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.

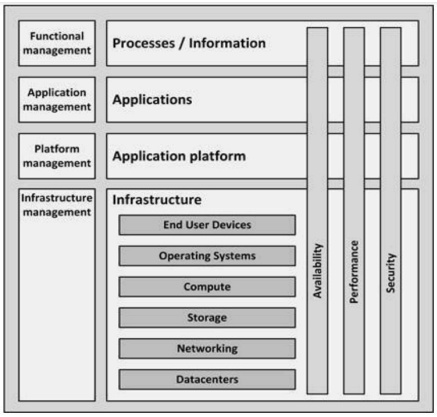


Figure 2: The infrastructure model

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

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

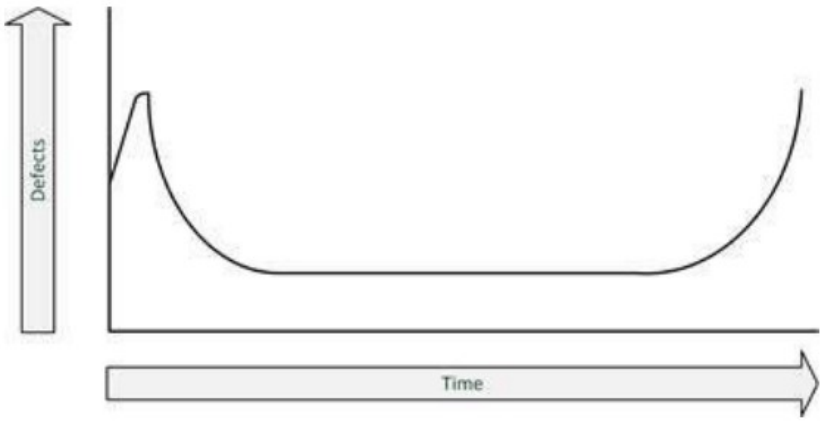
- $\implies$ The more components a system includes (and each component is critical for the total system) the lower the total availability becomes.
- Hence we try to avoid many systems in series and rather keep them in parallel.
- For parallel systems :
- Say :
    - There are N components out of which 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**
    - **User Misuse**: Overloading systems (e.g., generating multiple large reports) or accidental lockouts (e.g., repeated password failures).
    - Administrative Mistakes: Switching off the wrong server, restoring incorrect backups, mislabelling cables, or typos in commands (e.g., `sudo rm -rf / *.back` vs. `sudo rm -rf /*.back`).
    - **Testing Gaps**: Untested failover procedures or maintenance in production environments.

    - **Mitigation**:
        - Standardized procedures
        - Restricted administrative access
        - Automated deployment tools
        - User awareness (e.g., UNIX login warnings).
- Software Bugs :
    - Bugs in applications, drivers, or operating systems (e.g., Windows' Blue Screen of Death) can crash systems, corrupt files, or disrupt networks.
- Physical Defects :
    - Common Failures like Fans (dust-clogged bearings), disk drives (motor wear), tapes/robots (mechanical stress), and aging capacitors/batteries.
    - Environmental Factors like Temperature fluctuations, moisture, vibrations, and frequent power cycling accelerate degradation.

BATHTHUB CURVE (COMES UNDER PHYSCIAL DEFECTS)



- 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.
- Environmental Effect (Earthquakes, Fire accidents etc.) can also lead to unforeseen scenarios of unavailability.

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.

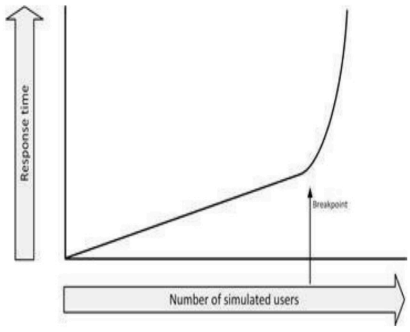
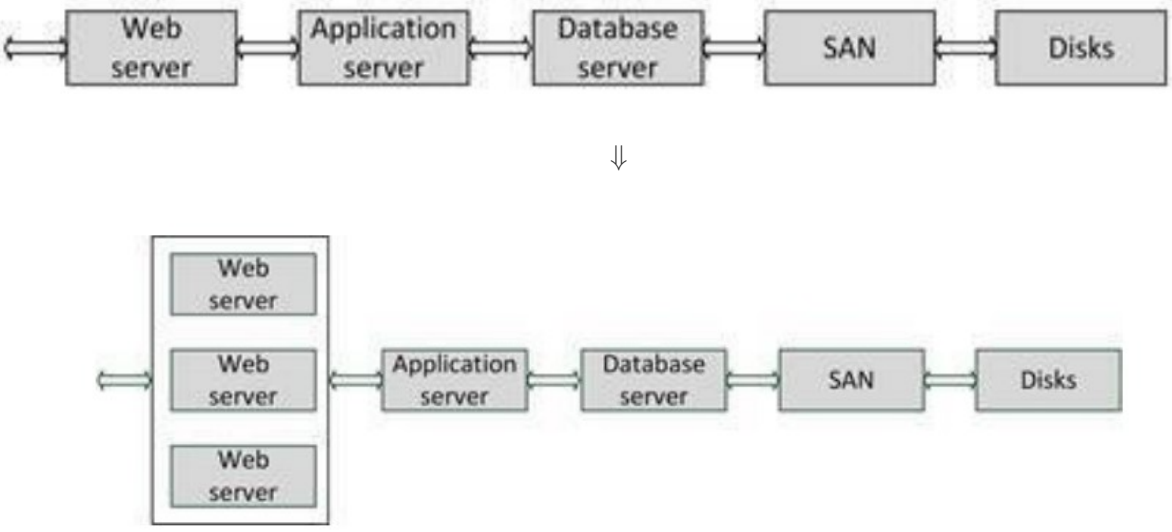


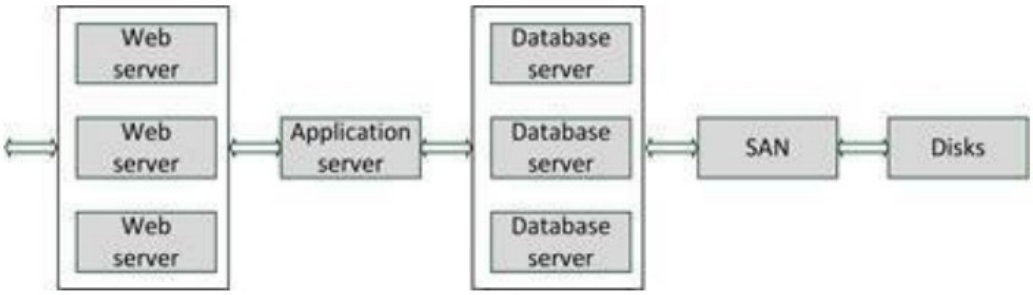
Figure 16: Performance breakpoint

- 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.
      - Basically adding more compute power to a single component be it CPU, GPU , Harddisk etc.
    - **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

VPN Type	Description	Use Case	Security	Speed
L2TP/IPsec VPN	Combines Layer 2 Tunnelling 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

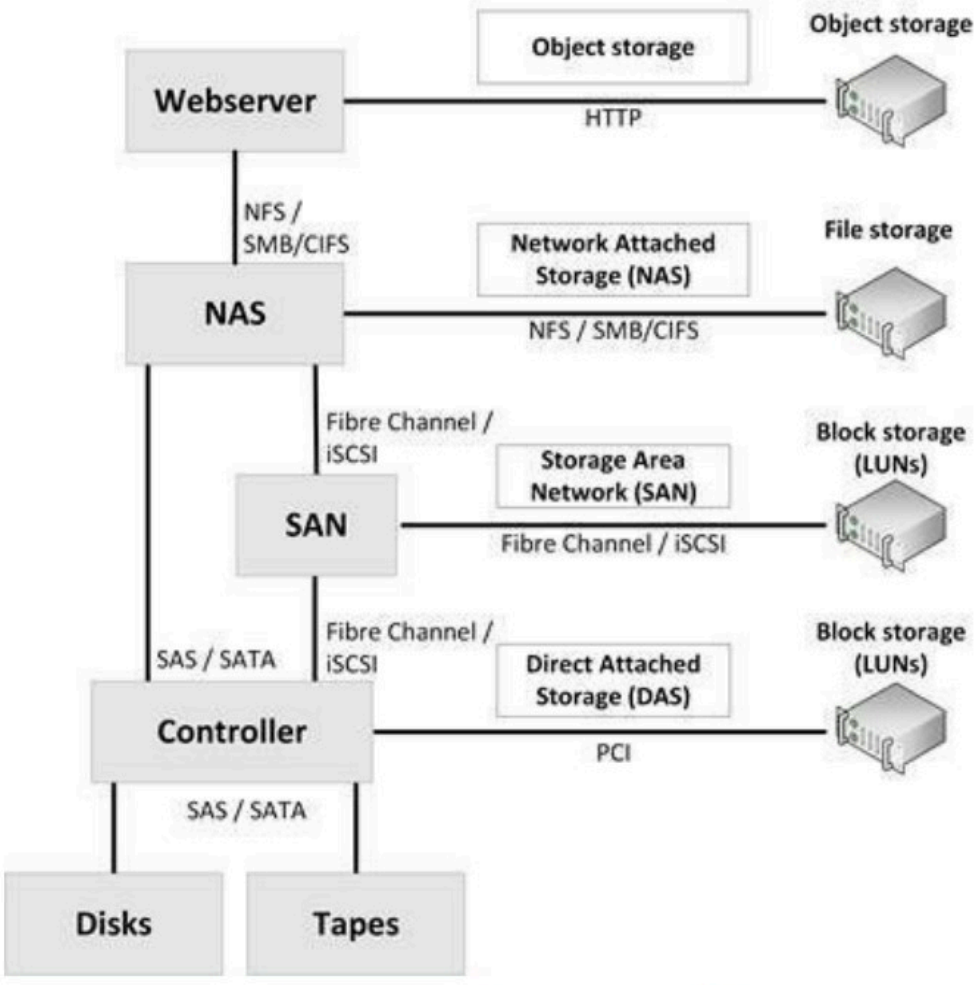


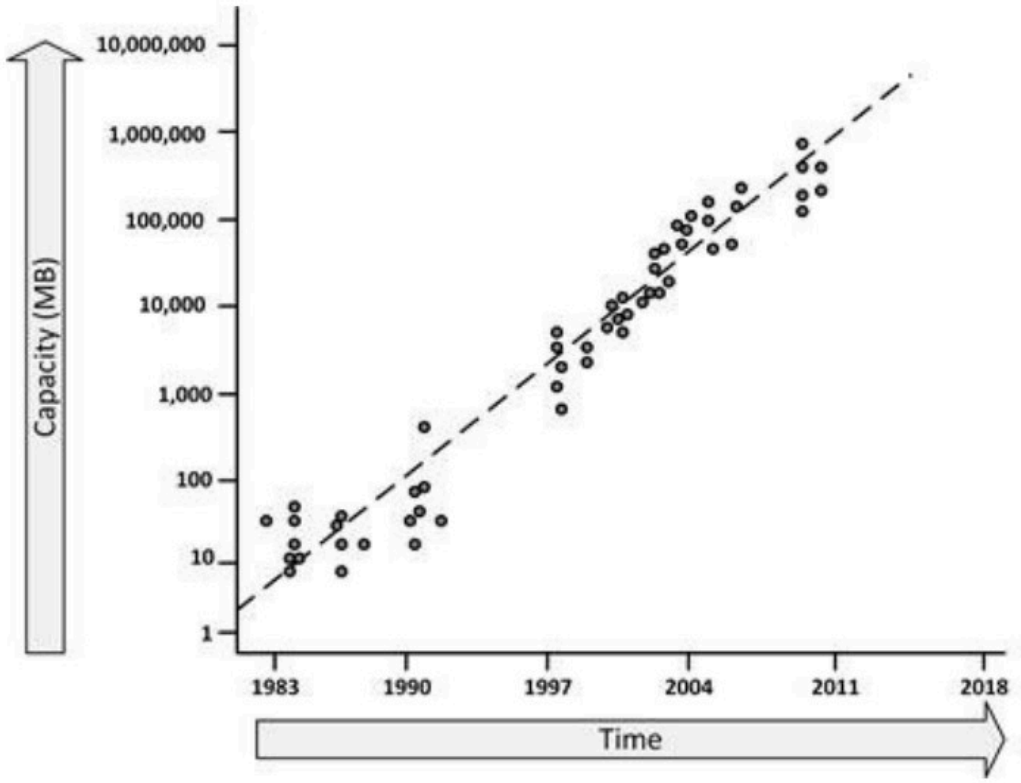
Figure 59: Storage model

- **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.

