UNIT-1

Intro :

- Definition : (We have 2-3 definitions but this was the one that I found the most relevant and above all, meaningful)
    - IT infrastructure consists of the equipment, systems, software, and services combined and used in common across an organization, regardless of the mission/program/project. IT Infrastructure also serves as the foundation upon which mission/program/project specific systems and capabilities are built.
- But here's the catch : the definition also depends upon the one who is actually using the infrastructure.

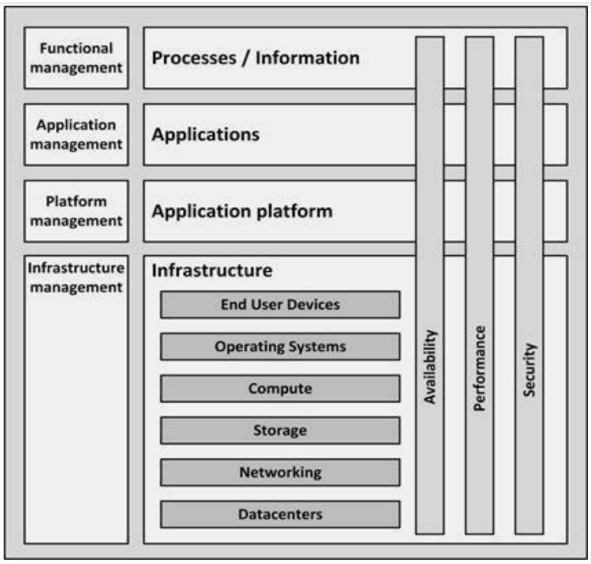


Figure 2: The infrastructure model

1. **PROCESS INFORMATION BUILDING BLOCK**:
    - Organisations implement business processes for their serving their tasks which are mostly organisation specific .
    - Example : Business processes for an insurance firm can be
        - creating a policy
        - ticket for claims
        - payment invoices
        - . . . and so on
    - Functional management is the part of the system management components that is responsible for configuring the system to serve business needs
2. **APPLICATION BUILDIND BLOCK**:

- This includes 3 types of blocks :
  - **Client Applications** :
    - Generally run on end-user devices like PC,laptops etc.
    - Example : Web Browser , Notepad etc.
  - **Office Applications** :
    - They provide a standard server based application for organization level usage.
    - Example : mail servers , portals, collaboration tools etc.
  - **Business Specific Applications** :
    - They are one level up of Office application with very high specificity .
    - Example :
      - **CRM** : Customer Relationship Management
      - **ERP** : Enterprise Resource Planning
      - **SCADA** : Supervisory Control And Data Acquisition
      - Some applications that are designed for specific business applications like **IMS** (institute management system).
- Applications management is the part of the system management components that is responsible for configuring the system for other technical operations.

3. **APPLICAITON PLATFORM BUILDING BLOCK**:
   - Many Application need some additional services (known as application platforms) that enable them to work.
   - It's more like a framework you need to actually build a working application but the framework is a bit more of a fundamental application. Like you'll need Sk-Learn or pytorch to build a machine learning or deep learning application.
   - There are mainly 4 types of Application platform building tools :
     - **Front end Servers** :
       - Act as web server to host applications that provide end user interaction with the application via web browser (basically servers that host your website).
       - Example : - Apache HTTP Server - Microsoft Internet Information Services – IIS
     - **Application Servers**:
       - The containers running the actual application .
       - Example :
         - IBM WebSphere
         - Apache Tomcat
         - Red Hat JBoss
         - Windows .Net
     - **Connectivity**:
       - Consists of FTP servers, Extraction, Transformation andLoad (ETL) servers, and Enterprise Service Buses (ESBs).

        - Basically it's the services and infrastructure that allow different components of an application to communicate and exchange data with each other i.e both within the application and with external systems.
        - Example :
            - Microsoft BizTalk
            - the TIBCO Service Bus
            - IBM MQ
            - SAP NetWeaver PI
    - **Databases**:
        - Store the data.
        - Example :
            - Oracle RDBMS
            - IBM DB2
            - Microsoft SQL Server
            - PostgreSQL
            - MySQL
- This part of the infrastructure is mainly managed by system managers who specialize in a specific technology.

4. **INFRASTRUCTURE BUILDING BLOCKS**:
    - This mainly consists of the hardware and protocol aspect of the infrastructure.
        - **END USER DEVICE**: PCs, laptops, mobile devices, printers etc.
        - **OPERATING SYSTEM**
        - **COMPUTE** : Physical and virtual devices in datacentre. (basically servers)
        - **STORAGE**
        - **NETWORKING** : mainly to connect all components.
        - **DATACENTERS** : locations that host all the hardware.

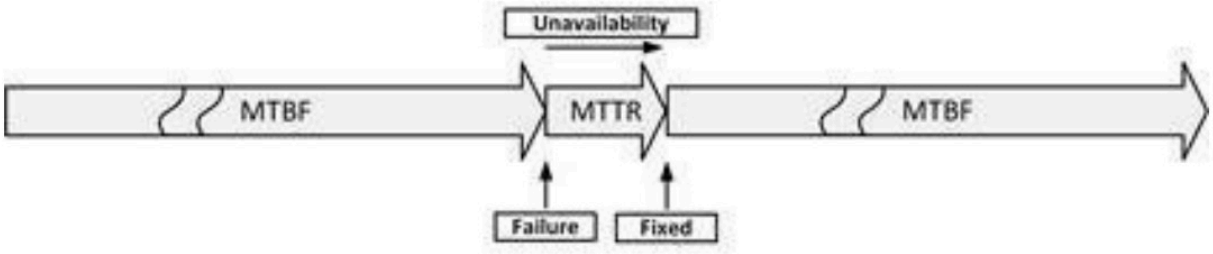
NON FUNCTIONAL ATTIRBUTES

- They describe the qualitative behaviour of a system.
- They may not be related directly to the primary functionalities of a system, but they do have a huge impact on the performance and overall reliability of the whole infrastructure.
- They are :
    - **Availability**
    - **Scalability**
    - **Reliability**
    - **Stability**
    - **Testability**

    - **Recoverability**
- Mainly we'll focus on security, performance and availability.
- Many of the non functional attributes of an application are delivered by relying on the infrastructure. This can be realised by the below points :
    - An application using an infrastructure that has several single point failures will not work out well no matter how well the application is built.
    - If the infrastructure is not highly available then no matter the design of the application, it simply can't be scaled to a handle heavy usage.

AVAILABILITY

- As we know, everyone wants their services and systems available all the time(which can't be guaranteed ever) as the downtime or unavailability is noticed pretty quickly. (Classic Example: How do we feel when any messaging service or any social media gets down.)
- The availability of a system is usually expressed as a **percentage of uptime** in a given time period (usually one year or one month).
- What is uptime ? The total time for which the services were available for active use.
- Typical requirements used in service level agreements today are 99.8% **or** 99.9% **availability per month** for a full IT system.
- To meet this requirement, the availability of the underlying infrastructure must be much higher, typically in the range of 99.99% or higher.
- 99.999% uptime is also known as **carrier grade availability**.
- Factors used to calculate availability are :
    - **Mean Time Between Failures(MTBF)** : average time that passes between failures.
    - **Mean Time To Repair (MTTR)** : time it takes to recover from a failure.



- MTBF expressed in hours (how many hours will the component or service work without failure).

- Formula

$$\textbf{MTBF} = \frac{\textbf{total time of operation}}{\textbf{number of failures}}$$

- Method of calculation :
  - testing time $= 3$ months
  - number of disks $= 1000$
  - Number of disks that failed in that duration = 5
  - $\implies$ for one year, $5 \times 4 = 20$ disks would have failed (extrapolating the 3 month testing results to a year) or alternatively we can say that the system failed 20 times.
  - Total run-time of disks$= 1,000 \times 365 \times 24 = 87,60,000$
  - MTBF $= \frac{\textbf{total time of operation}}{\textbf{number of failures}} = \frac{87,60,000}{20} = 4,38,000$
  - Since here we're **extrapolating** the MTBF value for 1 year (by calculating the values for 3 months and then extending those values for 1 year), in reality it only shows the uptime for initial months extending up-to 1 year.
  - UNOFFICIAL NOTE :
    - **Why to extrapolate ?** In reality no one got time to test out this thing for a year. That's why keep things practical and test out for 1 month or 3 months and then say that yes , this shit can go on this hours of uptime before it ducks up.

