

Cloud Computing

Prepared by Dr. Sherif Mostafa

IT381

Lecture 9



Storage Architecture and Types (Centralized Vs Distributed)

What is virtualization architecture and its types

What is cloud storage?

Benefits of a virtualization architecture

Virtualization and Containerization

Centralized Storage vs. Distributed Storage.

What is virtualization architecture?

A virtualization architecture is a conceptual model of a virtual infrastructure that is most frequently applied in cloud computing. Virtualization itself is the process of creating and delivering a virtual rather than a physical version of something. This could be a desktop, an operating system (OS), a server, a storage device or network resources.

The architecture clearly specifies the arrangement and interrelationships among the particular components in the virtual environment.

What is cloud storage?

In cloud computing, virtualization facilitates the creation of virtual versions of hardware such as desktops, as well as virtual ecosystems for OS, storage, memory and networking resources. A virtualization architecture runs multiple OSes on the same machine using the same hardware and also ensures their smooth functioning.

What is cloud storage?

In a virtualization architecture, specialized software is used to create a virtual version of a computing resource. This eliminates the need to re-create an actual version of that resource. A logical name is assigned to the resource and a pointer is provided to that resource on demand. As a result, multiple OSes and applications can run on the same machine and multiple users (or organizations) can share a single physical instance of a resource or application at the same time.

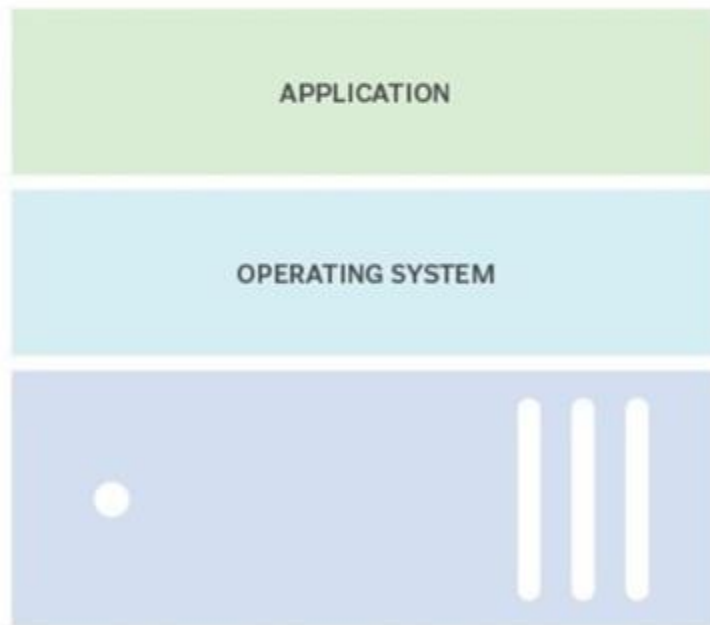
What is cloud storage?

The virtualization architecture is a visual depiction or model of virtualization. It maps out and describes the various virtual elements in the ecosystem, including the following:

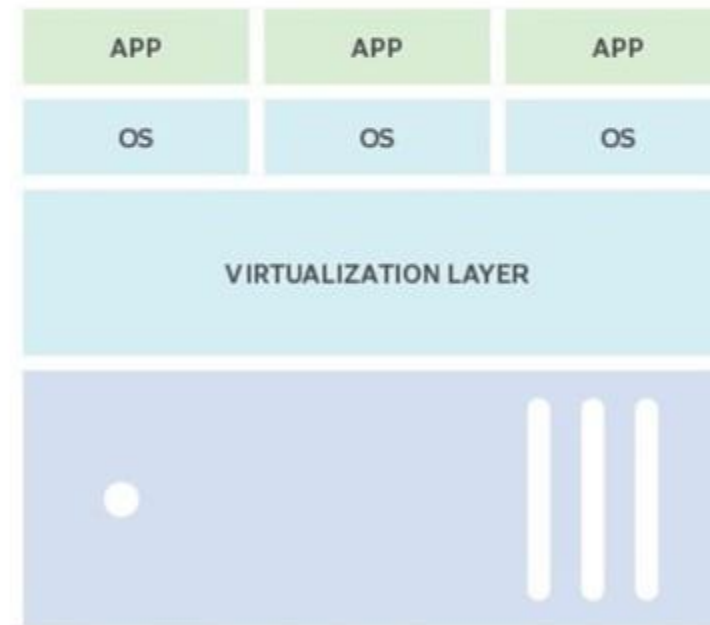
- application virtual services
- infrastructure virtual services
- virtual OS
- hypervisor

The application and infrastructure virtual services are embedded into a virtual data center or OS. The hypervisor separates the OS from the underlying hardware and enables a host machine to simultaneously run multiple VMs which will share the same physical resources.