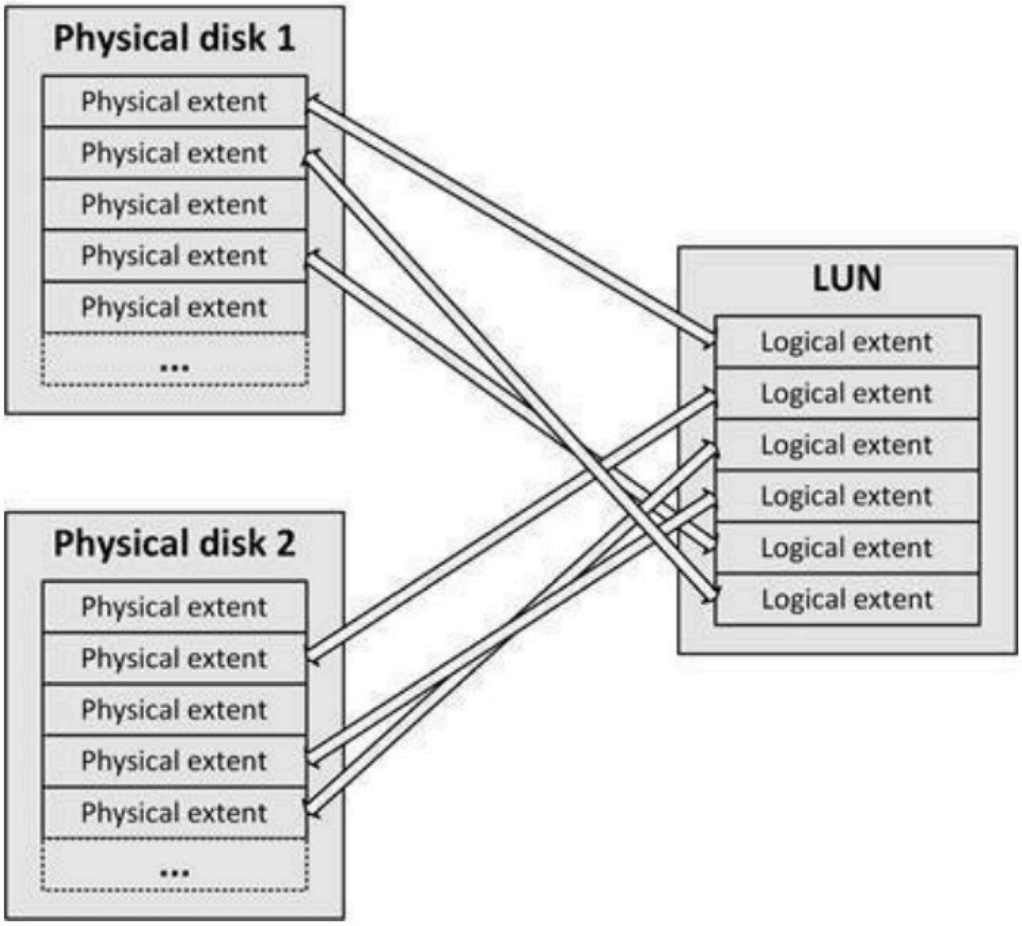
DISK CAPACITY : KRYDER'S LAW

- 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).



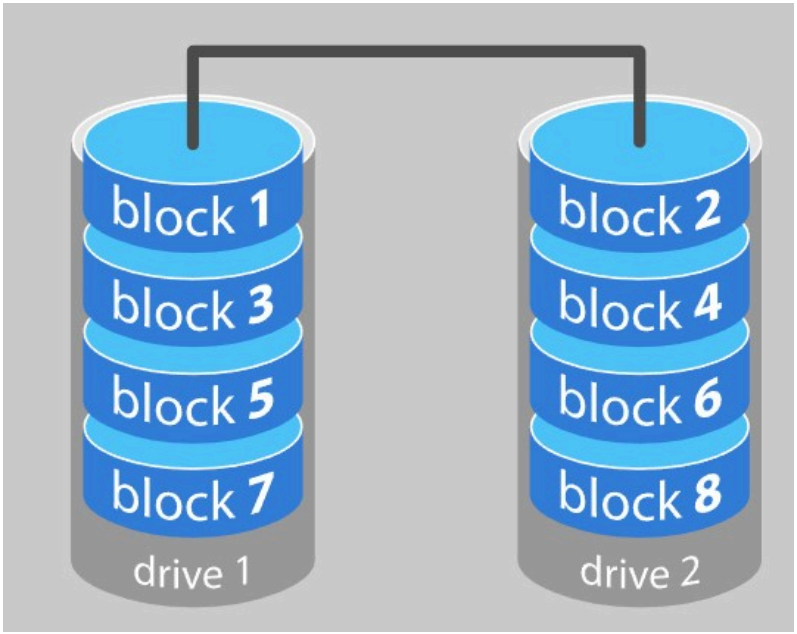
- Here the all the physical disks are fragmented into small pieces call physical extent. - Using these physical extents, new virtual disks (LUNs) are composed and presented to the OS.

RAID

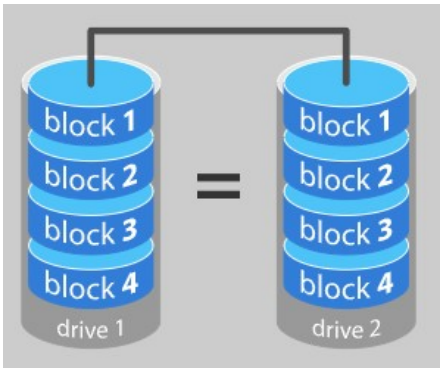
- There are 5 levels of implementation for RAID :
  - **RAID 0 -** Striping
  - **RAID 1 -** Mirroring
  - **RAID 5 -** Striping with distributed parity
  - **RAID 6 -** Striping with distributed double parity
  - **RAID 10 -** Striping and Mirroring

SOURCE FOR RAID THEORY

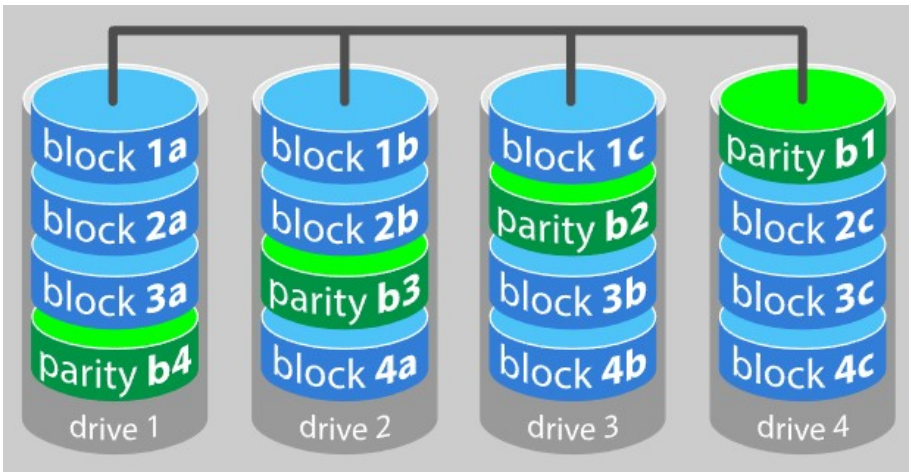
- **RAID 0** : **STRIPPING**
  - TLDR : Data split into 2 blocks and they are made available at the same time. This makes the IO operations fast.
  - Problem : Fault intolerant. If one drive fails, data for that drive is lost.



- **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.

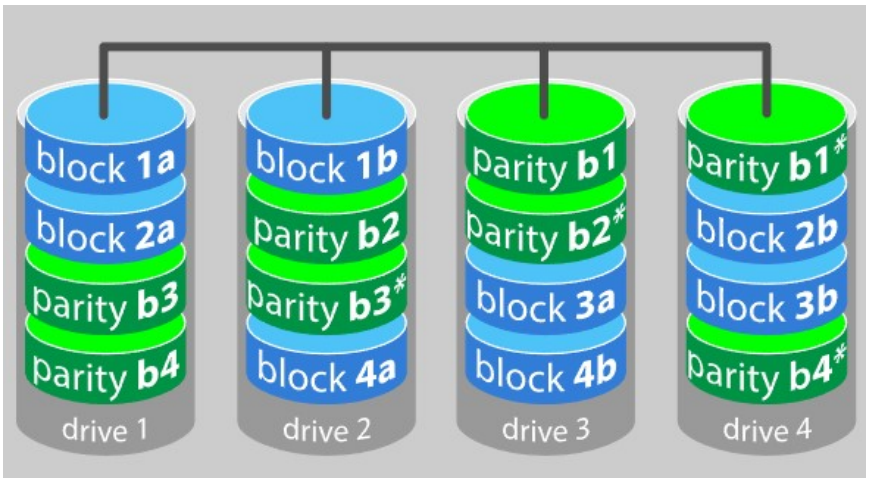


- **RAID** 5 : **STRIPPING WITH PARITY**
  - TLDR : Data is stripped across the drives (minimum 3 required). parity checksums of all the data blocks is distributed across all drives. So in case one drive fails, it's data can be reconstructed back .
  - Problem :
    - Small Throughput
    - Slow when data volume is large

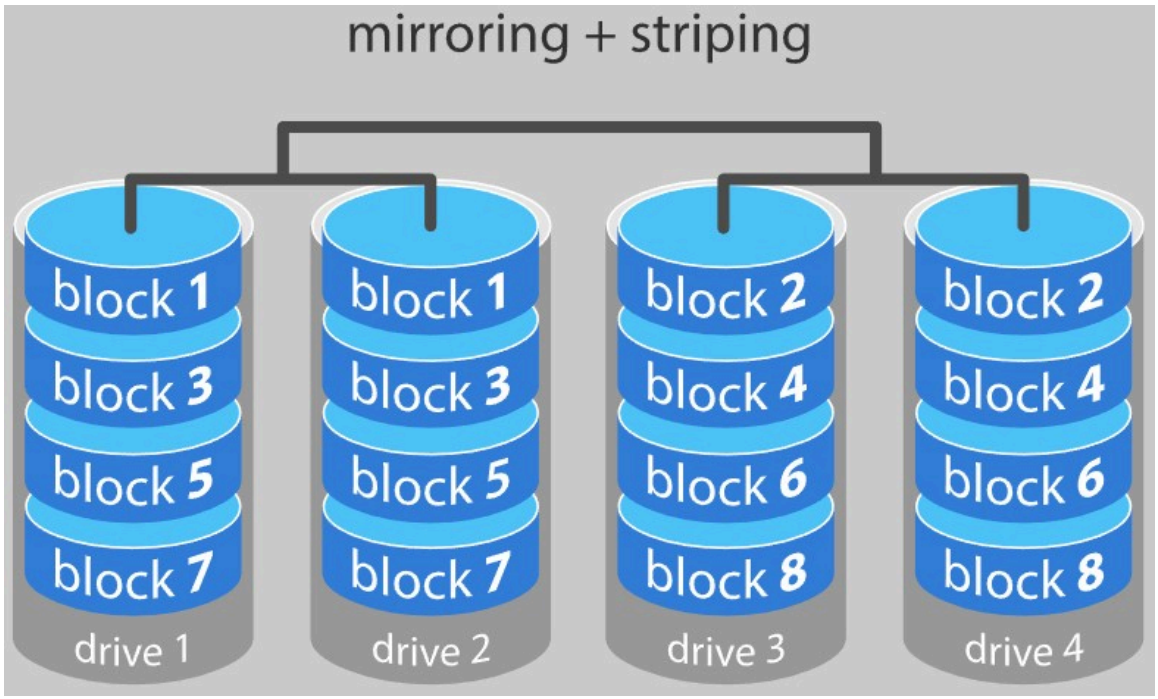


- **RAID** 6 : **STRIPPING WITH DUAL PARITY**

- 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.