- **MTTR** is generally kept very low by having a service contract with a supplier and keeping some spare parts on-site for speedy repair.
- **NOTE:** :
  - $\textbf{availability} \propto \frac{1}{\textbf{MTTR}}$
  - $\textbf{availability} \propto \textbf{MTBF}$

$$\textbf{Availability} = \frac{\textbf{MTBF}}{\textbf{MTTR}+\textbf{MTBF}} \times 100\%$$

- **Serial Availability** : when a **failure in one part causes the failure of the entire system**,the availability of such system is called Serial Availability.
- For such systems availability is calculated by the product of availability of all the components.
- Formula :

$$\textbf{Serial Availability} = \prod^{x \in components} Availability_x$$

Component	MTBF (h)	MTTR (h)	Availability	in %
Power supply	100,000	8	0.9999200	99.99200
Fan	100,000	8	0.9999200	99.99200
System board	300,000	8	0.9999733	99.99733
Memory	1,000,000	8	0,9999920	99.99920
CPU	500,000	8	0.9999840	99.99840
Network Interface Controller (NIC)	250,000	8	0.9999680	99.99680

- Say all the above said components are in series, therefore you to calculate the availability of the system, you need to multiply all the values in the availability column i.e

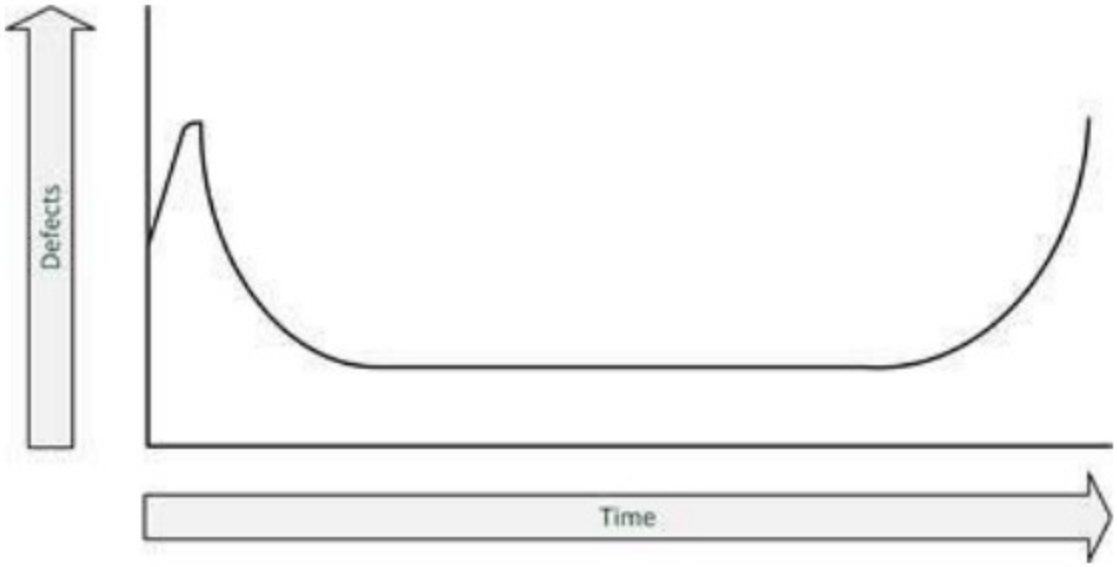
$$\text{availability} = 0.9999200 \times 0.9999200 \times 0.9999733 \times 0.9999920 \times 0.9999840 \times 0.9999680 = 0.9997733 = 99.97733\%$$

- $\implies$ The more components a system includes (and each component is critical for the total system) the lower the total availability becomes.
- Hence we try to avoid many systems in series and rather keep them in parallel.
- For parallel systems :
- Say :
  - There are N components out of which at max M can fail.
  - $\implies$ $A_{M,N} = \sum_{i=0}^{M} \frac{N!}{i! \times (N-i)!} \times A^{N-i} \times (1-A)^i$
  - Special case given in the book : $M = 1, N = 2$
  - $A = 1 - (1 - A_x)^2$
  - A_x is the availability of the component x (basically 2 same type of components in parallel) (SOURCE)

SOURCES OF UNAVAILABILITY

- Human Errors
- Software Bugs
- Physical Defects
- Environmental Effect (Earthquakes, Fire accidents etc.)

BATHTHUB CURVE



- In most cases the availability of a component follows a so-called bathtub curve.
- This says that a **component failure is most likely when the component is new**. -In the first month of use the chance of a components failure is relatively high. Sometimes a component doesn't even work at all when unpacked for the first time. This is called a **DOA component – Dead On Arrival**.
- When a component still works after the first month, it is likely that it will continue working without failure until the end of its technical life cycle.

AVAILABILITY PATTERNS

- **SPOF (Single Point Of Failure)** is a component in the infrastructure that if fails can lead to downtime for the entire system (complete disaster).
- One solution can be **RAID : Redundant Array of Independent Disks** can be used to handle SPOF in storage systems as failure of one disk does not affect the availability of the storage system.
- Other methods include Server Cluster, Double Networks, Dual Datacentres etc but they themselves are also not very SPOF proof. They may end-up sharing some common infrastructure behind the scenes and may lead to an unforeseen SPOF.
- More reliable ways of making infrastructure SPOF proof are **redundancy, failover, and fallback**.

REDUNDANCY

Mainly involved **duplication of critical components in a single system**, to avoid a SPOF. Redundancy is usually implemented in power supplies (a single component having two power

supplies; if one fails, the other takes over and life goes on...), network interfaces, and SAN HBAs (Host Bus Adapters) for connecting storage.

FAILOVER

It's an **automatic/semi-automatic switch-over to a standby system (component)**, either in the same or in another datacentre, upon the failure or abnormal termination of the previously active system (component). Examples Windows Server failover clustering, VMware High Availability and (on the database level) Oracle Real Application Cluster (RAC).

FALLBACK

It's a **manual switchover to an identical standby computer system in a different location**, typically used for disaster recovery. There are three basic forms of fallback solutions:

- **Hot site** : A fully configured fallback datacentre, fully equipped with power and cooling. The applications are installed on the servers, and data is kept up-to-date to fully mirror the production system. Sites of this type requires constant maintenance of the hardware, software, data, and applications to be sure the site accurately mirrors the state of the production site at all times.
- **Warm site** :It's a computer facility **readily available with power, cooling, and computers, but the applications may not be installed or configured**. But external communication links and other data elements, that commonly take a long time to order and install, will be present. It **needs less attention when not in use** and is much **cheaper**.
- **Cold site** : It's a site **ready for equipment to be brought in during an emergency, but no computer hardware is available at the site**. The cold site is **a room with power and cooling facilities**, but computers must be brought on-site if needed, and **communications links may not be ready**. Applications will need to be installed and current data fully restored from backups.

NOTE : SKIPPING BUSINESS CONITNUITY(Page No. 55 of PDF SECTION 4.4.4) (REASON : I didn't find it of much importance)

PERFORMANCE

- Generally disregarded when the system is performing fast and efficiently but highly criticized when the performance is slow and inefficient.

PERCEIVED PERFORMANCE

- Refers to how quickly the **system appears** to perform a given task.
- The user may be aware of the fact that the task assigned is complex and time consuming but just to make the user satisfied and calm, we tend to provide progress bars or splash screens so that

the user may be able to see the progress of the task they assigned.

