Sanjid akram N

11 jan 2025

DATA CENTER ASSIGNMENT

Data centre:

A Data Center is a physical facility that houses a large number of computers, servers, storage devices, and networking equipment.

In simpler terms, it's like a "big building" where a company's digital "brain" (computers, databases, etc.) is located, allowing users or businesses to access their data, applications, or websites.

Creating Virtual Machines (VMs) from One Box:

o Different Linux Distributions:

o Linux distributions (distros) are various versions of the Linux operating system tailored for different use cases or preferences. Some common families of Linux distros include:

➢ Ubuntu family: Based on Debian, Ubuntu is user-friendly and often used in desktops and servers.

➢ Red Hat family: Includes distributions like Red Hat Enterprise Linux (RHEL), CentOS, and Fedora, which are used for enterprise servers and systems.

➢ Debian family: Includes Debian, Ubuntu, and others.

➢ Oracle Linux: A distribution of Linux developed by Oracle, often used in enterprise environments, especially with Oracle software.

o APT and APT-get in Ubuntu:

o APT (Advanced Package Tool) and APT-get are tools used in Ubuntu (and other Debian-based distributions) to manage software packages. APT helps you install, upgrade, and remove software from your system.

➢ For example, to install a package, you would run:

sudo apt-get install <package-name>

➢ APT automatically handles dependencies, ensuring that the required libraries or packages are also installed.

o Snapshots of VMs:

o VM Snapshots are used to capture the state of a virtual machine at a specific point in time. This allows you to revert the VM back to this exact state if needed.

➢ Why?: Snapshots are used to preserve the VM's state so that you can:

▪ Restore the machine to a previous configuration if something goes wrong.

▪ Test changes without affecting the running system.

➢ Persistence: Data stored in the hard disk is persistent, meaning it remains intact even after a reboot. However, data stored in RAM (memory) is lost after a reboot, so snapshots help capture and persist the VM's state.

o Program Data Goes to RAM:

o RAM (Random Access Memory) is temporary storage that holds the data needed by running programs. Once the program stops running, the data in RAM is lost.

o Stack: Used for storing local variables and function calls. The stack has a predefined size and is used for primitive data types (integers, chars, etc.).

o Heap: Used for dynamic memory allocation. Objects or data structures are stored here, and the memory is managed via pointers.

o String Pool in Java:

o In Java, a String Pool is a special storage area in memory where strings are stored. When you create a string, Java checks if the string already exists in the pool. If it does, Java reuses that string; if not, it adds the string to the pool.

o This helps reduce memory usage since multiple identical string objects are not created.

o Serialization:

o Serialization is the process of converting an object's state into a format (like a byte stream) that can be stored or transmitted. This is useful for saving an object's state (e.g., in a file or database) and later restoring it.

o Example: Saving an object’s state to a file, and then reading it back from the file later.

o VM Instance State Stored as Snapshot:

o When you take a snapshot of a VM, you capture its state: the running processes, memory, and configuration.

o This allows you to restore the VM to that exact state later, making it useful for backups, testing, or recovery scenarios.

o Persistence in VM:

o After you install software (like Java) on a VM, the changes are saved to the VM’s hard disk. So, even after a reboot, Java will still be installed because its data is persistent in the hard disk.

o Encrypted VM State:

o Encryption is often used to protect sensitive data. If a VM's state (or snapshot) is encrypted, it ensures that unauthorized users cannot view or modify the VM's configuration or data.

o Scaling VMs:

o Scale factors refer to the ability to resize a VM:

➢ Increase Size: Allocate more resources (e.g., CPU, RAM, storage) to the VM.

➢ Decrease Size: Reduce resources allocated to the VM to save costs or optimize performance.

➢ This is useful when your application needs more resources (scale up) or when resources are not fully utilized (scale down).

o DHCP Server:

o DHCP (Dynamic Host Configuration Protocol) is used to automatically assign IP addresses to devices on a network.

o How it works: A DHCP server manages a pool of IP addresses and allocates them to devices (like VMs, computers, or smartphones) as they join the network.

o Example:

➢ When a VM starts, it sends a DHCP request for an IP address.