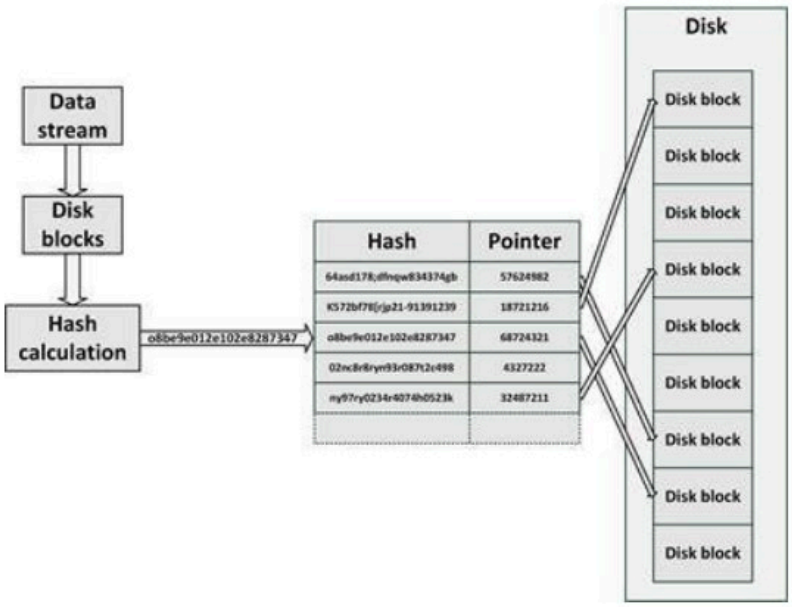


- **RAID** 10 : **STRIPPING & MIRRORING**
  - TLDR : Mix of RAID 0 and RAID 1 i.e (figure will explain better).
  - Problem :
    - storage inefficient



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

- Before jumping there , what's provisioning ?
- It's basically the allocation and configuration of resources like storage, compute power, network capacity etc.
- Now comes the 2 types :
- **THICK STORAGE** :
  - when a storage volume is created, the entire allocated space is immediately reserved for that volume, even if it is not fully used .
  - one user might have access to $40GB$ of storage. In thick provisioning, that user would continue to have exclusive access to the full $40GB$ even if they only use $20GB$.
- **THIN PROVISIONING** :
  - It's based on the principle of overcommitting resources i.e presenting more logical storage to the user than the physical storage actually available.
  - But how it's done ?
  - A virtual storage is created (way more than the actual physical storage in general say $1TB$) but only a small part of the physical memory is truly allocated (say $100GB$)
  - With time , as the application needs memory, more memory is allotted to it.
  - The resource consumption is monitored and resources are freed as soon as the usage goes low.

DIRECT ATTACHED STORAGE (DAS)

- Basically local storage.
- A storage system where one or more disks are connected via SATA protocol to a controller.
- This controller provides a disk block to the computer organized by LUNs.
- These LUNs are then used by OS to manage memory.

- In serves, DAS is mainly used for boot device or caching.

STORAGE AREA NETWORK (SAN):

- A specialized storage network that consists of SAN switches, controllers and devices.
- It's task is to connect large storage to multiple servers .
- SAN physically connects these servers disk controllers via special networking technology like Fibre Channel
- Data is accessed through LUNs (Logical Unit Numbers), which are partitions of storage volumes.

Hardware Bus Adapters (HBAs) :

- A hardware interface card that acts as a bridge between a server and storage devices in a network (Like SANs).
- It's main job is to offload data transfer tasks from the server's CPU and provide high-speed data transmission between the server and storage devices.

SAN CONNECTIVITY PROTOCOLS

- Needed to connect Storage Devices to Servers via SAN.
- There are 3 :
  - Fibre Channel
  - FCoE
  - iSCSI
- (Syllabus has Fibre Channel I'll do that only)
- **FIBRE CHANNEL** :
  - 2 Layer Network protocol specially for transporting storage data blocks.
  - operates at speeds of 2Gbit/s, 4Gbit/s, 8Gbit/s, or 16Gbit/s.
  - Fibre Channel can run on both twisted pair copper wire (i.e. UTP and STP) and fibre optic cables.
  - This Protocol is very reliable, with guaranteed zero data loss.
  - Each Fibre Channel device has a unique World Wide Name (WWN), which is similar to an Ethernet MAC address.
  - Fibre Channel can be implemented in 3 different topologies :
    - **Point-to-Point** : Two devices are connected directly to each other.
    - **Arbitrated loop** (aka FC-AL) : In this topology, all devices are in a loop. Most early Fibre Channel systems worked this way.
    - **Switched fabric** - All devices are connected to Fibre Channel switches, a similar concept as in Ethernet implementations. Most implementations today use a switched fabric.

Network Attached storage (NAS) :

- It's a network device that provides a file system storage to the OS on a regular TCP/IP network.
- It's generally an appliance (basically it's own OS and hardware)that implements file services and holds the disk on which the data is stored (In some cases NAS is used only as a gateway that provides file services but actual storage is present elsewhere) .
- **Clustered NAS** : A NAS that uses distributed file system running simultaneously on multiple servers .

SAN Vs NAS:

SAN	NAS
Provides block level storage	Provides a shared file system
Provides access to one server per LUN	Multiple servers can access the same filesystem
No such feature available	Uses active directory service or LDAP for managing file permissions
No file level services available	Provides redundancy, load balancing, replication, and snapshot technology
No default support for file recovery available	File recovery features like file-level snapshots and file un-deletion available

OBJECT STORAGE :

- A storage architecture where the data is stored as objects.
- The object is defined as a file with metadata and a globally unique identifier called Object ID.
- Metadata can be filename , date and time stamps, owner, access permissions, the level of data protection, and replication settings .
- Each file (object) can be uniquely identified and accessed using it's object storage without knowing it's actual location.
- Data in object storage can't be modified. Instead, if a file is to be modified, the original file must be deleted and a new file must be created, thereby creating a new object ID.
- Due to this, it's not suited for data needing frequent access and modification.
- But this is useful for storing archives, backups etc which isn't updated much.

STORAGE SECURITY

- Data at rest : This is data in the stored in disks and drives.
- **DISK ENCRYPTION** :
  - **Self-Encrypting Drives (SEDs)** :
    - It's built on the device hardware itself.
    - The data is auto encrypted before being written on the disk.
    - Encryption keys are again stored in the disk itself.
    - Authentication is required to access the data i.e the user needs to provide a password to start the boot sequence for decryption.
  - **Cryptographic Disk Erasure (CDE)** :
    - It's meant to delete the encryption key on the disk.
    - Same as deleting the disk data as the encrypted data can no longer be read from the disk.
  - Disk encryption is primarily useful when disks are physically lost or stolen.
  - Disks in data-centres are rarely stolen, unlike laptops, desktops, or removable media.
  - Maintenance contracts between the vendor and data-centre may require returning failed disks to vendors.
  - If a failed disk cannot be erased, returning it without encryption risks data exposure.
- **TAPE ENCRYPTION** :
  - Since tapes are easy to carry, therefore encrypting them becomes more important than ever.
  - self-encrypting tape drives are be used to encrypt data on the tapes.
  - LTO tape drives come with AES-256 encryption in their hardware.
  - To access a tape, the original keys must be used to decrypt the data, as without keys, encrypted data becomes inaccessible and is effectively destroyed.
- **SAN ZONING** :
  - A method in which Fibre Channel Devices are organised into logical groups within a SAN fabric.
  - Devices within a SAN can only communicate with others in the same zone.
  - This restriction makes it harder for hackers and viruses to access all disks in a SAN.
  - The SAN switch inspects all packets on the fabric. It only forwards packets to ports that are authorized to receive them.
  - Zoning ensures that operating systems only see the LUNs associated with them, instead of all LUNs in the SAN. This prevents operating systems from altering data on disks that don't belong to them.
- **SAN LUN masking** :
  - In this methodology, LUN available to some hosts and unavailable to other hosts.
  - LUN masking is implemented primarily at the HBA level, not in the SAN switch.

UNIT 3

CLOUD COMPUTING

- It refers to the setup where the storage of data , access of data and computations on data , all 3 are done on remote servers hosted on internet instead of using local server or hard-drive.

LAYERD ARCHITECTURE OF CLOUD COMPUTING

- There are 4 layers primarily :

1. **APPLICAITON LAYER** :
   - This is the place where the applications based on cloud are located.
   - They take advantage of automatic scaling for better performance, functionality and low cost.
   - Applications are divided into Execution layers and Application layers.
   - Application layer determines whether communication partners are available or whether enough cloud resources are accessible for the required communication .
   - To be specific , Application layer is responsible for processing IP traffic handling protocols like Telnet and FTP.
2. **PLATFORM LAYER** :
   - The operating system and application software make up this layer.
   - The objective of this layer is to deploy applications directly on virtual machines.
   - The user should be able to rely on the platform to provide them with Scalability, Dependability, and Security Protection.
   - The platform layer's goal is to lessen the difficulty of deploying programmers directly into VM containers.
3. **INFRASTRUCTURE LAYER** :
   - This layer serves as the Central Hub of the Cloud Environment, where resources are regularly added by utilizing a variety of virtualization techniques.
   - It's main task is to divide the physical resources into virtual resources using virtualization technologies.
   - Virtualization technologies like Xen, KVM, Hyper-V, and VMware are used to create a pool of compute and storage resources.
   - Enables dynamic resource assignment which optimizes the resource consumption and performance of the application.
4. **DATACENTRE LAYER** :
   - Responsible for Managing Physical Resources such as servers, switches, routers, power supplies, and cooling systems. (BASICALLY Hardware handling)
   - Physical servers are connect through high-speed devices such as routers and switches to the data centre for fast paced data transfer and computing.