PERFORMANCE DURING INFRASTRUCTURE DESIGN

- We intend to support a basic required performance of a system even under increased load.
- Not only under normal state where systems work as expected but during special states, it should also have a minimum performance benchmark. These special states are mostly
  - part failure
  - system maintenance
  - backup
  - batch processing
- To determine the performance of a system, we use the below methods :
  - **BENCHMARKING**
    - A benchmark uses a specific test program to assess the relative performance of an infrastructure component.
    - Benchmarking is used to assess the performance characteristics of computer hardware.
    - Example :
      - The floating-point operation performance (**Floating Point Operations Per Second – FLOPS**)
      - Number of instructions per second (**Million Instructions Per Second – MIPS**) of a CPU.
    - They measure the raw performance, not accounting the typical usage components under consideration.
  - **USER VENDOR BENCHMARKING**
    - Basically going to the big old players who are in the industry of IT Infrastructure management like oracle, IBM, AWS etc.
  - **PROTOTYPING**
    - Build a prototype to estimate the performance at an early stage.
    - This acts as a proof of concept and should be used to test the most tough/complex part of the project/infrastructure.
    - A proof of concept shows this at a time not too much money is spent yet and shows to both the project team and the customer that the project's highest risk has been taken care of .
  - **USER PROFILING**
    - It's used to predict the load a new software system will pose on the infrastructure, and to be able to create performance test scripts that put representative load in the infrastructure before the software is actually built. (Basically predicting the usage)

- This can be done by defining a number of user groups of the new system (also known as personas that will typically use our system) and by creating a list of tasks they will perform on the new system.
- A limited number of personas will be interviewed and to get an idea of how they're going to use the system. This translates to a list of task that will be performed on the system.
- For every task we can an estimation can be given on as to how much and how often will the user use the system's resources and functionalities to get a task done.
- **SUMMARY** : Basically estimating the processes and their resource requirements for getting a certain number of jobs done.

PERFORMANCE OF RUNNING SYSTEMS

1. **BOTTLENECKS**

- Performance of a system is based on the performance of all its components, and the interoperability of various components.
- At times it happens that under high load, a **component may reach it's best performance or capacity**.
- This may eventually cause the **overall system to slow down due the the limited performance of that single component**.
- Therefore the **performance of a system as a whole is totally dependant on the performance of that one singe component**.
- This component is referred to **bottleneck**.

BOTTLENECK LAW

According to the **Bottleneck law**, every system, regardless of how well it works, has at least one bottleneck that limits its performance. This is perfectly okay when the bottleneck does not negatively influence performance of the complete system under the highest expected load.

PERFORMANCE TESTING

1. LOAD TESTING : Shows the system performance under the expected load.
2. STRESS TESTING : Shows the system reaction to extreme load. Mainly done to check the breaking point of the system .
3. ENDURANCE TESTING : Shows system behaviour under expected load for a prolonged duration.

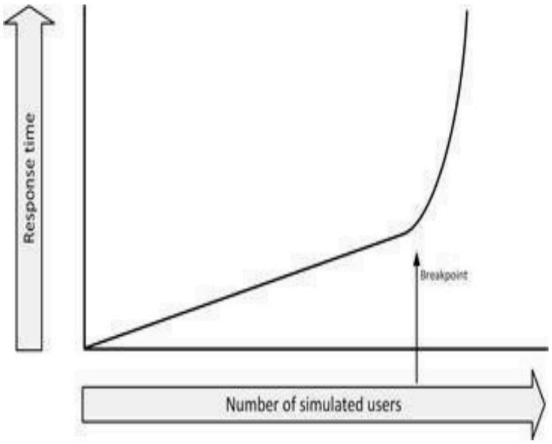


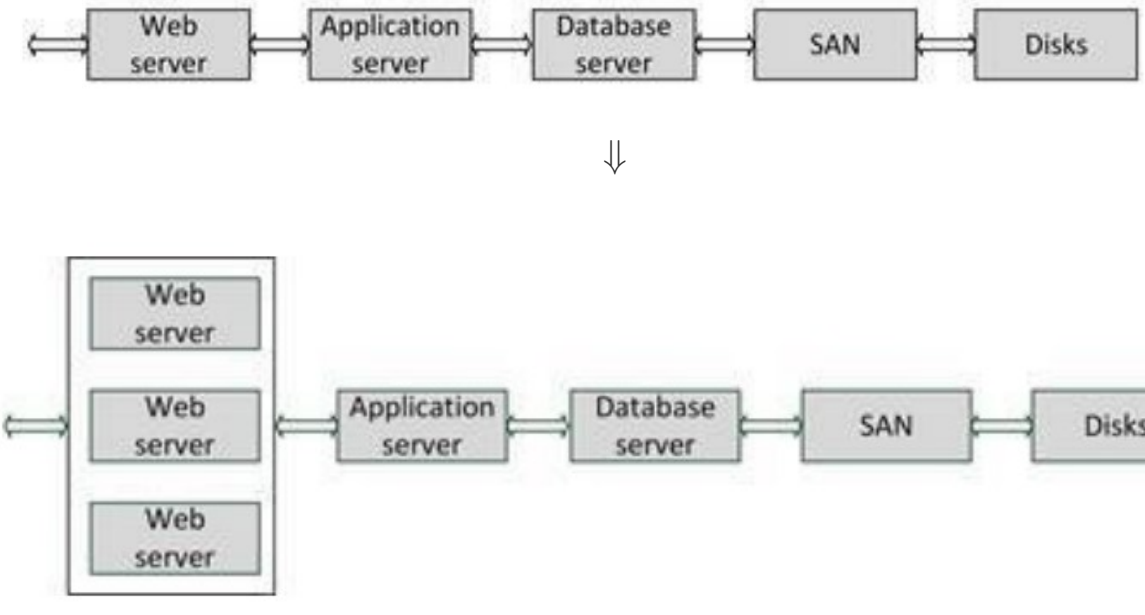
Figure 16: Performance breakpoint

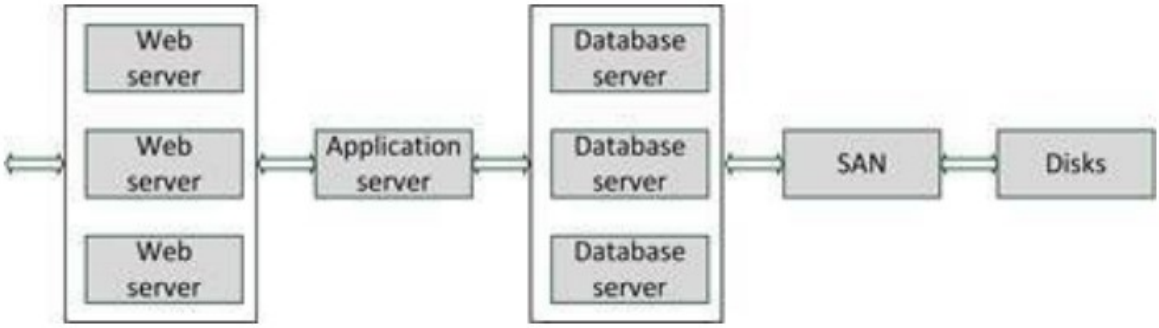
- To carry out performance testing , the following process is followed :
- **TLDR** : emulate some users using $1-2$ servers and use 1 server to coordinate tasks and collect metrics for reporting and analytics. Generally the load is increased gradually to see the performance as the usage increases. Also the testing should be done in a production like environment because the development environment has different settings and configurations from production and may provide unreliable results.

