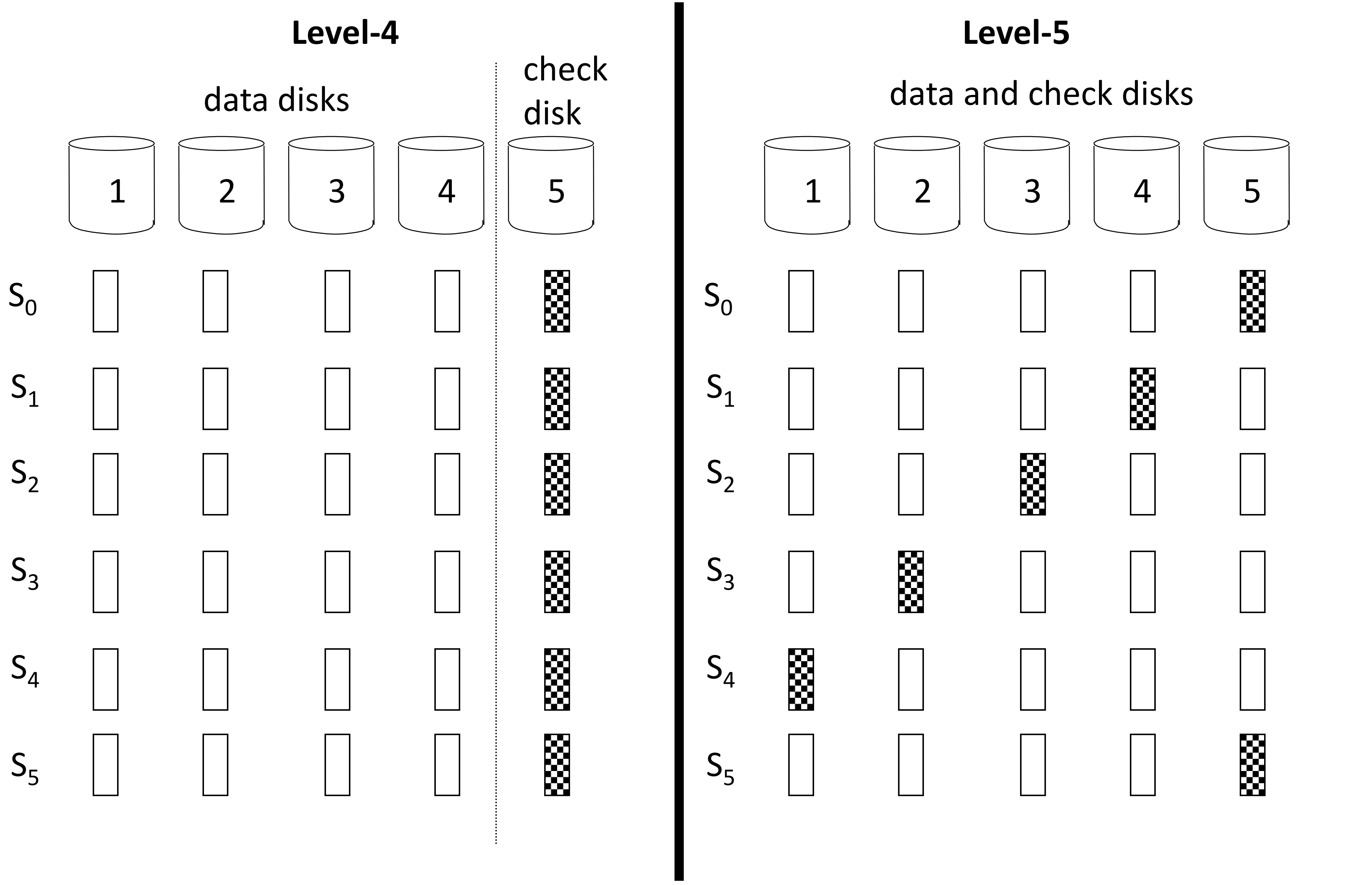
RAID Level-4

- Doing multiple small reads is now faster than before
- However, writes are still very slow
  - this is because of calculating and writing the parity blocks
- Also, only one write is allowed at a time
  - all writes must access the check disk so other writes have to wait

RAID Level-5

- Level-5 stripes file data and check data over all the disks
  - no longer a single check disk
  - no more write bottleneck
- Drastically improves the performance of multiple writes
  - they can now be done in parallel
- Slightly improves reads
  - one more disk to use for reading

RAID Level-5


Level-4
data disks
check
disk
1
2
3
4
5
S_0
S_1
S_2
S_3
S_4
S_5
Level-5
data and check disks
1
2
3
4
5
S_0
S_1
S_2
S_3
S_4
S_5

RAID Level-5

- Notice that for Level-4 a write to sector 0 on disk 2 and sector 1 on disk 3 both require a write to disk five for check information
- In Level-5, a write to sector 0 on disk 2 and sector 1 on disk 3 require writes to different disks for check information (disks 5 and 4, respectively)
- Best of all worlds
  - read and write performance close to that of RAID Level-1
  - requires as much disk space as Levels-3,4

RAID Level-10

- Combine Level-0 and Level-1
- Stripe a files data across multiple disks
  - gives great read/write performance
- Mirror each strip onto a second disk
  - gives the best redundancy
- The most high performance system
- The most expensive system