CLOUD DEPLOYMENT MODELS

1. Public Cloud
2. Private Cloud
3. Hybrid Cloud
4. Community Cloud
5. Multi-Cloud

- **PUBLIC CLOUD** :
  - Here, infrastructure and services are provided over the internet to the general public or industries.
  - Service provider owns the infrastructure, and users pay on a subscription or per-use basis.
  - Advantages:
    - Minimal Investment
    - No Setup Cost
    - No Infrastructure Management
    - No Maintenance
    - Dynamic Scalability
  - Disadvantages:
    - Less Secure
    - Limited Customization
- **PRIVATE CLOUD** :
  - Dedicated to a single organization
  - Provides full control over infrastructure, security, and resources.
  - Managed by an organization's own IT department with firewall protection.
  - AKA Internal Cloud

    - Advantages:
        - Better Control
        - Higher Security & Privacy
        - Supports Legacy Systems
        - High Degree of Customization
    - Disadvantages:
        - Less Scalable
        - Higher Cost
- **HYBRID CLOUD** :
    - It's like a mix of public and private clouds with a specialized software.
    - Helps organizations to store sensitive data in private clouds while using public clouds for cost efficiency.
    - Advantages:
        - Flexibility & Control
        - Cost Efficiency
        - Secure
    - Disadvantages:
        - Complex to Manage
        - Slow Data Transmission
- **COMMUNITY CLOUD** :
    - It's a shared cloud infrastructure for multiple organizations with common goals or industry requirements.
    - Generally Managed by either a third party or a group of organizations.
    - Advantages:
        - Cost-Effective
        - Higher Security
        - Shared Resources
        - Supports Collaboration & Data Sharing
    - Disadvantages:
        - Limited Scalability
        - Rigid Customization
- **MULTI CLOUD** :
    - Uses multiple public clouds.
    - Distributes workloads across different cloud providers thereby reducing downtime.
    - Advantages:
        - Best Features of Multiple Providers
        - Reduced Latency
        - High Availability
    - Disadvantages:
        - Complex System
        - Security Risks

TLDR :

Cloud Model	Key Features	Advantages	Disadvantages
Public Cloud	Open to public, provider-owned infrastructure	Low cost, no setup, no maintenance, high scalability	Less secure, limited customization
Private Cloud	Dedicated to one organization, full control	High security, supports legacy systems, customizable	High cost, limited scalability
Hybrid Cloud	Mix of public & private clouds	Flexible, cost-efficient, secure	Complex management, slower data transmission

Cloud Model	Key Features	Advantages	Disadvantages
Community Cloud	Shared by multiple organizations with common goals	Cost-effective, secure, promotes collaboration	Limited scalability, rigid customization
Multi-Cloud	Uses multiple public clouds	High availability, reduced latency	Complex, security concerns

COMPUTE BUILDING BLOCKS

- **PROCESSORS** :
  - A CPU (Processor) executes a set of instructions to get a task done.
  - A typical CPU instruction set consists of a fixed number of instructions like ADD, SHIFT BITS, MOVE DATA, and JUMP TO CODE LOCATION etc.
  - Assembly Language is a set of English Mnemonics that map to Machine Instruction code which is in-turn mapped to a binary instruction.
  - Assembly programming language is the lowest level programming language for computers .
  - A CPU needs a high frequency clock to operate, generating clock cycles.
  - Each instruction needs a single or multiple clock cycles to complete.
  - The speed at which the CPU operates is defined in GHz (billions of clock ticks per second).
  - Because of these high clock speeds CPUs are able to execute instructions very fast.
  - Each CPU is designed to handle data in chunks.
  - These chunks are called words which have with a specific size.
  - The word size is reflected in many aspects of a CPU's structure and operation.
- **RAM** :
  - Random Access Memory
  - Any piece of data stored in RAM can be accessed in the same amount of time, regardless of its physical location.
  - Transistor based technology.
  - Implemented using Integrated Circuits.
  - SRAM (Static RAM)
    - Uses flip-flop circuits to store data.
    - Requires 6 transistors per bit.
    - Faster access times compared to DRAM.
    - Commonly used in cache memory and video RAM.
  - DRAM (Dynamic RAM)
    - Uses one transistor and one capacitor per bit.
    - Stores data as a charge in a capacitor $(charged = 1, discharged = 0)$.
    - More memory-efficient than SRAM due to fewer required components.
    - Needs regular refreshing to retain data .
    - Special hardware circuits handle the refresh process automatically.
- **INTERFACE** :
  - Interfaces are the point of connection between the computer and external peripherals.
  - External Interfaces :
    - $RS-232$
    - USB
    - Thunderbolt
    - They use the connectors located outside the computer.
  - Internal Interfaces :
    - PCI
    - PCIe
    - They're located on the system board of the computer.
  - $RS-232$:
    - Standardized serial communication interface used for connecting electric typewriters, modems, and electronic terminals.

- **USB** :
  - Offers higher speeds, lower power consumption, and plug-and-play functionality.
  - The USB interface can provide operating power to attached devices.
- **THUNDERBOLT** :
  - AKA Light Peak
  - Appeared first on Apple laptops.
  - It's backward compatible
- **PCI**:
  - Peripheral Component Interconnect.
  - It's a shared parallel bus architecture.
  - Allows communication between System board and PCI adapters
- **PCIe** :
  - PCI Express
  - It's a faster alternative, uses point-to-point serial links instead of a shared bus, with links built from multiple lanes (x1, x2, x4, etc).
  - A hub on the system board routes PCIe connections, allowing multiple devices to communicate simultaneously.

COMPUTE VIRTUALIZATION

- AKA Server virtualization AKA Software Defined Compute
- It introduces an abstraction layer between computer hardware and the OS that uses the hardware.
- Virtualization allows multiple OS to run on a single physical machine.
- Virtualization decouples and isolates virtual machines from the physical machine and from other virtual machines by means of a virtualization layer.
- A virtual machine is a logical representation of a physical computer in software.
- Virtualization allows for new virtual machines to be provisioned as needed, without the need for an upfront hardware purchase.
- This saves a lot of time and cost effective.
- consolidating many physical computers as virtual machines on less number but bigger physical machines, huge amounts of money can be saved on hardware, power, and cooling as fewer physical machines are needed and so the cost of maintenance and chances of hardware failure is reduced.

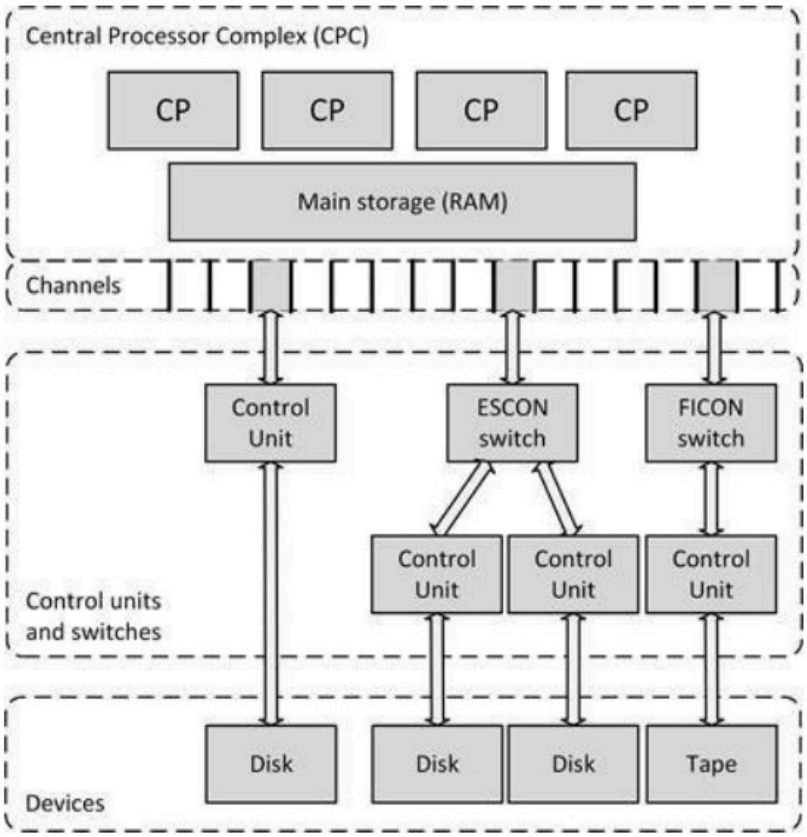
CONTAINER VIRTUALIZATION

- OS was designed to handle a variety of tasks at a time.
- But with time , dependencies, versions and application requirements change and handling them becomes difficulty in the same system.
- This problem could be solved via virtualization i.e running a whole dedicated OS for a service but that's a huge overhead to bear.
- Container virtualization is a server virtualization method in which the kernel of an operating system provides multiple isolated user-space instances independent of each Other.
- Containers look and feel like a real server from the point of view of its owners and users, but they share the same operating system kernel.
- This isolation enables the operating system to run multiple processes, where each process shares nothing but the kernel.
- Benefits :
  - **Isolation** : application and their components can be encapsulated in containers, each operating independently and isolated from each other thereby resolving the dependency conflicts if any.
  - **Portability** : since containers typically contain all components the application needs to function, including libraries and patches therefore, containers can be run on any infrastructure that is capable of running containers using the same kernel version.
  - **Easy deployment** : containers allow developers to quickly deploy new software versions, as their containers can be moved from the development environment to the production environment unaltered.

MAINFRAMES

- They are high performance computer made for high-volume, I/O intensive computing.
- Mainframes are highly reliable and can typically run for years without a downtime.

- Much backup components are built in for critical processing, enabling hardware upgrades and repairs while the mainframe is operating without downtime. They are expensive computers, mostly used for administrative processes, optimized for handling high volumes of data.
- They are expensive computers, mostly used for administrative processes and are optimized for handling high volumes of data.
- **ARCHITECTURE** :



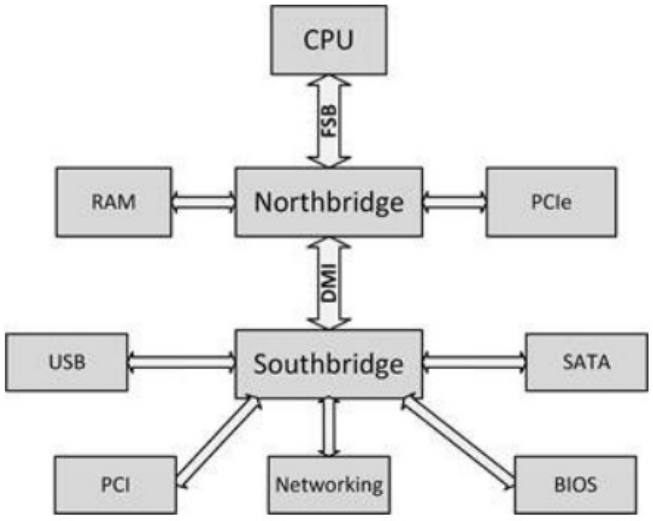
- Made up of :
- Processing Units (PUs),
- Memory, I/O channels
- Control units
- External devices
- **Processing Units** :
  - One question : why not call CPU ?
  - Cause mainframes have multiple Processing Units therefore there's nothing central .
  - Compilation of all the PUs in a mainframe is called a Central Processor Complex (CPC).
  - CPC is placed in its own cage inside the mainframe consisting of $1 - 4$ book packages. Each book package consists of processors, memory, and I/O connections.
  - A book package is something similar to motherboard which is made of :
    - PU
    - RAM
    - IO connections
  - Mainframes use custom made processors which are specially designed for high performance and high reliability computations.
  - They are optimized for handling large scale transactions, parallel processing, and high reliability.
  - While firmware configuration, each PU is assigned a specific task for which they are optimized like database processing, encryption, I/O operations etc.
- **Main Storage** :
  - Each book package contains $4 - 10$ memory cards which are hot swappable i.e they can be removed/replaced without powering off the machine.
- **Channels , ESCONS , FICONS** :
  - A channel is a pathways that provides a data and control path between I/O devices and memory.