PERFORMANCE PATTERNS

- Performance on the upper layers can be improved by optimising the application and database than just bluntly adding compute power.
- **CACHING**
  - retaining frequently used data in high speed memory, reducing access times to data thereby improving the overall performance.
- **DISK CACHING**
  - Disks being mechanical in nature are inherently slow.
  - To speed up reading of data, they implement disk caching.
  - This is generally implemented within the storage block itself (the very hardware) but is also present in the OS as well.
- **WEB PROXIES**
  - When users searches some information on internet, instead of fetching all requested data from the internet every time, the previously accessed data can be cached in a proxy server and fetched from there.
  - This not only gives the user a faster speed perception but also reduces the overall load on the servers to query and get details.
- **OPERATIONAL DATA STORES**

    - It's a read-only replica of a part of a database.
    - Most read requests are redirected towards the ODS instead of the main database thereby reducing the load on the main Database.
- **FRONT END SERVERS**
    - They store the most accessed parts of a web page like landing page or static images thereby reducing the load on the main web environment.
- **IN MEMORY DATABASES**
    - Here, the whole database can be run from memory instead of from disk.
    - In-memory databases are used in situations where performance is crucial (like in real-time SCADA systems).
    - Special arrangements must be made to ensure data is not lost when a power failure occurs.
- **SCALABILITY**
    - It means the ease with which new components can be added or modified to the existing system to handle more load.
    - There are 2 types of scaling :
        - **VERTICAL SCALING**
            - here we add resources to a single component.
            - generally we add CPUs or memory to a server.
            -
        - **HORIZONTAL SCALING**(aka scale out)
            - Here we add more components to the infrastructure, such as adding a new web server in a pool of web servers, or adding disks in a storage system.
            - It works best when the system is partitioned because in that case, individual components can be scaled up.





    - **NOTE** : Doubling the number of components does not necessarily double the performance. Because of overhead (for instance, the extra scheduling needed in multiprocessor systems, or buffering and link state issues in network connections) doubling components usually only provides about $70\% - 80\%$ performance increase. Adding more components leads to even more diminishing returns.
    - **TLDR For Note** : Doubling the resources also increases the efforts for scheduling the resources and increased network load. Doubling the resources only gives about $70\% - 80\%$ performance increase and not 100% increase.
- **LOAD BALANCER**
    - Load balancer uses multiple components – usually servers – that perform identical tasks.
    - Then redistributes the tasks to members in a server farm.
    - Load balancer observes the current load on each server in the farm and redirects the incoming requests to the least busy server.
    - More advanced load balancers can spread the load based on the number of connections a server has, or the measured response time of a server.
    - For a load balancer to work, servers must be functionally identical to each other
    - For instance, each web server in a load balancing situation must be able to provide the same information.
    - Furthermore, the application running on a load balanced system must be designed to handle requests that can be handled by a different server. The application (or at least the load balanced parts of the application) must be stateless for this to work.
- **HIGH PERFORMANCE CLUSTER**
    - High performance clusters provide a vast amount of computing power by combining many computer systems.
    - Usually a large number of servers are used, connected by a high-speed network like gigabit Ethernet or InfiniBand. Such a combination of relatively small computers can create one large supercomputer.
    - **TLDR** : Combine many computers into a cluster and connect them via high speed network and they act like a single large supercomputer.
- **GRID COMPUTING**

    - **TLDR** : Many High Performing Clusters that are spread across different geographical locations.
- **DESIGN FOR USE**
    - **TLDR** : Basically the design of the architecture should be according to the application it's going to be used for.

UNIT 2

TCP IP PROTOCOL SUITE

- TCP IP is a 4 layered model.
- The layers and the corresponding protocols used in them are :

1. **APPLICATION LAYER**
    - It's responsible for end-to-end communication and error-free data delivery.
    - Main protocols present in this layer : HTTP , SSH , NTP
2. **TRANSPORT LAYER**
    - This layer exchanges data receipt acknowledgments and retransmit missing packets to ensure that packets arrive in order and without error.
    - Main protocols present in this layer : TCP , UDP
3. **NETWORK LAYER**
    - Responsible for routing packets of data from one device to another across a network. It does this by assigning each device a unique IP address, which is used to identify the device and determine the route that packets should take to reach it.
    - Main protocols present in this layer are : IP , ICMP , ARP
4. **NETWORK ACCESS LAYER**
    - Responsible for generating data and initiating connection requests. It operates on behalf of the sender to manage data transmission, while the Network Access layer on the receiver’s end processes and manages incoming data.
    - Main protocols present in this layer are : LAN , WAN

LAN (LARGE AREA NETWORK)

- Used to connect network devices in such a way that personal computers and workstations can share data, tools, and programs. The group of computers and devices are connected together by a switch, or stack of switches, using a private addressing scheme as defined by the TCP/IP protocol.

- LAN covers a smaller geographical area (approximately few kilometres).
- Data transmission speed is fast given the number of connections made.
- Can be used to share peripheral devices such as printers and scanners.

MAN (Metropolitan Area Network)

- covers a larger area than that covered by a LAN and a smaller area as compared to WAN .
- Range is about $5 - 50km$.
- It covers a large geographical area and may serve as an ISP (Internet Service Provider).

(DIDN'T FIND ANYTHING ABOUT HEIRARCHIAL LAN DESIGN)

Multiservice Access Technologies

- They are the network system infrastructure which allow multiple communication service like voice, image, video and internet etc.
- They're designed to efficiently support diverse traffic types while optimizing bandwidth and simplifying network management.
- Mainly has 3 layers :

1. **ACCESS LAYER**
    - The most diverse part of the architecture that connects the end user to service provider's network.
    - It supports transmission and receiving of diverse media types.
    - Handles initial traffic aggregation and ensures reliable connectivity.
    - Supports both active (powered network elements) and passive (no powered components) solutions.
2. **AGGREGATION LAYER**
    - collects traffic from multiple access nodes and prepares it for transport to the core network.
    - Implements VLANs, QoS policies, and traffic shaping.
3. **CORE LAYER**
    - connects the aggregation layer to the internet, data centres, and other service networks.
    - Routes large volumes of aggregated traffic efficiently.
    - Ensures redundancy, load balancing, and low-latency paths.
    - Supports signalling and session management for multiservice applications.

WiFi Protocols

- Full Form : Wireless Fidelity
- The protocols were developed by IEEE

- Each standard has 2 parameters :
  - Speed : Base Unit : Mbps (Mega bytes per second)
  - Frequency : Works in 2 frequency bands : $2.4GHz$ and $5GHz$

Version	Introduced in	Frequency band used	Maximum speed provided
IEEE 802.11a	1999	5 GHz	54 Mbps
IEEE 802.11b	1999	2.4 GHz	11 Mbps
IEEE 802.11g	2003	2.4 GHz	54 Mbps
IEEE 802.11n	2009	Both 2.4 GHz and 5 GHz	600 Mbps
IEEE 802.11ac	2013	5 GHz	1.3 Gbps
IEEE 802.11ax	2019	Both 2.4 GHz and 5 GHz	Up to 10 Gbps

VPN

- Virtual Private Network
- Acts as a private tunnel for your internet traffic, preventing anyone from tracking and monitoring online activity.
- **WORKING**
  - Upon activating a VPN , it connects to a server provided by the VPN provider.
  - Encrypts the data that is to be transmitted.
  - The internet traffic is now routed through the VPN server, which can be located in any country. This makes it appear as though the browsing is being done from the server’s location, masking our actual IP address.
  - Once the data reaches the server, it's decrypted and then sent to the desired location.
  - Once the response is received, the same route is followed for taking back the response to the user.

VPN Type	Description	Use Case	Security	Speed
Remote Access VPN	Allows individuals to connect remotely to a network from anywhere.	Remote workers, traveling professionals	High	Moderate
Site-to-Site VPN	Connects two networks securely over the internet.	Businesses with multiple locations	Very High	High
Mobile VPN	VPN for mobile devices ensuring uninterrupted access while switching networks.	Healthcare, logistics, field workers	High	Moderate
MPLS VPN	A secure, efficient, and scalable solution for large enterprises.	Large enterprises with multiple office sites	Very High	Very High
PPTP VPN	An older VPN protocol known for speed but lacks security.	Legacy systems, basic VPN needs	Low	Very High
L2TP/IPsec VPN	Combines Layer 2 Tunneling Protocol with IPsec for better security.	Corporate environments, reliable security	High	Moderate
OpenVPN	An open-source VPN protocol known for its flexibility and strong encryption.	Advanced users, custom setups	Very High	Moderate
IKEv2/IPsec VPN	A fast and secure protocol that excels in mobile device use.	Mobile users, stable connections	Very High	High

WAN IP Addressing

- A WAN IP address is the public-facing address assigned to the router by your Internet Service Provider (ISP). It identifies your network to the outside world.

NOTE : NEED TO DIVE INTO MORE DETAIL EVEN FROM SYLLABUS POINT OF VIEW.