➢ The DHCP server responds with an available IP address from its pool, and the VM can use this IP to communicate on the network.

o

o Summary:

o Virtual Machines (VMs) allow you to run multiple OS instances on one physical machine, with each VM behaving like a separate computer.

o Linux distributions come in different families, like Ubuntu (Debian-based) and Red Hat (RHEL-based), with different tools (e.g., APT) for managing software.

o VM Snapshots save the entire state of a VM, allowing you to restore it to a previous configuration.

o RAM holds temporary data, and hard disk provides permanent storage for program data.

o DHCP servers automatically assign IP addresses to devices, making network management easier.

 In Aws vm called as EC2 instance

 All services using vm

 Vpc doesn’t belong to vm

 subnet

Key Components of a Data Center:

1. Servers: These are the computers that process and store data.

2. Storage Devices: These devices hold the data, like hard drives or solid-state drives (SSDs).

3. Networking Equipment: Devices like routers, switches, and firewalls that manage data traffic between servers and external networks.

4. Cooling Systems: To keep all the equipment from overheating, data centers are equipped with powerful cooling systems.

5. Power Supply: Data centers have backup generators and uninterruptible power supplies (UPS) to ensure continuous power in case of outages.

6. Security system

TYPES OF DATA CENTERS

1. On-premise Data Centers:

• What it is: A data center owned and managed by your company on your premises (in your office or building).

• Example: A company has its own servers and equipment in a server room within its office.

• Pros: Full control over your infrastructure.

• Cons: High costs and maintenance.

2. Colocation Data Centers:

• What it is: You rent space in a data center and place your own equipment there. The provider takes care of power, cooling, and security.

• Example: A company rents space in a facility and uses its own servers and equipment.

• Pros: No need to build a data center yourself, but you still control the hardware.

• Cons: You still manage your own equipment.

3. Cloud Data Centers:

• What it is: These are remote data centers managed by cloud providers (like AWS, Microsoft Azure, Google Cloud). You rent resources like storage, servers, and apps over the internet.

• Example: Using cloud services like Amazon Web Services (AWS) or Google Cloud to store data or run applications without owning hardware.

• Types:

o IaaS: Rent virtual machines and infrastructure.

o PaaS: Rent a platform for building apps without managing servers.

o SaaS: Use software like Gmail or Dropbox without managing anything.

Tomcat:

• What it is: A server used to run Java-based web applications.

• Example: If you're running a Java-based website, Tomcat is used to host and serve the site.

Design Considerations:

1. Reliability:

• What it is: Ensures the data center is always available and operates without interruptions.

• Key Factors:

o Fault tolerance: The ability to keep working even if something fails (e.g., backup systems or redundant parts).

o Backup and Redundancy: Having backup systems like power sources (generators, UPS) and additional hardware to prevent downtime.

2. Security:

• What it is: Protecting the data center from unauthorized access and cyber threats.

• Key Factors:

o Physical security: Locks, cameras, and guards to protect the physical premises.

o Cybersecurity: Firewalls, encryption, and intrusion detection to safeguard data and systems from hackers.

3. Scalability:

• What it is: The ability to grow and handle increased demand.

• Key Factors:

o As your company or services grow, the data center should be able to expand (add more servers, storage, etc.) without issues.

4. Energy Efficiency:

• What it is: Reducing energy consumption and costs.

• Key Factors:

o Using energy-efficient hardware, cooling systems, and power-saving techniques helps lower costs and reduce environmental impact.

5. Disaster Recovery:

• What it is: Plans and systems to recover quickly after a disaster (like a fire, flood, or power failure).

• Key Factors:

o Backups: Keeping copies of important data off-site.

o Disaster recovery plans: Having processes to restore operations and minimize downtime in case of an emergency.

Summary:

• Reliability: Ensures uptime with backups and redundancy.