- Channels can connect to control units directly or via switches.
- Specific slots in the I/O cages are reserved for specific types of channels, which include the following:
  - **Open Systems Adapter (OSA)**
  - **Fibre Connection (FICON)**
  - **Enterprise Systems Connection (ESCON)**

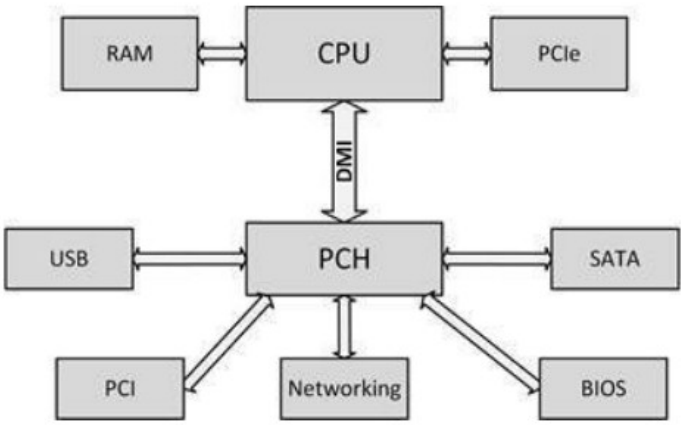
- **Control Units** :
  - They contains logic to work with a particular type of I/O device, like a printer or a tape drive.
  - Some control units can have multiple channel connections providing multiple paths to the control unit and its devices, increasing performance and availability.
  - Control units can be connected to multiple mainframes thereby creating a shared IO system.
- **MAINFRAME VIRTUALIZATION** :
  - the designing of a mainframe is always done with the perspective of virtualization.
  - A single mainframe can run a multiple virtual machines with each virtual machine acting a mainframe in it's own .
  - Mainframes offer Logical Partitions (LPARs) and they act as a separate mainframe where each LPAR run with it's own OS.
  - Furthermore, we can also define the maximum number of processes a LPAR can run.

X 86 Servers

- It's a server architecture.
- **ARCHITECTURE** :
- Consists of a CPU from a X 86 family.
- The X -86 architecture is mainly defined by the X -86 chipset.
- The old X -86 chips used a north-bridge south-bridge architecture.



  - FSB (Front Side Bus) connected CPU to the north-bridge for high speed data transfer.
  - Furthermore the north-bridge connected RAM and PCIe bus.
  - The south-bridge connected the slower components like BIOS,SATA adapters, USB ports, PCI bus.
  - North-bridge and south-bridge are connected by DMI (Direct Media Interface).
- The new X -86 architecture, the north-bridge and south-bridge are replaced by PCH (Platform Controller Hub) architecture.



- The functionality of the south-bridge is managed by the the PCH chip that directly connects to the CPU via the same old DMI.
- The north-bridge functions are now transferred to the CPU.

(Just know the diagrams and you'll be done with it)

SUPER COMPUTERS

- It's architecture was designed to maximise the calculation speed.
- Used for tasks like weather forecasting calculations, oil reservoir simulations, the rendering of animation movies etc.
- Graphics Processing Units (GPUs) are also used together with CPUs to accelerate specific calculations.
- Where a CPU consists of a few cores that are mainly optimized for sequential serial processing, a GPU has mainly a parallel architecture consisting of many of smaller and way more efficient cores designed for handling multiple tasks simultaneously.

COMPUTE AVAILABILITY

- Making Servers highly available can be achieved by the following ways :

1. Hot Swappable Components
2. Parity and ECC memory
3. Lock-stepping

1. **Hot Swappable Components**

- Hot swappable components are the components like memory, CPUs, interface cards, and power supplies that can be installed, replaced, or even upgraded while the server is in a running state.
- What's important here is that the virtualization and operating systems using the server hardware must be aware that components can be swapped while the system is on the run.

2. **Parity and ECC memory**

- Parity bits are used to detect memory errors by adding an extra bit to ensure an even or odd number of ones in a byte.
- If a bit flips, the parity checker detects an error but cannot identify or fix the flipped bit.
- This identification and fixing of the flipped bit is done by ECC (**Error Correction Code**).
- These ECC memory chips use Hamming Code or Triple Modular Redundancy (TMR) for detecting and correcting the flipped bit.
- Hamming code can correct single bit errors occurring in data whereas TMR memory is able to repair two failing bits.
- Multi-bit errors in the same memory location are extremely rare and don't pose much of a threat to memory systems.

3. **Lock-Stepping**

- It's is an error detection and correction technology for servers.
- In this correction method, multiple systems perform the same calculations and compare results.

- If the results match, the calculations are correct.
- If they differ, an error is detected.
- With two systems, errors can be identified but not corrected, while using more systems allows for correction through a voting mechanism.
- This technique ensures high reliability but is costly and hence is mainly used in critical systems.
- Lockstepping works by synchronizing systems per atomic transaction, comparing results at each step.

COMPUTE PERFORMANCE

(Basically discussing techniques for boosting performance) But before jumping into all that , one important thing : **MOORE'S LAW** : It states that the number of transistors on a chip doubles approximately every two years, leading to exponential growth in computing power.

- INCREASING CPU POWER
  - We can modify the following metrics for speeding up computations done via CPU :
    - increasing the clock speed
    - caching
    - prefetching
    - branch prediction
    - pipelines
    - use of multiple cores

1. **INCREASING CLOCK SPEED** :

- CPU clock speed is measured in Hertz (Hz) – clock ticks or cycles per second.
- The CPU processes instructions stored in memory. These instructions need to be fetched, decoded, executed and the result must often be written back to memory.
- Each step in the sequence is executed when an external clock signal is changed from 0 to 1 (the cock tick). The clock signal is supplied to the CPU by an external oscillator.
- Depending on the type of CPU instruction, one or more clock ticks are needed to execute the instruction.
- Thereby increasing the clock speed, more number of instructions can be executed in a given unit time.
- The oscillator speed is used for other parts of the system board. This speed is known as the Front Side Bus (FSB) speed.
- A multiplier is a clock multiplier circuit that maintains the ratio of the oscillator speed to the clock speed.
- A higher multiplier means higher clock speed for a CPU.
- Note that the multiplier only affects the CPU and not the FSB.
- Therefore we can say that increasing the oscillator frequency affects both the CPU and the FSB Speed but the multiplier only affects the CPU's speed.

2. **CPU CACHING** :

- We have SRAM and VRAM .
- SRAM (static) is fast but expensive.
- DRAM (dynamic) is cheap but slow. (Used in main memory)
- Solution ?
- Hybrid solution : Put caches of VRAM on SRAM.
- The cache temporarily stores frequently used data from slower RAM to speed up access.
- This avoids the slow fetch process from main memory, improving CPU performance.
- Main memory RAM runs Slower than the Cache memory.
- Most CPU contains 2 levels of cache :
- Level 1 cache and Level 2 cache.
- Level 1 cache is faster and smaller.
- If we want to run the CPU with 0 latency, we keep Level 1 cache close to the CPU core as compared to Level 2 cache.
- The Level 1 cache gets it's data from Level 2 cache which in-turn gets it's data form RAM.
- Level 1 Cache is split into 2 parts Read and Write whereas Level 2 is both Read and Write in the same chip.
- Multi-core CPU have a 3^{rd} type of Cache which is common to all the cores.

3. **PIPELINES**

- Pipeline allows multiple instructions to be processed simultaneously at different stages.
- While one instruction is being executed, the next instruction can be fetched, and another can be decoded, creating an overlap and increasing processing speed.
- The width of the pipeline is the same as the number of instruction stages.
- Ideally this leads to one instruction being executed per available clock tick.
- But because some instructions need the output from the previous instruction as input, therefore, these instructions are held/kept waiting until the previous instruction has completed its current stage.

4. **PREFETCHING AND BRANCH PREDICTION**

- Whenever the CPU needs data for performing computations, it first searched it's level 1 & 2 caches for finding it.
- In case it's not found then it asks RAM to fetch the data for him.
- This is slow .
- Here Prefetching comes into picture : when the CPU fetches an instruction from RAM, the system also loads the next expected instructions into the cache.
- This helps ensure the CPU gets data quickly without waiting for RAM access.
- But at times there are JUMP instruction in the flow thereby missing out data from the cache.
- To reduce this , the CPU predicts when it's going to get a JUMP instruction and preloads that instruction into the cache before hand.

5. **SUPERSCALAR CPU**

- These CPUs can process multiple instructions per clock .
- This is achieved by sending multiple instructions to redundant functional units within a single CPU core.
- These functional units include components like the Arithmetic Logic Unit (ALU), bit shifters, and multipliers, which execute instructions in parallel through multiple data paths.
- The CPU must analyse data dependencies between instructions.
- A dispatcher circuit reads instructions from memory, determines which ones can run simultaneously, and assigns them to different execution units.
- This parallel execution greatly improves performance but also makes the CPU's internal logic significantly more complex.

6. **MULTICORE CPU**

- High clock speed results into the heating up of the CPU thereby limiting the compute speed.
- A multi-core processor is a CPU with multiple separate cores, each with their own cache.
- It is the equivalent of getting multiple processors in one package.
- If these cores were placed on a single chip without any modification, the chip would consume twice as much power and generate a large amount of heat.
- To solve this, the cores in a multi-core CPU run at a lower frequency to reduce power consumption.
- This leads to reduced load on a single Core and thereby allowing parallel computation to be performed at a CPU level.

7. **HYPER THREADING**

- It allows a single physical CPU core to function as two logical processors, enabling it to run two threads simultaneously.
- This improves efficiency by keeping the processor pipelines busier, though it does not double performance like adding real cores.
- Hyper-threading requires both CPU and BIOS support to function.

COMPUTE SECURITY

1. **PHYSICAL SECURITY** :

- Servers should have USB ports disabled in BIOS to prevent unauthorized access.
- BIOS must be password Protected.

2. **VIRTUALIZATION SECURITY** :

- Use of Virtualization introduces security risks in it's own ways.
- Firewalls and Intrusion Detection Systems (IDSs) in the hypervisor should be deployed. (hypervisor is a software that allows multiple Virtual machines to run on a single device)

3. **DMZ SECURITY** :

- Use separate physical machines for servers inside the De-Militarized-Zones to ensures better isolation and security from external threats, even when running virtual machines.

4. **SYSTEM MANAGEMENT CONSOLE** :

- This console manages all the virtual machines running.
- This console should have Strict access control, separation of duties, and logging user activities to provide a secured physical machine are essential for protection.

OPERATING SYSTEMS

POPULAR OS

- zOS
- Linux
- Windows
- IBM i(OS/400)
- Open VMs
- UNIX
- BSD
- OpenBSD

OS AVAILABILITY

- To make OS available all times, Failover Clustering is used.

FAILOVER CLUSTERING

- A failover cluster is a group of independent servers (nodes) that work together to ensure high availability of applications.
- Each server runs an identical operating system and is connected to other nodes via a network.
- The cluster is managed by cluster software, which detects failures and automatically shifts workloads from a failed node to another available node.
- **How does this work ?**
- Failover Clustering groups related application components into a resource pool (also called an **application package**). This package contains all necessary scripts and configurations required to **start, stop, monitor, and migrate** an application between cluster nodes.
- Each resource pool consists of:
  - **Application Name & Identifier** – Unique identifier for the application.
  - **Start Script** – Commands to start the application.
  - **Stop Script** – Commands to stop the application.
  - **Monitor Script** – Continuously checks application status and initiates a restart or failover if it stops responding.
  - **Virtual IP Address** – Ensures applications remain accessible after failover.
  - **Storage Mount Points** – Defines which disks must be available for the application.
- If a node fails, the cluster software automatically moves the resource pool to another node and remounts storage, ensuring continuous availability.
- Failover cluster relies on a redundant network of physical Ethernet connections for communication between nodes. This network is mainly used for:
  - Heartbeats – Small packets sent between nodes to check their availability.
  - Membership Updates – Nodes inform each other of their operational status.