STORAGE BUILDING BLOCKS

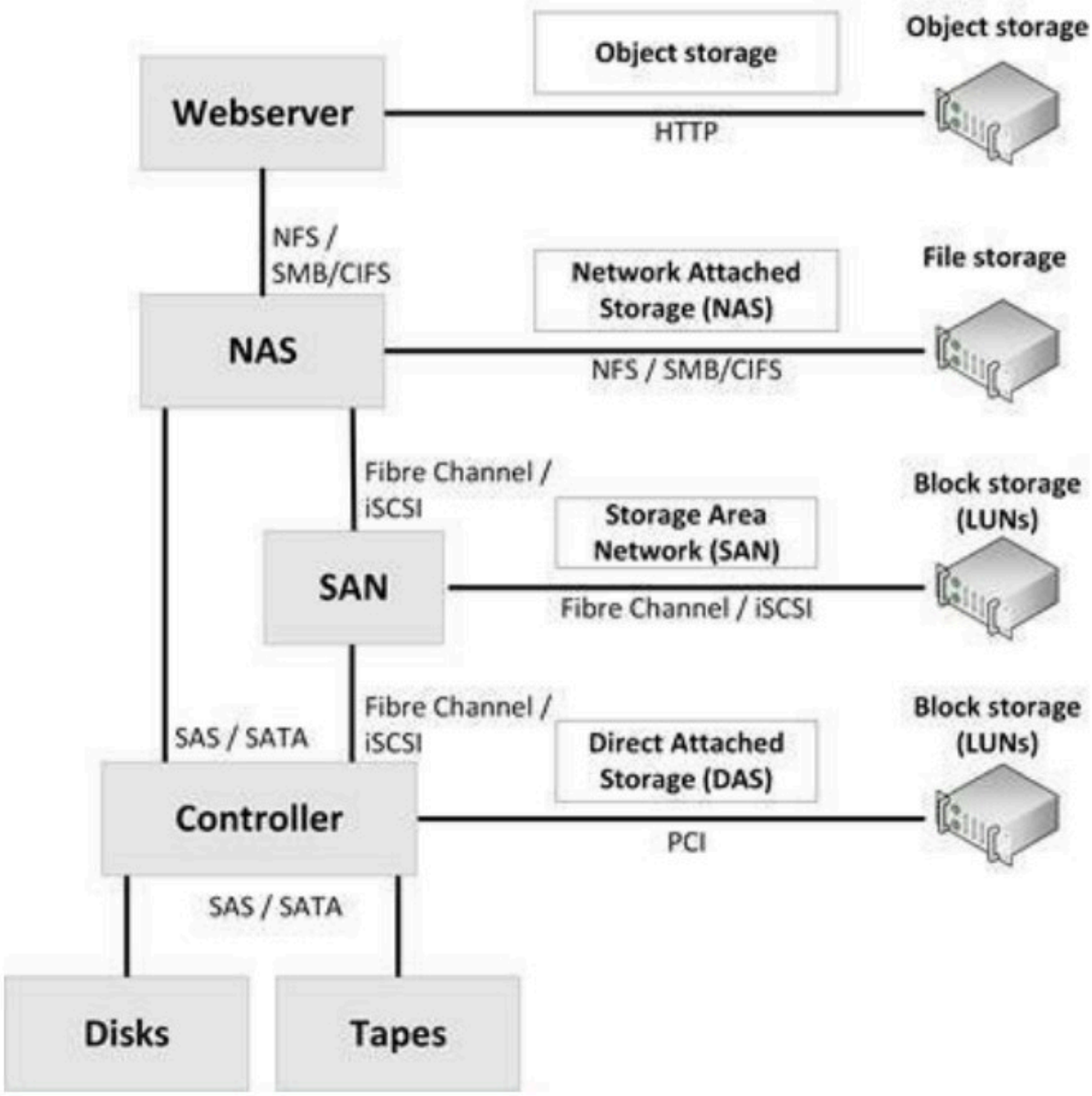


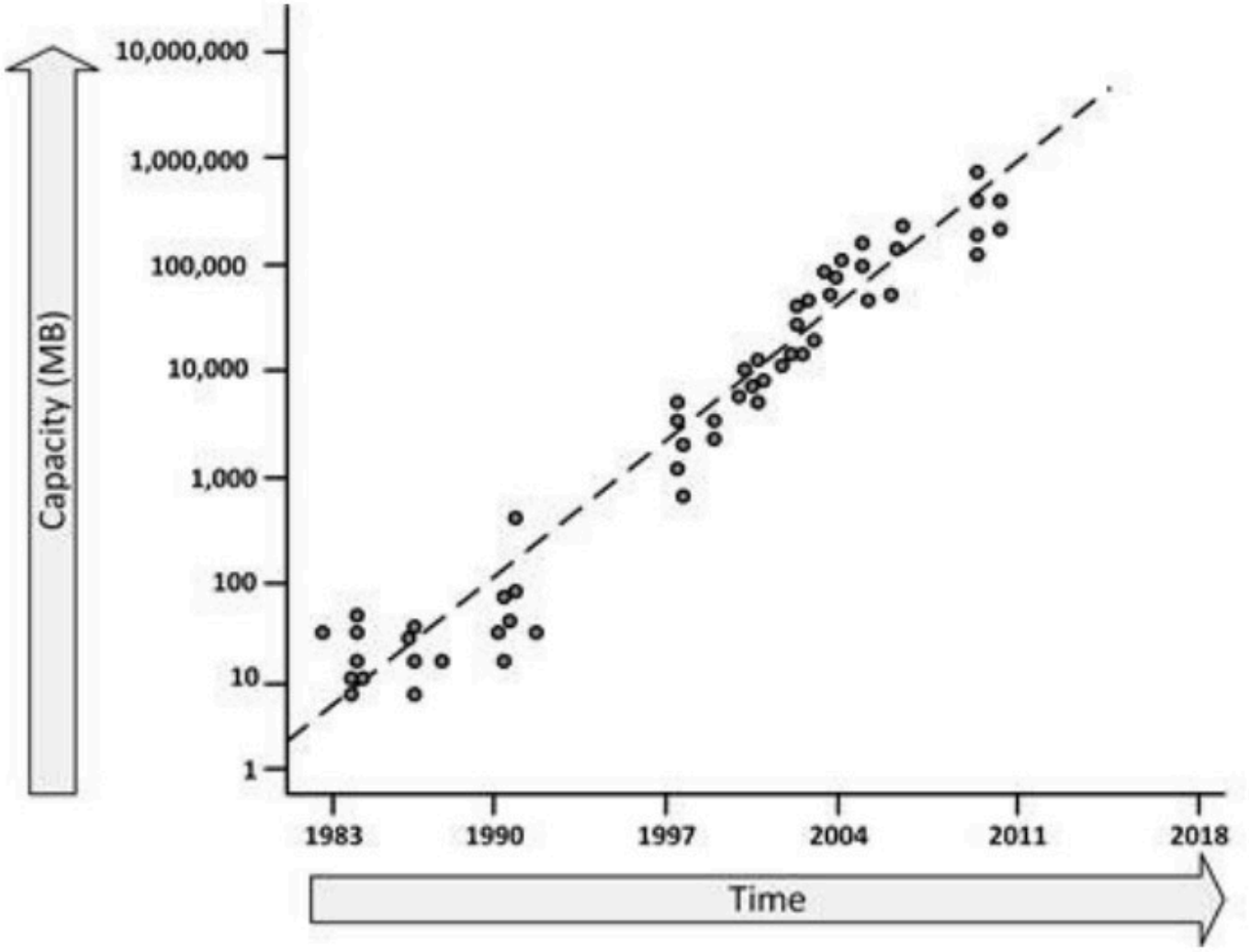
Figure 59: Storage model

- **DISKS**
  - Mainly 2 types are used :
    - Mechanical hard disks
    - Solid State Drives (SSD)
  - Disks are connected to disk controllers using a command set.
  - Command sets are nothing but protocols for communication between device and disk.
  - There are 2 types of command sets :
    - ATA : Advanced Technology Attachment(aka IDE)
      - Uses simple hardware.
      - Mostly used in PCs