Traditional and virtual architecture



Traditional architecture



Virtual architecture

Cloud computing and virtualization architecture

Virtualization and virtualization architecture are important ideas in cloud computing. In fact, the terms are often used interchangeably because cloud computing by definition involves virtual ecosystems. Regardless of whether the ecosystem is private (private cloud) or public (public cloud), virtualization reduces the need for organizations to maintain a physical (on-premises) infrastructure for their computing requirements.

Cloud computing and virtualization architecture

With cloud computing and a virtualization architecture, applications can be shared across multiple active users. In the case of public clouds like Amazon Web Services (AWS) or Microsoft Azure, they can also be shared across multiple enterprises (also known as customers or tenants).

There are two major types of virtualization architecture: hosted and bare-metal. It's important to determine the type that will be used before implementing virtualized systems.

Hosted architecture

In this type of setup, a host OS is first installed on the hardware, followed by the software. The software, which is a hypervisor or virtual machine (VM) monitor, is required to install multiple guest OS or VMs on the hardware in order to set up the virtualization architecture. Once the hypervisor is in place, applications can be installed and run on the VMs just like they are installed on physical machines.

A hosted architecture is best suited for the following:

- software development
- running legacy applications
- supporting multiple OSes
- simplifying system configuration

Types of virtualization architecture - Bare-metal architecture

Bare-metal architecture:

In this architecture, a hypervisor is installed directly on the hardware instead of on top of an OS. The installation for the hypervisor and VMs happens in the same way as with hosted architecture. A bare-metal virtualization architecture is suitable for applications that provide real-time access or perform some type of data processing.

Benefits of bare-metal virtualization include the following:

- effective resource management
- high resource availability
- higher scalability
- better system performance and stability