    - State Change Notifications – When a failure occurs, nodes receive notifications to take appropriate action.
- If a node stops producing heartbeats, the clustering software assumes the node to have failed and triggers a failover process .
- Most of the failover clusters use shared memory. This means that :
    - Every active application has exclusive access to a disk. (though only one node is allowed to mount and use that particular disk.)
    - When an application fails over to another node, the same data remains accessible.
    - The cluster software automatically remounts the correct storage to the new active node.
- Thought most use shared memory , but essentially there are 2 types of Cluster Storage :
    - **Shared-Nothing Clustering**:
        - Only one node accesses a given disk at any time.
        - After a failover, the disk is remounted on the new active node.
        - This approach prevents data conflicts.
    - **Distributed Lock Management (DLM) Clustering**
        - Multiple nodes can simultaneously access the same storage.
        - A lock management system ensures data integrity.
        - Used in advanced clustering setups requiring concurrent access.
- Every active application in a failover cluster has a standby counterpart on a passive node.
- The passive node remains idle until needed. If an active node crashes due to any reason, the passive node takes over immediately.
- Basically it's a fallback node which remains inactive and becomes active only when the main node fails .
- When applications restart on another node, they go through standard crash recovery:
    - The file system checks and repairs corrupted data before re-mounting on the memory.
    - The application itself performs it's necessary recovery procedures implemented internally.
- **TYPES OF CLUSTERING ARCHITECTURE** :
    i. $N + 1$ **Clustering** :
        - N nodes actively run applications.
        - 1 spare node remains idle until a failure occurs.
    ii. $N + 2$ **or** $N + 3$ **Clustering**:
        - Multiple spare nodes for increased redundancy.
        - Example: 4 active nodes and 2 spares ($4 + 2$ setup).
    iii. **N to N Clustering**:
        - No dedicated spare nodes.
        - Every node keeps some spare capacity to handle failovers.
        - More efficient than N+1 because all resources are used.
- **SPLIT BRAIN PROBLEM** :
- Suppose we have even number of clusters get split into 2 equal halves and loose connection between them.
- So each half in itself is working but isn't aware of the other half's status.
- Due to this 2 scenarios may occur :
    i. The first half assumes the other half is failed.
        - A failover condition is triggered and one half tried to take up the workload of the other half (which in reality is still running)
        - This is fatal as this may lead to data-overwriting and data-corruption.
    ii. Both the halves assume that they lost the cluster.
        - In this scenario, all the nodes shut-down
        - This causes unnecessary downtime.
- To Solve this problem we use the concept of Quorum Disk.
- A quorum disk is a shared disk that acts as a virtual third node in the cluster.
- The quorum disk belongs to only one node at a time.
- If a node fails or loses connection, it releases the quorum disk.
- The surviving node gets control of the quorum disk and wins the majority vote.

OS SECURITY

- The following measures can be taken for ensuring the safety and security of OS :

1. **PATCHING** :

- Operating system vendors provide patches to fix bugs, close security holes, or improve functionality.
- Patches come in three types:
  - **regular patches** for low-priority fixes,
  - **hot-fixes** for urgent security flaws, and service packs, which bundle multiple updates.
- While applying patches promptly is recommended, they should be tested before deployment to avoid potential issues that may affect the functionality of the OS.

2. **HARDENING** :

- It's a process in which an operating system disables all unnecessary services, removes unused accounts, and apply security patches.
- A standardized configuration template ensures that all systems maintain consistent security levels throughout the infrastructure.

3. **VIRUS SCANNING** : -Installing virus scanners on vulnerable operating systems that helps them protect against malware and other sorts of threats.

- To minimize performance impact while scanning, virus scanners are configured in such a way that they focus on high-risk files and directories based on some risk assessment parameters.

4. **HOST BASED FIREWALLS** :

- Host-based firewalls are built into almost all operating systems providing an extra security layer by blocking unwanted network traffic.
- They use rule sets to allow or deny communication based on source/destination IPs, ports, and processes.
- They block the suspicious and malicious network connection requests and packets.

5. **LIMITING USER ACCOUNTS**

- Default user accounts should be removed or should have their passwords changed to prevent unauthorized access.
- Super user accounts (root, administrator etc) should only be used for assigning permissions, with secure passwords stored safely.
- Encrypted passwords should never be disclosed to avoid brute force attacks.

IMPROVING PERFORMANCE OF OS

1. **INCREASING MEMORY**:

- OS needs memory for a smooth functioning.
- When an application needs more memory, it can simply do it in 2 ways :
  i. By sending less used memory pages to disks (Paging)
  ii. by moving an entire application's allocated memory to disk (swapping)
- Unused memory is utilized for disk caching and improving performance.
- More memory allows better disk caching thereby reducing disk read times.

2. **DECREASING KERNEL SIZE** :

- Some OS allow disabling unused kernel features to reduce kernel size.
- A smaller kernel improves efficiency and security.
- Disabling unused kernel features simplifies it thereby reducing the chances of crashing.
- Since kernel is always to be present in the memory, therefore a small kernel will consume less memory and therefore increases the performance.
- The Boot-time also reduces upon downsizing the kernel.
- To create a smaller kernel, the kernel must be recompiled or re-linked.

UNIT 4

What's a Datacentre ?

- A place where most of the IT hardware is placed .

TYPES OF DATACENTRES :

- **Sub Equipment Room (SER)** :
  - AKA Patch Closet
  - Contains patch panels (it's a large hardware assembly used to manage network cable connections in a structured way.) and some small equipment.
- **Main Equipment Room (MER)** :
  - A small datacentre in the organisation's building.
- **Organization owned datacentre** :
  - An organization's own datacentre
  - Often located in multiple placed with fallback and fall-over abilities.
- **Multi-tenant datacentre** :
  - A datacentre owned by a service provider
  - Generally used by multiple organizations.

Datacentre Location

- You checkout a hell lot of things before setting up your datacentre somewhere
  - Datacentre should be big enough to allow future expansion.
  - The location shouldn't be vulnerable to flooding,hurricanes,earthquakes etc.
  - Shouldn't be near a dump-yard or a chemical factory
  - Shouldn't be in a politically unstable area.
  - The power supply should be available throughout.
  - The location of the datacentre shouldn't be visible on a public map
  - ... and the list goes on

VIRTUALIZATION

- Here we add an additional layer between OS & hardware.
- This additional layer helps in running multiple OS onto one single Device.
- This helps to modify the virtual machine as per our need without an Upfront hardware.
- By consolidating many Physical computers as Virtual Machines of Fewer (but bigger and powerful hardware) a lot of power and hardware cost can be saved .

Software Defined Compute :

- It's basically the use of software to manage the compute resource distribution instead of relying on hardware separation.
- All physical machines run a hypervisor & all hypervisors are managed using one layer that uses the software to manager resource allocation.
- Using the hypervisor , all the CPU , memory , disk & networking resources of the physical device can be dynamically allocated to the virtual machine which needs it the most.
- Many platforms allows the movement of virtual machines hassle-free form one physical machine to another.
- This not only helps to shut down the original machine for maintenance but also keeps the virtual machines running without interruption.
- In case of failure of the physical machine, the virtual machines can resume their state on the new physical devices .
- There is also the provision of Lockstepping where the states of 2 virtual machines is in sync on the memory level so that if one Virtual machine fails , the other can takeover instantly .

DISADVANTAGES OF VIRTUALIZATION :

- Due to the ease of creation of Virtual machines, they are created for every sort of task which otherwise could be done in an efficient manner thereby adding a constrain on the available resources.
- Some systems require additional hardware and settings to be able to run virtual machines.

Virtualization Technologies :

1. EMULATOR :

- A software that allows you to run applications on the devices that weren't meant to be run on the user device.
- It's done by the process of reproducing the behaviour of the intended device through a process of Translation where the Emulator translates the CPU instructions into the intended device's instruction.
- Apart from CPU instructions, Emulator is also supposed to emulate the behaviour of various other components like network adapters , keyboards, disk access etc.

2. LOGICAL PARTITIONS (LPARs):

- An LPAR is a hardware-assisted virtualization technique that divides a physical computer system into multiple independent, isolated virtual systems.
- Each LPAR is bounded to one or more compute resource like CPU , NIC etc.

3. HYPERVISER :

- It acts as a resource manager for virtual machines needing compute resources on a device.
- They control the computer's physical hardware system including BIOS , Virtual memory and virtual devices.
- IMPLEMENTATION :
  - **BINARY TRANSLATION** :
    - Some CPU instructions needs to be modified before sending them for processing.
    - Because binary translation is performed on the binary code that gets executed on the processor, it does not require changes to the operating system running in the virtual machine.
  - **PARAVIRTUALIZATION** :
    - AKA OS Assisted Virtualization
    - Here the virtual OS asks the Hypervisor to allow it for executing some privileged commands.
    - This greatly reduces the overall load on the hypervisor.
  - **HARDWARE ASSISTED VIRTUALIZATION** :
    - Here certain instruction sets called Virtual Machine Extensions (VMX) are pre-implemented for supporting virtualization at hardware level.
    - This eliminated the need of Paravirtualization and Binary Translation.
    - VMX enabled processors allow trapping of sensitive instructions from virtual machines and handling them safely

VIRTUAL MEMORY MANAGMEMT :

1. MEMORY OVERCOMMIT :

- This is based on the assumption that all the assigned memory is not utilized by the application it's allotted to.
- Here the Hypervisor Commits more memory to the combined virtual machines that actually available in the hardware.
- Hypervisors can identify idle memory and then re-allocate it to the application in need of more resources .
- This can turn into a disaster while booting the physical machine.
- While booting, all the virtual machines may get loaded demanding memory that has been committed to them but isn't physically available on the device.

2. MEMORY SHARING :

- The whole idea is to map the pages containing identical memory content to one single memory page containing the same memory content.
- The hypervisor scans for memory pages for identical memory content and then remaps the Virtual machine's page to read-only shared copy of the page.
- In case of a write attempt on a shared page, the hypervisor allocates a new copy of the page with read-write permissions for that virtual machine.
- This reduces the memory requirements of the overall system drastically.

VULNERABLITY PATCHING :

- It includes the applying updated issued by the vendor to reduce security vulnerabilities , bugs and other software optimizations .
- Patch Updates include :
  - Security Updates
  - Bug Fixes
  - Downtime minimization
  - Regulatory Compliance Updates

WEB 2.0

- It's a term used for the second generation of internet where the content is more of user generated in contrast to the older version where the websites were static.
- Example : Social Media . It's a place where we can produce content like text post,videos, images etc.
- This not only allows the users to upload information but to interact with it and connect & collaborate with other fellow users.

WEB 3.0

- Here the technologies and concepts under focus are :
  - Blockchain
  - AI
  - Semantic Understanding
  - Token Based Economies
- It's aimed at making the internet space more intelligent , secure and user-empowering by removing the central authorities that control the web (like google, meta etc.) and giving back the control to the users.