• Security: Protects against physical and cyber threats.

• Scalability: Allows growth as your needs increase.

• Energy Efficiency: Reduces energy consumption and costs.

• Disaster Recovery: Prepares for emergencies to minimize downtime.

These factors are crucial for ensuring that the data center can operate smoothly, securely, and efficiently.

Types of Data Centers:

1. On-premise Data Centers:

o Owned and Operated: The company owns and manages the data center.

o Complete Control: The company has full control over the hardware, software, and security.

o High Initial Investment: Setting up an on-premise data center requires a significant upfront cost for equipment and infrastructure.

o Example: A large corporation builds its own data center to store its servers and sensitive data.

Why Load Balancing is Important?

Common Load Balancing Techniques:

1. Round Robin: Distributes requests to servers one by one in a circular order. Each server gets a request in turn.
2. Least Connections: Routes traffic to the server with the least active connections, ensuring servers are not overloaded.
3. Least Response Time: Directs requests to the server with the fastest response time, improving performance.
4. Source IP Hashing: Uses the IP address of the client to determine which server will handle the request, ensuring that requests from the same client are consistently directed to the same server.
5. Weighted Round Robin: Distributes traffic based on the server's weight or capability. More powerful servers get more traffic.

Types of Load Balancers:

1. Hardware Load Balancers:

o High Upfront Cost: Requires purchasing physical devices (hardware).

o Example: A company buys a dedicated hardware appliance that handles all incoming network traffic and distributes it to the servers.

2. Software Load Balancers:

o Application Layer: Operates at the application layer (Layer 7) and can handle specific traffic based on the type of request.

o Performance: Generally not as reliable or fast as hardware-based load balancers.

o Example: Software like NGINX or HAProxy used for load balancing requests to web servers.

Benefits of Load Balancers:

1. High Availability (HA): Ensures that the application remains available even if one or more servers fail by automatically redirecting traffic to healthy servers.

Key Components of High Availability (HA):

• Redundancy: Extra servers, power supplies, and components to avoid single points of failure.

• Failover: Automatic switching to backup systems when the primary system fails.

• Health Monitoring: Constant monitoring of systems to detect failures early and reroute traffic before it causes a problem.

• Load Balancing: Distributing network traffic to multiple servers to avoid overloading any single server and ensure a smooth user experience.

Summary:

• Load Balancing is crucial for ensuring fault tolerance, improving performance, and managing traffic to multiple servers.

• Types of Load Balancers: Hardware (expensive) vs. Software (more flexible but less reliable).

• Key Components of HA: Redundancy, failover, health monitoring, and load balancing ensure high availability and minimal downtime.

Storage Types

• Direct Attached Storage (DAS):

* DAS refers to storage devices that are directly attached to a single computer or server, without being connected to a network. It is typically used for personal or small-scale applications.

* How it works: DAS storage is directly connected via interfaces such as USB, SATA, SAS, or Thunderbolt.

* Local storage directly attached to servers.

*Lack of scalability

*Potential to lost data

*Device specific

* Examples: External hard drives, internal hard drives, SSDs (Solid-State Drives).

• Network Attached Storage (NAS):

*NAS is a storage device connected to a network, allowing multiple users and devices to access data over the network. It's often used for centralized file storage and sharing.

*How it works: NAS devices typically use Ethernet or Wi-Fi to connect to a local area network (LAN), and they present storage over protocols like SMB/CIFS (Windows), NFS (Linux/Unix), or AFP (Apple).

*File-based storage accessible over a network.

*Moderate scalability and perfomance

*Examples: Synology NAS, QNAP NAS, WD My Cloud.

• Storage Area Networks (SAN):

*SAN is a high-speed network that connects storage devices (such as disk arrays) with servers, enabling block-level data access. Unlike NAS, which provides file-level access, SAN provides block-level access to data, typically used for large-scale enterprise applications.