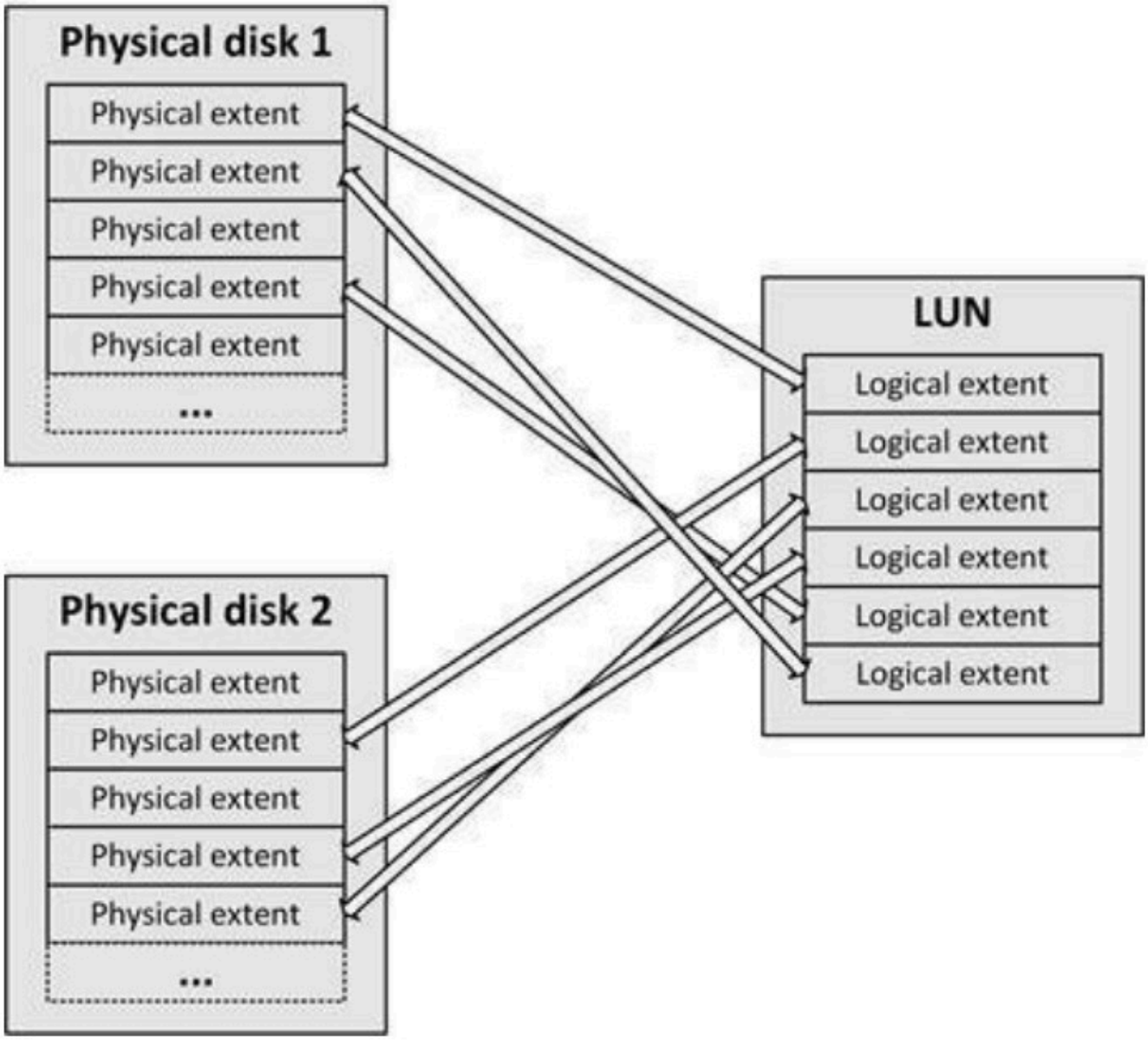
        - SCSI : Small Computer System Interface
            - set of standards for physically connecting and transferring data between computers (mostly servers) and peripheral devices, like disks and tapes.
    - **MECHANICAL HARD DISKS**
        - Consist of vacuum sealed cases with one or more spinning magnetic disks on one spindle.
        - Has many number of read/write heads that can move to reach each part of the spinning disks.
        - Types :
            - Serial ATA (SATA) disks
            - Serial Attached SCSI (SAS) disks
            - Near-Line SAS (NL-SAS) disks
    - **SOLID STATE DRIVES**
        - Doesn't have moving parts.
        - It's based on Flash technology which is a semiconductor-based memory that preserves its information when powered off.
        - SSDs are connected using a standard SAS disk interface.
        - Due to no moving parts, they're faster than mechanical disks.

DISK CAPACITY : KRYDER'S LAW

- the density of information on hard drives has been growing at a rate, increasing by a factor of 1000 in 10.5 years, which roughly corresponds to a doubling every 13 months.



- **TAPES**
  - Tapes are suitable for archiving, since tape manufacturers guarantee a long life expectancy.
  - Not suited for large storage of data.
- **CONTROLLERS**
  - They connect disks and/or tapes to a server, typically implemented as a PCI expansion boards in the server.
  - Controllers implement high performance, high availability, and virtualized storage using RAID (Redundant Arrays of Independent Disks) technology.
  - A controller virtualizes all physical disks connected to it, presenting one or more virtual disks, called Logical Unit Numbers (LUNs).



- Here the all the physical disks are fragmented into small pieces call physical extent. - Using these physical extents, new virtual disks (LUNs) are composed and presented to the OS.

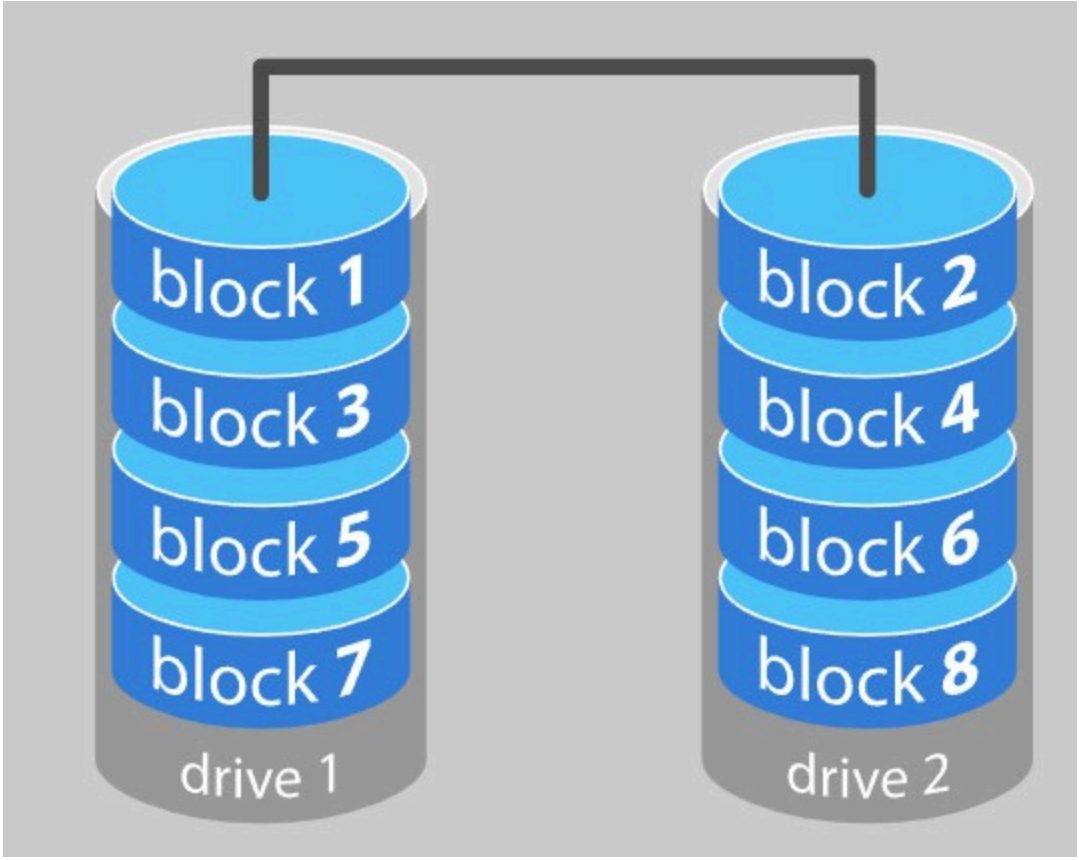
RAID

- There are 5 levels of implementation for RAID :
  - **RAID** 0 **-** Striping
  - **RAID** 1 **-** Mirroring
  - **RAID** 5 **-** Striping with distributed parity
  - **RAID** 6 **-** Striping with distributed double parity
  - **RAID** 10 **-** Striping and Mirroring