hypervisors in virtualization architecture

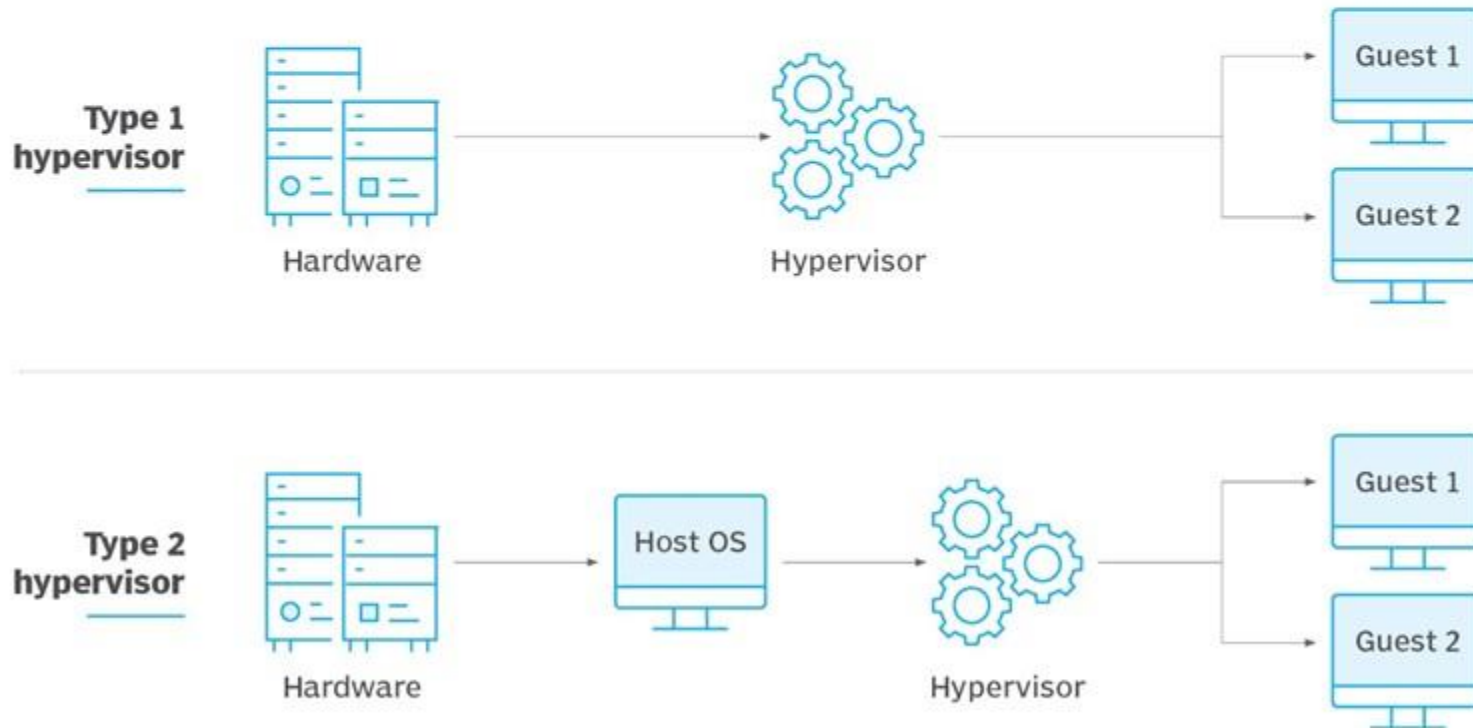
Virtualization is hypervisor-based. The hypervisor isolates the OS and applications from the underlying computer hardware so the host machine can run multiple VMs as guests that share the system's physical compute resources, such as processor cycles, memory space and network bandwidth. Simply put, the hypervisor separates the available power, memory or storage resources and assigns some of these resources to each VM as needed in the virtualization architecture.

hypervisors in virtualization architecture

Type 1 hypervisors, sometimes called bare-metal hypervisors, run directly on top of the host system hardware. Bare-metal hypervisors offer high availability and resource management. Their direct access to system hardware enables better Performance, scalability and stability. Examples include the following:

- Microsoft Hyper-V
- Citrix XenServer
- VMware ESXi

Hypervisor types



type 2 hypervisor, also known as a hosted hypervisor, is installed on top of the host OS, rather than sitting directly on top of the hardware as the type 1 hypervisor does. Each guest OS or VM runs above the hypervisor. The convenience of a known host OS can ease system configuration and management tasks. However, the addition of a host OS layer potentially can limit performance and expose possible OS security flaws. Examples include the following:

- VMware Workstation Pro
- VMware Fusion
- Oracle VirtualBox
- Oracle Solaris Zones
- Oracle VM Server for x86