*How it works: SANs often use Fibre Channel, iSCSI, or FCoE (Fibre Channel over Ethernet) to connect storage devices and servers. Data is accessed as blocks rather than files.

*High-speed network of storage devices, often used for large-scale enterprise data storage.

*Examples: EMC VMAX, NetApp FAS, Dell PowerMax

TYPES OF STORAGES

PRIMARY STORAGE

Primary storage refers to the storage that is directly accessible by the CPU and is used to store data and instructions that are actively being processed.

1. RAM (Random access memory)

Description: RAM is the most common type of primary storage and is used to store data and instructions that the CPU needs to access quickly while performing tasks.

Characteristics:

• Fast read and write access.

• Volatile memory (data is lost when power is turned off).

• Temporarily stores data being processed by running applications.

1. ROM(Read only memory)

Definition: ROM is a type of non-volatile memory used in computers and other electronic devices to store permanent data or instructions that are not meant to be altered or modified during normal operation.

Characteristics:

• Non volatile

• Read only

• Slower access

SECONDARY STORAGE

1. HDD (Hard Disk Drives)

2. SSD (Solid-State Drives)

Description: HDDs are mechanical storage devices that use spinning disks to read/write data. They are widely used for long-term storage.

Characteristics:

1. Larger capacity compared to primary storage.

2. Slower read/write speeds due to mechanical components.

3. Non-volatile (data persists even without power).

Description: SSDs are storage devices that use flash memory to store data, providing faster read/write speeds compared to HDDs.

Characteristics:

1. Faster than HDDs but generally more expensive.

2. Larger capacity compared to primary storage.

3. Non-volatile, retains data without power.

OPTICAL DISC

• An optical disc is a storage medium that uses laser light to read and write data on a reflective surface.

• Optical discs are widely used for storing data such as software, music, videos, and backups.

• Example : CD, DVD, Floppy Disc

RAID LEVELS

RAID 0 (Striping)

• Configuration: Data is split into blocks and distributed across multiple disks (at least 2).

• Redundancy: No redundancy—if one drive fails, all data is lost.

• Performance: High performance, as data is read and written in parallel to multiple drives.

• Capacity: Total capacity is the sum of the capacities of all disks.

• Use Case: Suitable for applications requiring high performance and where data loss is not critical (e.g., temporary data, non-essential files).

RAID 1 (Mirroring)

• Configuration: Data is duplicated (mirrored) across two or more disks.

• Redundancy: High redundancy—if one drive fails, the data is still available from the other drive(s).

• Performance: Good read performance (because the system can read from multiple disks), but write performance is similar to a single disk.

• Capacity: Total capacity is the size of one drive (since data is duplicated).

• Use Case: Suitable for situations where data integrity is critical and write performance is not as important (e.g., personal computers, critical data storage).

RAID 5 (Striping with Parity)

• Configuration: Data is striped across multiple disks (at least 3), with parity information distributed across all disks.

• Redundancy: Moderate redundancy—if one disk fails, the data can be rebuilt using the parity data from the remaining disks.

• Performance: Good read performance, but write performance is slower compared to RAID 0 and RAID 1 due to the overhead of parity calculations.

• Capacity: Total capacity is the sum of all disks minus one (because one disk is used for parity).

• Use Case: Suitable for applications that require a balance of redundancy, performance, and storage capacity (e.g., file servers, databases).

RAID 6 (Striping with Double Parity)

• Configuration: Similar to RAID 5 but with two sets of parity data, which are stored across different disks (requires at least 4 disks).

• Redundancy: High redundancy—can tolerate the failure of two disks simultaneously without data loss.

• Performance: Read performance is good, but write performance is slower than RAID 5 because of double parity calculations.

• Capacity: Total capacity is the sum of all disks minus two (because two disks are used for parity).

• Use Case: Suitable for environments where data protection is more important than write performance (e.g., critical business data storage).

RAID 10 (RAID 1+0)

• Configuration: Combines the features of RAID 1 and RAID 0. Data is mirrored (RAID 1) and then striped (RAID 0).