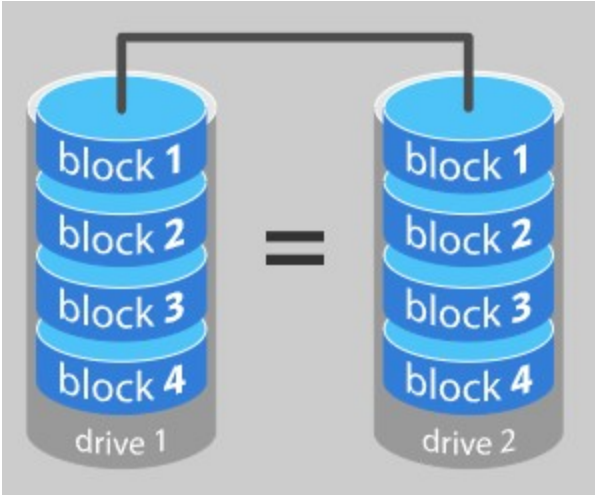
SOURCE FOR RAID THEORY

- **RAID** 0 : **STRIPPING**

- TLDR : Data split into 2 blocks and they are made available at the same time. This makes the IO operations fast.
- Problem : Fault intolerant. If one drive fails, data for that drive is lost.

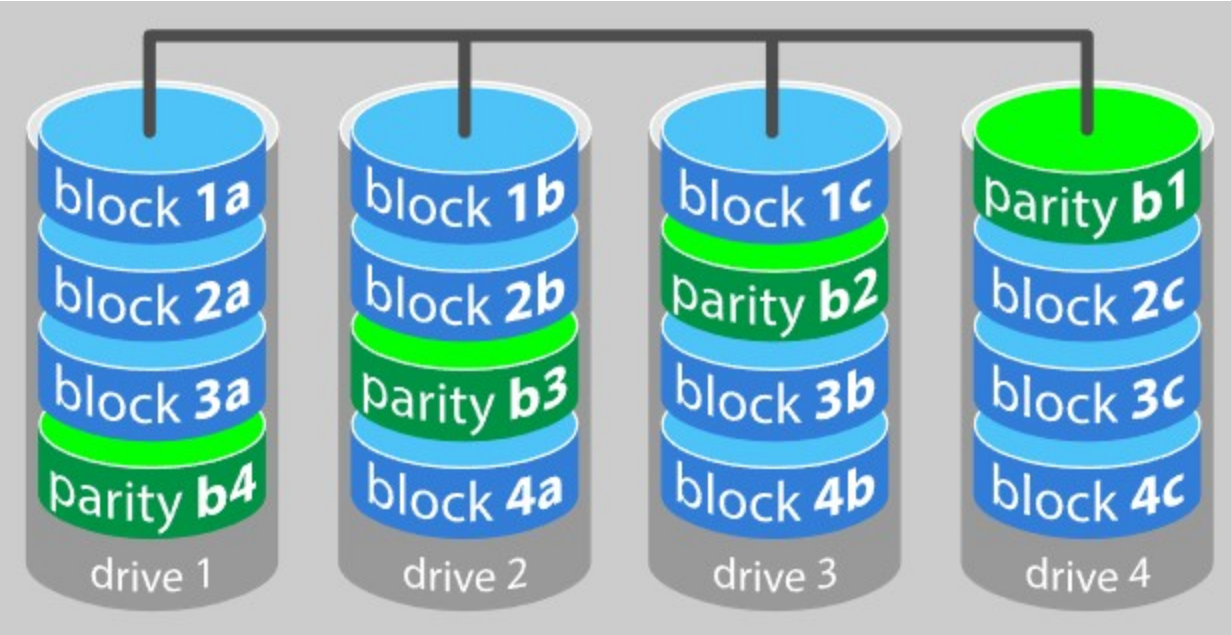


- **RAID** 1 : **MIRRORING**
  - TLDR : Take 2 drives and make the same data available on both the drives .
  - Problem :
    - Due to double writing of data, it's not memory efficient.
    - That means the failed drive can only be replaced after powering down the computer it is attached to.

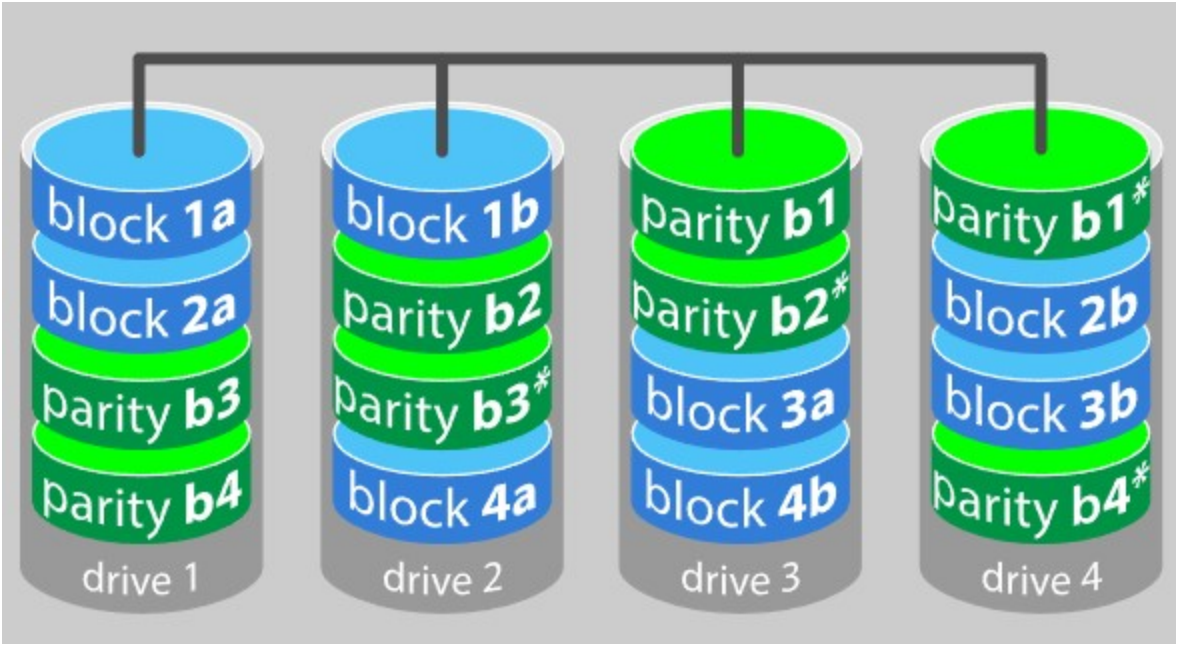


- **RAID** 5 : **STRIPPING WITH PARITY**

- TLDR : Data is stripped across the drives (minimum 3 required). parity checksums of all the data blocks is distributed across all drives. So in case one drive fails, it's data can be reconstructed back .
- Problem :
  - Small Throughput
  - Slow when data volume is large