UNIT 5

RISK MANAGEMENT

- Risk management is necessary because of the levels of efforts put into building of these large scale systems.
- A risk list can be used to quantify the risks. This list contains :
  - Asset Name : the component that needs protection
  - Vulnerability : a weakness or some sort of vulnerability that may lead to an attack.
  - Exploit : The way one can use the vulnerability.
  - Probability : The probability of such an event happening .
  - Impact : The severity of the damage one can cause exploiting the vulnerability. Like :
    - 4: Catastrophic: Complete mission failure, death, bankruptcy
    - 3: Critical: Major mission degradation, major system damage, exposure of sensitive data
    - 2: Moderate: Minor mission degradation, minor system damage, exposure of data
    - 1: Negligible: Some mission degradation
  - $\mathbf{Risk = Probability \times Impact}$

RISK RESPONSE :

- For every type of risk these are the possible responses one can respond with :

1. ACCEPTANCE : Simply accept the risk if the cost associated with the risk is not too high.
2. AVOIDANCE : Simple don't do something that leads to a risky state.
3. TRANSFER : Transfer the risk to an Insurance Company.
4. MITIGATION : A process where a plan is developed to manage & eliminate the risks or their setback as much as possible. This includes:
    - Designing systems with the target of minimizing risk.
    - Incorporating safety devices for preventing any kinds of exploit attempts.
    - Provision of warning devices which can alert and make people aware of the critical condition of the system and needs immediate attention.
    - Implementation of Training procedures for making the staff able enough to handle situations of emergency.

EXPLOITS

- Key-loggers can be installed to devices for sending malicious data to third party.
- Network sniffers can show packets travelling in the network and can read them to read sensitive information.
- Data inside backup drives may get stolen.
- Disposed off disks may land into wrong hands.

SECURITY CONTROLS

- CIA Triad :
    - Confidentiality : Prevents unintentional access
    - Integrity : Unauthorized modification aren't done to the data and it's consistent.
    - Availability : Ensures timely access of data to authorized personnel.
- There are certain levels for each of the CIA requirements :

Confidentiality Level	Description
1	Public information
2	Information for internal use only
3	Information for internal use by restricted group
4	Secret: reputational damage if information is made public
5	Top secret: damage to organization or society if made public

Integrity Level	Description
1	Integrity of information is of no importance
2	Errors in information are allowed
3	Only incidental errors in information are allowed
4	No errors are allowed, leads to reputational damage
5	No errors are allowed, leads to damage to organization or society

Availability Level	Description
1	No requirements on availability

Availability Level	Description
2	Some unavailability is allowed during office hours
3	Some unavailability is allowed only outside of office hours
4	No unavailability is allowed, $24/7/365$ availability, risk for reputational damage
5	No unavailability is allowed, risk for damage to organization or society

ATTACK VECTORS

1. MALICIOUS CODE :

- These are the applications which if run may lead to network stress , data sharing to unauthorized entities etc.
- There are different forms :
  - Worms
  - Virus
  - Trojan

2. DENIAL OF SERVICE ATTACK :

- Denial of Service (DoS) attack is an attempt to overwhelm a system with excessive requests, causing service disruption or downtime. It typically targets public-facing servers like web servers using a flood of (sometimes malformed) requests, overloading system resources or crashing the server.
- Prevention :
  - Separate internal business services from public ones.
  - Host public services on cloud platforms.
  - Use auto-scaling and virtualization to absorb traffic spikes.
  - Limit bandwidth or request rates on vulnerable ports (e.g., DNS, NTP).
  - Set low DNS TTL to enable fast traffic redirection.
  - Monitor traffic volume, request sources, and latency for early detection.
- Mitigation :
  - Informing the internet service provider (ISP).
  - Blocking IPs generating excessive requests.
  - Switching to alternate servers/IPs.
  - Scaling the infrastructure under attack.
  - Using Content Delivery Networks (CDNs) to distribute and filter traffic globally.

3. **PHISHING** :

- An email is sent to the target pretending to be their Bank (typically) and lures the target into sharing credentials for their sensitive information like Credit Card PIN, OTP etc.

4. **SOCIAL ENGINEERING** :

- Here we simply use our social skills to manipulate users into sharing their details/credentials or granting access to their device.

5. **BAITING** :

- Here the attacker baits the user into connecting a USB Flash Drive with their device. The user's curiosity to know what's inside the flash drive is exploited and then the user data

MISSED TOPICS :

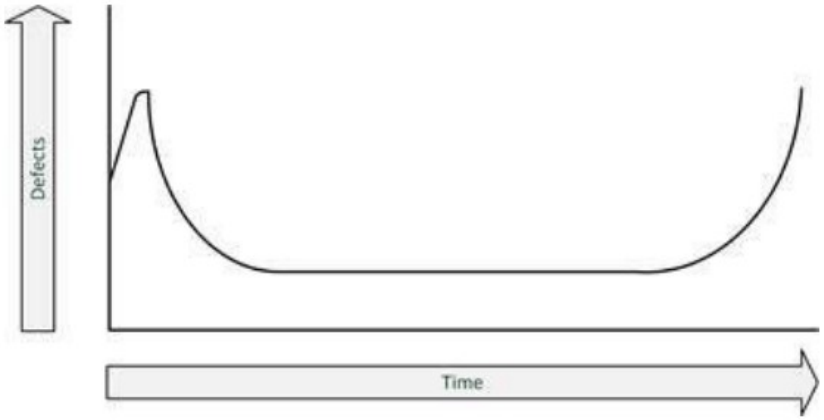
1. Green Computing : SOURCE FOR THEORY

- Green computing, or sustainable IT, focuses on minimizing the environmental impact of technology through eco-friendly design, manufacturing, usage, and disposal
- This includes reducing carbon emissions, energy consumption, electronic waste, and promoting renewable resources.
- All the services around IT Sector contributes a lot when it comes to global greenhouse gas emissions, with data centres alone capturing a big chunk of global energy.
- Contribution of stakeholders can certainly make things better.
- **Manufacturers** :
  - Innovate energy-efficient hardware.
  - Use non-toxic, sustainably sourced materials and reduce manufacturing waste.
  - Extend product lifespans, enhance reusability, and ensure recyclability.
- **Organizations** :
  - Optimize data centres via hot/cold aisle layouts, automated cooling, and power management.
  - Procure energy-efficient devices and schedule resource-intensive tasks to minimize energy use.
- **Individual Level** :
  - Utilize power-saving modes (sleep/hibernate), adjust screen brightness, and turn off devices when unused.
  - Refill printer cartridges, buy refurbished electronics
  - Recycle e-waste responsibly.
  - Prioritize Energy Star-certified products for efficiency.
- **Challenges**
  - **Awareness** about IT sector's climate impact is often overlooked, with market priorities favouring speed and size over sustainability.
  - **Costs** of Transitioning to green infrastructure as it requires significant upfront investment.
  - **Conflicting Priorities** where users balance environmental goals with needs like security (e.g., data centres) or portability (e.g., students opting for lightweight devices).
  - **Rapid Innovation** leads to frequent tech upgrades complicate efforts to extend product lifecycles and maintain eco-standards.

SOURCES OF UNAVAILABILITY

- **Human Errors**
  - **User Misuse**: Overloading systems (e.g., generating multiple large reports) or accidental lockouts (e.g., repeated password failures).
  - Administrative Mistakes: Switching off the wrong server, restoring incorrect backups, mislabelling cables, or typos in commands (e.g., `sudo rm -rf / *.back` vs. `sudo rm -rf /*.back`).
  - **Testing Gaps**: Untested failover procedures or maintenance in production environments.
  - **Mitigation**:
    - Standardized procedures
    - Restricted administrative access
    - Automated deployment tools
    - User awareness (e.g., UNIX login warnings).
- Software Bugs :
  - Bugs in applications, drivers, or operating systems (e.g., Windows' Blue Screen of Death) can crash systems, corrupt files, or disrupt networks.
- Physical Defects :
  - Common Failures like Fans (dust-clogged bearings), disk drives (motor wear), tapes/robots (mechanical stress), and aging capacitors/batteries.
  - Environmental Factors like Temperature fluctuations, moisture, vibrations, and frequent power cycling accelerate degradation.

BATHTHUB CURVE (COMES UNDER PHYSCIAL DEFECTS)



- 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.
- Environmental Effect (Earthquakes, Fire accidents etc.) can also lead to unforeseen scenarios of unavailability.

IDENTITY & ACCESS MANAGEMENT :

- It's all about having control on who can access the system be it people or other systems .
- It's done in 3 steps :
  - Identify : Provide your identity as in Who are you. (Like a username for a login)
  - Authenticate : Provide the proof of who are you (like a password for a login)
  - Authorize : If you're authentic then you're allowed to do whatever you're provided access for .
- **Trusted Computing Base (TCB)**
  - It's a set of components in a computer system that is critical for enforcing security.
  - It includes :
    - The kernel
    - Security policies.
    - User Authentication Mechanisms
    - Access Control Logic
    - System hardware that enforces security
  - Any corruption in the TCB File system or program can lead to the whole system getting compromised.
- **Single Sign On (SSO)** :
  - **Single Sign-On (SSO)** is an authentication process where a user logs in once, and gains access to multiple systems or applications without having to log in again for each of them.
  - **TLDR**, Basically Single login and you get access to pretty much everything you're entitled to
  - **Working** :
    - When you log in via SSO, you authenticate once with a central **Identity Provider (IdP)**.
    - The IdP issues a token (like a signed cookie or encrypted ticket).
    - All connected systems trust this token, and accept it instead of asking you to log in again.
  - **Risks** :
    - **Single Point of Failure**: If SSO is compromised, all apps are vulnerable.
    - **Session Hijacking** : If the session token is stolen, attacker gets full access.
    - **Phishing**: A fake SSO login page can trick users into revealing credentials.
- **Federated Identity Management** :
  - It's Like SSO extended across multiple organizations.

    - It allows a user from Organization A to access systems in Organization B, without needing separate credentials.
    - Organizations establish a trust relationship, and agree to use a common identity provider (or trust each other's identity providers).
- In IAM processes, there are mainly 3 ways via which we authenticate users :
    - **SOMETHING YOU KNOW** : It can be a Pin or a Password.
    - **SOMETHING YOU HAVE** : Like a key, or a device or a token.
    - **SOMEONE YOU ARE** : Like your biometric information like Fingerprint scan, Iris Scan etc.
- Multi Factor Authentication needs any sort of combination of the above 3 types for getting access.
- Ex : Accessing your account via credit card needs your **credit card (SOMETHING YOU HAVE) and it's PIN (SOMETHING YOU KNOW)**
- **ROLE BASED ACCESS MANAGEMENT**
    - Here permissions are granted to certain roles and those roles are then assigned to the user.
    - This creates a many-to-many mapping:
        - A role can be assigned to many users.
        - A user can have many roles.
        - A role can have many permissions.

Entity	Meaning	Example
Users (U)	People or systems that use the system	Alice, Bob, SystemX
Roles (R)	Named job functions or responsibilities	HR_Manager, Developer, Auditor
Permissions (P)	Approval to perform an operation on a resource	Read Payroll, Write Logs
Sessions (S)	Mapping of active roles for a user	Alice logs in with "Auditor" only

- **SEGREGATION OF DUTIES AND LEAST PRIVILEGES** :
    - Segregation of Duties (also called **Separation of Duties**) is a security principle that splits sensitive responsibilities across multiple people or departments.
    - This is based on the philosophy of

> **No one should have all the power**

- This was needed to avoid the complete control of the systems to one single person which if done can lead to that single user : - Make unauthorized changes, - Cover their tracks, - Or cause massive damage — without checks and balances.