• Redundancy: High redundancy—can tolerate the failure of one disk per mirrored pair.

• Performance: High performance for both read and write operations, as data is striped (RAID 0) and mirrored (RAID 1).

• Capacity: Total capacity is the sum of half of the disks (since data is mirrored).

• Use Case: Suitable for applications that require both high performance and redundancy (e.g., databases, high-performance servers).

BACKUP AND RECOVERY

A backup is the process of creating a duplicate copy of data that can be restored in case the original data is lost, corrupted, or inaccessible.

TYPES OF BACKUP

Full Backup

• Definition: A full backup is a complete copy of all selected data. It copies everything, including all files, folders, and system data (depending on the configuration).

• How It Works: Every time a full backup is performed, all data is backed up in its entirety, regardless of whether it has changed since the last backup.

Incremental Backup

• Definition: An incremental backup only backs up the data that has changed since the last backup (whether it was a full backup or the most recent incremental backup).

• How It Works: After an initial full backup, subsequent incremental backups only capture changes made to files since the last backup. This can be multiple times over a period.

Differential Backup

• Definition: A differential backup captures all the changes made since the last full backup. Unlike incremental backups, differential backups do not rely on previous differential backups, but only on the full backup.

Mirror Backup

• Definition: A mirror backup creates an exact copy (or "mirror") of the selected data.

It is similar to a full backup but continuously synchronizes data between the source and the backup location.

3-2-1 BACKUP STRATEGY

The 3-2-1 backup strategy is a widely recommended method for ensuring robust data protection and recovery. It helps mitigate the risks of data loss from various types of disasters (e.g., hardware failure, cyberattacks, accidental deletions). This strategy involves creating multiple copies of data and storing them in different locations to increase redundancy and resilience.2 Copies stores in two different media types and one in offsite.

BASIC SERVER COMPONENTS

• MOTHER BOARD: A motherboard is the central printed circuit board (PCB) in a computer that connects and allows communication between various hardware components. It serves as the backbone of the computer, providing essential connections for components like the CPU, RAM

• CPU

• RAM

• NIC

• STORAGE DRIVE

LOAD BALANCING

• ROUND ROBIN: Round Robin is one of the simplest and most commonly used load balancing algorithms. It is a method used to distribute client requests (or traffic) across a group of servers or resources in a circular order.

• LEAST CONNECTION: Least Connections is a dynamic load balancing algorithm that directs incoming traffic to the server with the fewest active connections at the time of the request. This method is designed to distribute load based on the number of active connections each server is currently handling, aiming to prevent overloading any single server.

• LEAST RESPONSE TIME: Least Response Time is a dynamic load balancing algorithm that routes incoming client requests to the server with the quickest response time at the moment of the request. The goal of this algorithm is to optimize user experience by sending traffic to the server that is not only least loaded but also currently capable of processing requests the fastest.

• SOURCE IP HASHING: Source IP Hashing is a load balancing algorithm that uses the client's IP address to determine which server in the pool should handle a particular request. The key idea behind this approach is to ensure that requests from the same client IP address are always directed to the same backend server, creating session persistence (also called sticky sessions).

• WEIGHTED ROUND ROBIN: Weighted Round Robin (WRR) is an enhancement of the traditional Round Robin load balancing algorithm. In Weighted Round Robin, each server in the pool is assigned a weight that reflects its capacity or performance. The load balancer distributes incoming requests across the servers, but it gives more requests to servers with higher weights, effectively allowing more powerful servers to handle more traffic.

TYPES OF LOAD BALANCER

HARDWARE LOAD BALANCER

Hardware load balancer is a physical appliance designed specifically for load balancing tasks. It is a dedicated device with specialized hardware and software to handle traffic distribution efficiently.

SOFTWARE LOAD BALANCER

A software load balancer is a software application that runs on general-purpose hardware (such as a server or virtual machine) to perform load balancing tasks. It uses algorithms and protocols to distribute traffic among multiple serve