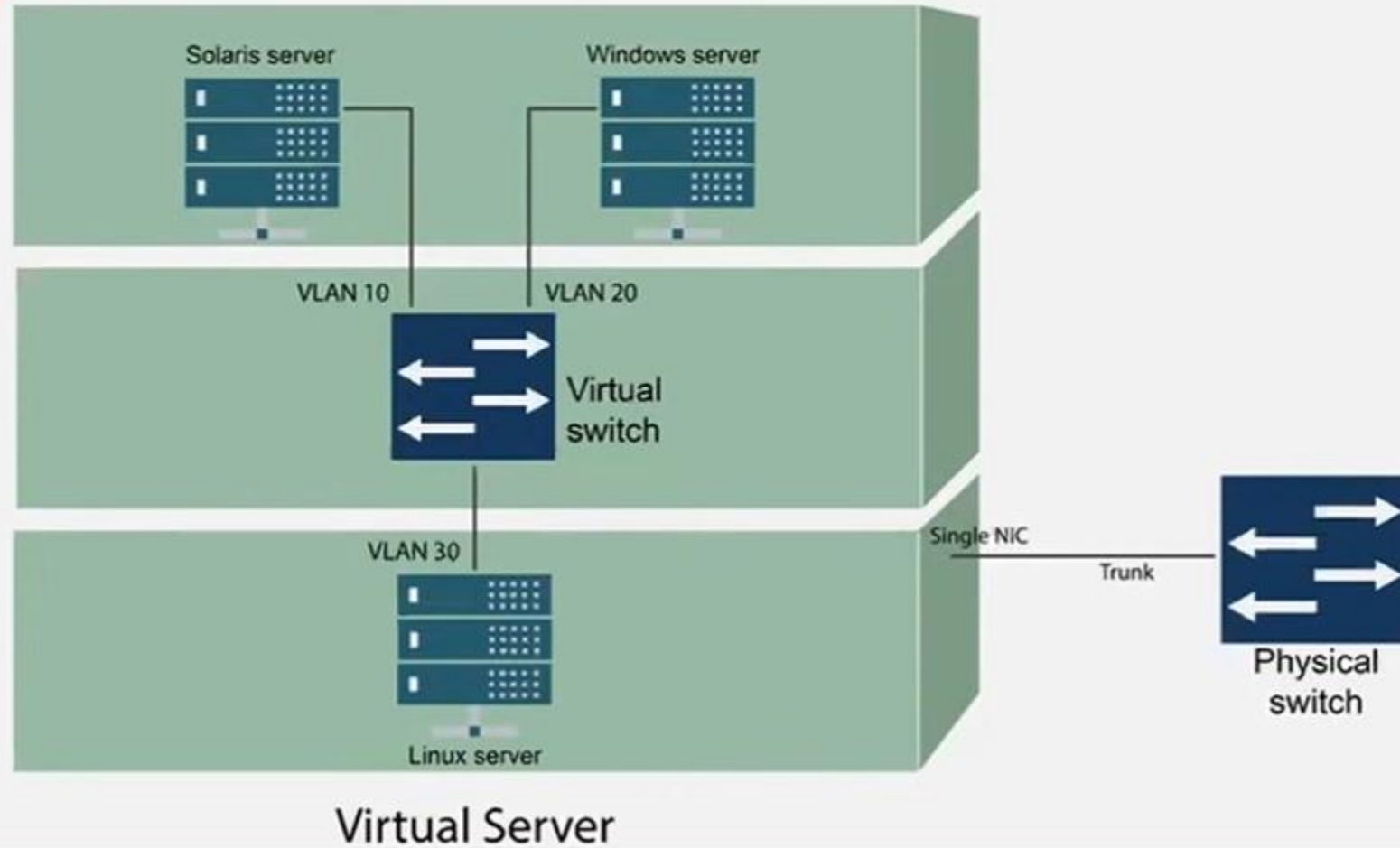
Benefits of a virtualization architecture

A virtualization architecture provides cloud-native organizations with a flexible, cost-effective, highly available way to run multiple virtual OSES or machines using one dedicated resource or host machine. It thus ensures more efficient and affordable resource allocation and also lowers IT infrastructure costs. Also, since the host user can restrict the number of users for the virtual resources, the environment provides better resource control while also reducing unnecessary power and resource consumption.

Benefits of a virtualization architecture

Virtualization also supports remote, anytime-anywhere access, which is critical for remote or geographically dispersed teams. It also reduces the need for physical resources such as servers, which ensures higher uptime and fewer disruptions, better fault-tolerance, efficient load balancing and more available resources. In addition, resources can be spun up or removed on demand, providing greater scalability and agility for enterprise teams.

Benefits of a virtualization architecture



Benefits of a virtualization architecture

All major public cloud providers provide virtualized architectures based on a pay-per-use pricing model. This ensures that organizations only pay for what they actually use instead of having to pay for unused or idle resources or capacity. Consequently, they get better control over their cloud budgets and spends.

What is cloud storage?

Virtualization is also useful from a security standpoint. Each VM can have many guest users that can be OSes, devices, applications, etc. Additionally, the various VMs can be isolated so each guest user gets an isolated virtual environment. Such isolation helps to protect sensitive or business-critical information.

Finally, a virtualization architecture supports aggregation through cluster management software. This software helps to connect homogenous sets of machines or networks that can then act as one unified, aggregated resource.