- **Principle of Least Privilege (PoLP)** :
    - Here the core idea is that :

> **A user/system should have only the minimum access required to perform their job — no more, no less.**

    - The table below sums up the PoLP :

Feature	Description
Minimal Access	Only what is needed (e.g., read-only instead of full control)
Time-Bound	Access should be temporary, if possible
Purpose-Limited	Access should be strictly for the task

    - **Two Men Rule** :
        - Two people must approve or execute a sensitive action.
        - Two systems managers must review and approve each other's work.
        - The purpose of two-man control is to minimize fraud or mistakes in highly sensitive or high-risk transactions.
- **Layered Security** :
    - aka **Defence-In-Depth strategy**.
    - It involves using multiple, independent security mechanisms to protect your system where each mechanism acts like an entry barrier.
    - Benefits :

        - Multiple layers armed with IDS (Intruder Detection Systems) gives the system multiple chances detecting and stopping the attacker/intruder.
        - Each layer requires different skills and information to break through.
        - Attackers/intruders don't know how many defences are there and of what kind.
    - Drawbacks :
        - Each layer must be configured, updated, and monitored.
        - More layers $\implies$ more administrative overhead.

GO LIVE SCENARIOS FOR PUTTING UP NEW INFRASTRUCTURE TO USE

- There are multiple scenarios while replacing the infrastructure for existing systems under use :

1. **BIG CHANGE OVER**
    - Entire switch happens at a specific point in time — old system off, new system on.
    - Requires short data migration, if any, just before go-live.
    - Highest risk scenario — rollback is nearly impossible after go-live.
    - High potential for downtime during switchover if issues occur.
    - Simple to execute due to clear-cut transition.
    - Risk is focused entirely on the switchover moment.
    - No fallback once the `point of no return` is crossed.
2. **Parallel Changeover**
    - Both old and new systems run together for a period (usually weeks).
    - Allows testing of the new system with live production data.
    - Enables rollback to the old system if some issues arise.
    - Reduces risk via real-time validation of new functionality & features.
    - It's a costly affair because of running and maintaining two systems simultaneously.
    - More effort are needed to keep both systems synchronized.
    - Useful for validating non-functional requirements like performance.
    - Requires duplication of interfaces and resources.
3. **Phased Changeover**
    - Gradual transition takes place as new system takes over in parts (modules/functions).
    - Controlled and incremental (means risk is distributed over time).
    - It requires interfacing between old and new systems during transition.
    - Interfaces add complexity and risk due to testing and integration challenges.
    - Old system must remain live until final component is migrated.
    - It leads to high costs due to prolonged dual-system support.
    - It reduces chance of complete failure drastically by isolating issues to individual phases.
    - Allows user adaptation in steps, improving change management.
    - Planning and sequencing are critical for success.
    - The whole transition process may become complex if dependencies between modules are tight/complex to maintain.

- In general , Go Live scenarios must have these points taken care of be it any scenario :
    - Go-live must be backed by a detailed, step-by-step plan.
    - Plan should include testing, improvement cycles, and **go/no-go** checkpoints.
    - Define a clear **point of no return** i.e a point after which rollback is not possible
    - On-site support is necessary post-go-live to handle immediate issues.
    - Documenting all steps and responsibilities is very essential

MANAGING THE INFRASTRUCTURE

- **Monitoring** :
    - This involves continuous inspection of the Components for events like error conditions or signs of failures.

- Monitoring systems like Nagios, Zabbix, HP Operations Manager, and BMC Patrol provide dashboards with overviews of an entire infrastructure landscape.
- Monitoring systems can have alarms configured that trigger if a certain threshold is reached.
- The monitoring system can forward these alarms to systems managers who can take action to fix the event that led to the alarm (preferably before the end users notice anything unusual).

- **Management Using SNMP (Simple Network Management Protocol)**
  - SNMP is a protocol used to remotely monitor and manage network-connected devices like:
    - Routers
    - Switches
    - Servers
    - Workstations
    - Printers
    - Power strips
  - The core components are :

Component	Role
Agent	Installed on each managed device. Knows what's happening locally (e.g., CPU usage, memory, config changes).
NMS (Network Management System)	Central system that **collects data from all agents**, displays info to administrators, and sends commands.

  - The agent on the monitored device communicates with the management server (the Network Management System NMS) that collects information from all attached devices.
  - An agent has local knowledge of the system it resides on, and translates that information to the SNMP protocol.
  - The NMS monitors and controls managed devices via the agents.
  - SNMP protocol allows reading counters and statistics over the network to an NMS, which in turn show them to systems managers using values or graphs.
  - This reading of values is done in regular polling intervals (like every 30 seconds).
  - **Traps (Alarms)**
    - If something critical happens (e.g., network load exceeds 80%), the agent pushes an alert to the NMS immediately.
    - This helps react fast to issues without waiting for the next polling cycle.
  - **Security** :
    - It's implemented using a shared secret string (called the community name) which provides access to agent functionality.
    - By default, the SNMP community strings are set to public and private for reading and writing configurations respectively.
  - **Logging** :
    - Logging is the process of recording everything important that happens on a device or in a system.
    - What Devices Generate Logs?
      - Operating systems (e.g., Linux, Windows)
      - Firewalls
      - Routers & switches
      - Applications
      - Web servers
      - Databases
      - Intrusion detection systems (IDS)
    - Logs can be used to :
      - Correlate events and identify sources of application issue.
      - Identify trends to predict or even prevent unavailability.
      - Find security vulnerabilities or security breaches.
    - Logging often generates large amounts of data every day
    - The level of logging (i.e the detail the ca be captured) is configurable.
    - Log analysis is performed for the following reasons:

- Compliance with security policies, law, or regulation
- System troubleshooting
- Forensics
- Security incident response

Monitoring (e.g., SNMP)	Logging
Real-time alerts	Post-incident analysis
Triggered by thresholds	Detailed event records
Focus on current health	Focus on **history** and **context**
Goal: respond fast	Goal: understand fully

Decommissioning IT Infrastructure

- When infrastructure (like servers, network devices, storage systems, etc.) reaches the end of its useful life or is no longer needed, it must be properly decommissioned to avoid the following possible issues :
  - Security risks
  - Compliance violations
  - Operational disruption

PREPARATION PHASE

Step	Explanation
1. Create a Plan	Interview system specialists. Define steps and pick a **planned date**.
2. Communicate Early	Inform users and teams **well in advance**. Avoid surprises.
3. Check Dependencies	Ensure **no other systems rely** on this one (use dependency maps, app logs, or **firewall logs** to trace access).
4. Plan for Data Retention	Ask: Do we need backups for compliance (e.g., tax or legal records)? If yes, **archive securely**.
5. Verify It's Unused	Use monitoring or **firewall traffic logs** to confirm no system is actively connecting to it.
6. Ask for Vendor Support	Some hardware/software vendors might need to help with shutdown, license release, or data wiping.
7. Inform the Data Centre Manager	Required for coordination during hardware shutdown or removal.

EXECUTION PHASE

Step	Explanation
1. Final Backup	Create one last snapshot/backup just in case rollback is needed.
2. Remove from Monitoring	Delete entries from systems like Zabbix, Nagios, Prometheus, etc.
3. Remove from Backup Schedule	Ensure it's no longer included in regular backup jobs.
4. Cut Network Access	Disable **firewall rules**, unplug network cables, or remove VLAN assignments.
5. Power Down the System	Turn off machine — but stay ready to power it back on if needed.
6. Physically Remove Hardware	Remove from rack space, power strips, etc.
7. Remove Associated Cables	Clean up patch panels, fibre connections, etc., to **free up resources**.

CLEAN-UP PHASE

Step	Explanation
1. Stop Paying Licenses	Cancel software licenses or cloud subscriptions.
2. Clean Firewall Rules	Remove security rules that referred to the decommissioned IP.
3. Delete Installation Media	If not reusable, remove from software vault to avoid clutter.
4. Update Documentation	Remove or revise architecture docs, operational manuals, network diagrams.
5. Wipe or Destroy Data Media	Use DoD-level wiping tools, degaussers, or physically destroy disks/tapes.
6. Delete Databases & Tables	Remove schemas/tables used solely by the decommissioned app.
7. Clean Up DNS & IPs	Free up **DNS entries** and **IP addresses** in the IPAM system.
8. Revoke Access	Delete user accounts, roles, or service credentials associated with the system.
9. Inform Finance Team	For **accounting and depreciation tracking** — especially if assets are capitalized.
10. Remove from CMDB	Update the **Configuration Management Database** to reflect this change.

RFID : SOUCRCE

- Radio Frequency Identification (RFID) is a form of wireless communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object or person.
- There are tags attached to objects/persons that contain electronically stored information that can be read from several meters away, without requiring direct line-of-sight.
- RFID is commonly used in :
  - Inventory management
  - Asset tracking
  - Access control
  - Supply chain logistics
- **WORKING** :

COMPONENTS

Component	Function
1. Scanning Antenna	Emits radio signals to detect tags.
2. Transceiver	Receives signals and converts them into digital data.
3. Transponder (in the Tag)	Responds to the antenna’s signal with data.

  - When the scanning antenna and transceiver are combined, they are referred to as an RFID reader or interrogator.
  - There are two types of RFID readers- fixed readers and mobile readers.
  - The RFID reader is a network-connected device that can be portable or permanently attached.
  - It uses radio waves to transmit signals that activate the tag.
  - Once activated, the tag sends a wave back to the antenna, where it is translated into data.
  - The transponder is in the RFID tag itself.
- Types of RFID Tags

1. Passive Tags

- No built-in power source, relying on the RFID reader.
- Less expensive, longer lifespan, shorter read range (up to a few meters).

2. Active Tags

- Have their own power source (battery), allowing for a longer read range (up to hundreds of meters).
- More expensive, limited lifespan due to the battery.

3. Semi-Passive Tags

- Small battery powers the tag's circuitry.
- Middle ground in terms of cost, range, and lifespan.

FEATURES

Feature	Explanation
Durability	Works in harsh environments, unlike barcodes.
Automation	Supports automatic identification, requiring no human line-of-sight.
Real-time Access	Instantly retrieves data from tags.
Bulk Reading	Can read **hundreds of tags** simultaneously.
Large Storage	Some RFID tags can store **significantly more data** than a barcode.
Traceability	Critical for **production lines** and **supply chains**.

Disadvantages along with Solutions

Risk Category	Risk Description	Mitigation Techniques
1. Data Interception	Eavesdropping on RFID communications to steal data	- Use RFID v3 with encryption - AES-based data encryption - Mutual authentication (challenge-response)
2. Signal Jamming	Disruption of communication between tags and readers	- Frequency diversity (use HF/UHF/LF selectively) - RF shielding in sensitive areas - RF interference detection
3. Unauthorized Tracking / Cloning	Cloning of RFID tags or tracking individuals or goods without consent	- Kill command or self-destruct tags - Randomized tag IDs (pseudonyms) - RFID-blocking wallets/covers
4. Tag Programming Complexity	Tags take time and effort to encode accurately	- Batch programming tools - Standardized templates - Use RFID middleware or edge gateways
5. Physical Tampering	Attackers tamper with tags or readers physically	- Tamper-evident tags - Physical security of reader zones - Integrate with security systems (alarms/IDS)
6. Unauthorized Access (Tag Cloning)	Reading/writing data using rogue readers	- Cryptographic tag signatures - Access control via whitelist readers - Regular key rotation
7. Privacy Concerns	Users being tracked or profiled secretly through RFID	- Encrypted memory on tags - Disable tag after use - Notify users of RFID usage & purpose

Risk Category	Risk Description	Mitigation Techniques
8. Interference	Tags malfunction due to metals, liquids, or dense RFID environments	- Use RFID-tag safe materials - Frequency planning - Reader sensitivity adjustment