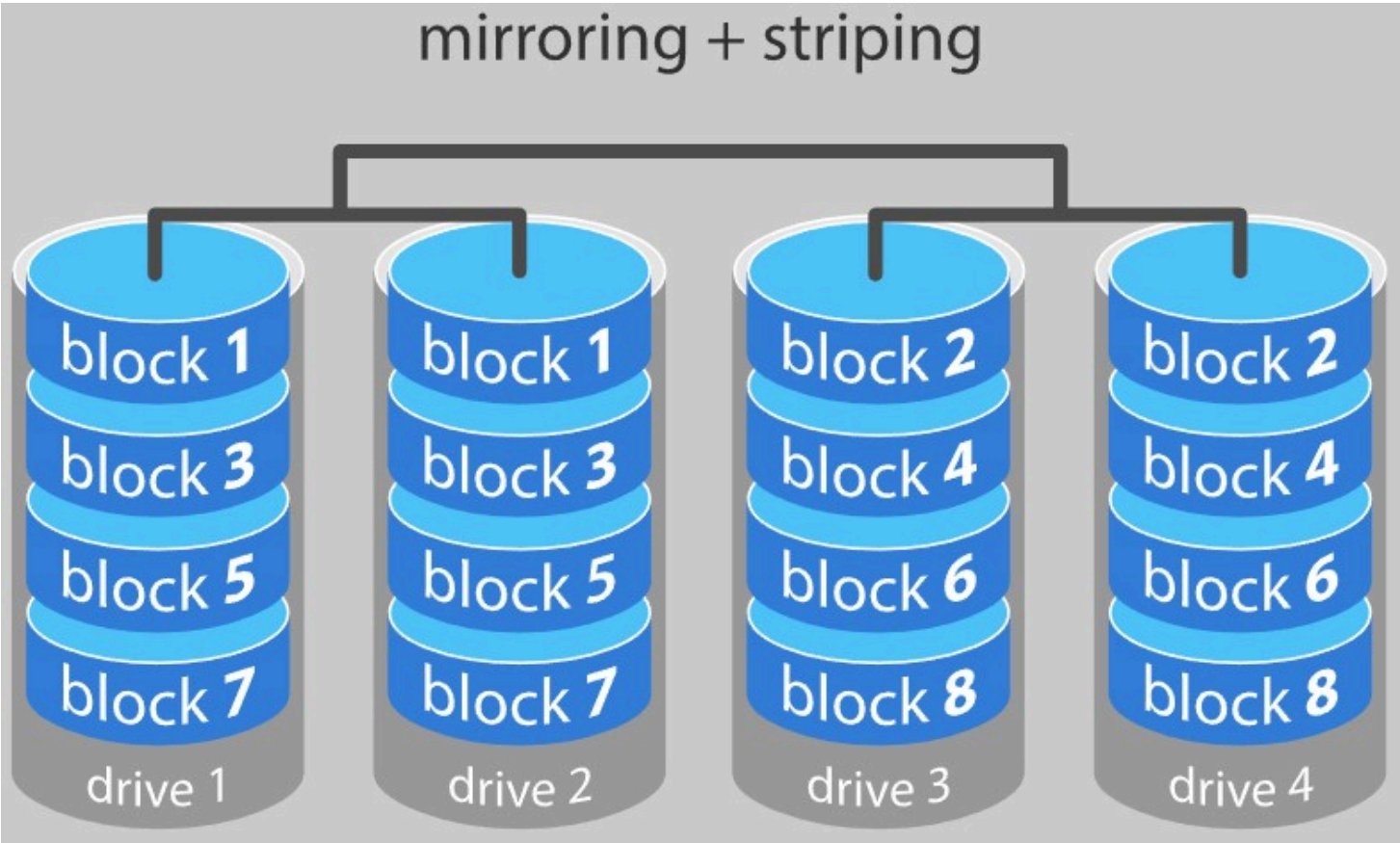


- **RAID** 6 : **STRIPPING WITH DUAL PARITY**
  - TLDR : Same a RAID 5 with the parity being duplicated .
  - Problem :
    - In an event of drive failure, throughput is severely affected.
    - In case of drive failure, rebuild will take long time.



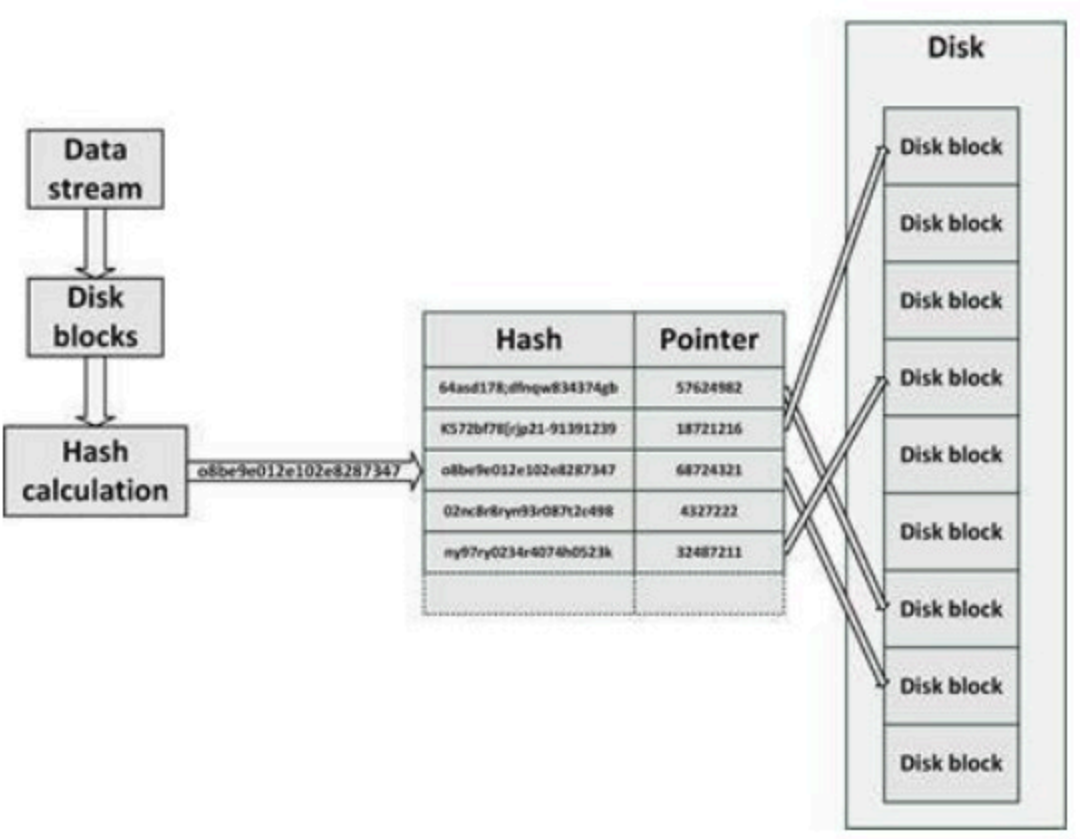
- **RAID** 10 : **STRIPPING & MIRRORING**

- TLDR : Mix of RAID 0 and RAID 1 i.e (figure will explain better).
- Problem :
  - storage inefficient



DATA DEDUPLICATION

TLDR : The deduplication system maintains a hast table which contains the mapping of memory pointers to hash value of the input data. Any duplicate data will not be stored (if the hash value was found in the hash table) but a pointer to the matching segment will be created.



CLONING AND SNAPSHOTS

- Both the processes intend to create a backup of the data.
- **CLONING** : Creates one full-copy of the disk. This cloned disk can be split-off at a specific point in time, for instance to make a backup of the data, without touching the original disks that are still on-line (i.e the data on original disk is still under change due to write/delete operations).
- **SNAPSHOTS** : Snapshot represents a point of time of the data on the disk. From the time the snapshot is active, no write operation is allowed. All write operation is done in a separate disk volume in memory. read operations are permitted for the frozen data. In case of read operation for updated data, the data is retrieved from the disk volume. A backup is created while the snapshot state is active and as soon as it's over the data waiting to be written on the original data on the disk is written.

THIN PROVISIONING

NOTE : PENDING. TO BE STUDIED (WENT BOUNCER)

DIRECT ATTACHED STORAGE (DAS)