Benefits of server virtualization



Cost savings



More efficient resource provisioning



Improved productivity



IT consolidation



Better management



App dev is easier, safer



Flexibility and scalability



Hosting multiple OSes



Improved storage capacity management



Business continuity/disaster recovery is easier

Virtualization and Containerization

In cloud computing, one alternative to hypervisor-based virtualization is containerization. Operating system virtualization, for example, is a container-based kernel virtualization method. In this architecture, an OS is adapted so it functions as multiple discrete systems, making it possible to deploy and run distributed applications without launching an entire VM for each one. Instead, multiple isolated systems, called containers, run on a single control host and all access a single kernel.

Virtualization and containerization

Like a VM, a container is a way to create self-contained virtual packages. Essentially, a container is a lightweight VM that is a subset of the same OS instance or the hypervisor. However, where virtualization is a way to run multiple OSes on the hardware of a single physical server, containerization is a way to deploy multiple applications using the same OS on a single VM.

Virtualization and containerization

A VM is best suited for applications where an operating system's full functionality is required. Containers are a better choice for tasks with shorter lifecycles and where there is a need to reduce the number of servers being used for multiple applications.

Centralized Storage vs. Distributed Storage.

Centralized storage plays a critical role in system design by consolidating data in a single location or server, making it easier to manage, secure, and maintain. In this model, all clients and systems access the data over a network, allowing for centralized control and streamlined operations. This approach is commonly used in enterprise environments, data centers, and organizations with significant data management needs. While it offers advantages in simplicity and control, centralized storage systems face challenges in scalability, performance, and resilience, making careful planning essential for a balanced, efficient design.

What is Centralized Storage?

Centralized storage refers to a system where all data is stored in a single, centralized location, typically on one or more servers in a data center. All users, applications, and devices access the data through a network, but the physical storage is managed in one central hub. This model is commonly used in enterprises for managing data, ensuring consistency, and simplifying maintenance and security protocols.

What is Centralized Storage?

Key Characteristics:

Single Point of Storage: Data is stored in one location, often on dedicated storage servers.

Centralized Management: Administrators can control and monitor data access, backups, and security from one place.

Network Dependency: All users access data over a network, making network performance critical to user experience.

Types of Centralized Storage Systems

There are several types of centralized storage systems, each designed to serve different needs in terms of scalability, performance, and data management.

Here are the most common types:

- Direct Attached Storage (DAS)
- Unified Storage
- Mainframe Storage

Types of Centralized Storage Systems

Direct Attached Storage (DAS):

Description: Storage that is directly attached to a single server, typically via internal drives or external hard drives.

Use Case: Small-scale environments or individual servers where storage is needed for local access only.

Benefits: Easy to set up, low cost, and provides fast access for the attached server.

Challenges: Limited scalability, lack of sharing across networks, and difficult to manage in larger setups.

Unified Storage:

Description: A storage system that supports both file-level (NAS) and block-level (SAN) storage protocols, allowing organizations to manage different storage types under one system.

Use Case: Organizations that need both file and block storage but want to manage it from a single system.

Benefits: Flexibility, centralized management, and the ability to serve multiple storage needs with a single solution.

Challenges: Complexity in configuration, higher cost than dedicated NAS or SAN systems.

Mainframe Storage:

Description: Centralized storage designed specifically for mainframe computing environments, often used in large enterprises for mission-critical applications.

Use Case: Large enterprises running legacy systems or applications that require high throughput, availability, and disaster recovery.

Benefits: High reliability, strong integration with mainframe environments, and robust fault tolerance.

Challenges: Expensive and typically confined to large-scale enterprises with legacy infrastructure.

Design Considerations for Centralized Storage

When designing a centralized storage system, there are several important factors and considerations to ensure optimal performance, scalability, security, and reliability. Here are the key design considerations:

Capacity Planning: Determine the amount of data that needs to be stored and estimate future growth.

Performance: The speed at which data can be read from or written to storage is critical, especially for high-traffic environments.

Data Availability and Redundancy: Ensure data is always accessible, even in case of hardware failures or disasters.

Design Considerations for Centralized Storage

Security: Protect stored data from unauthorized access and cyberattacks.

Scalability: The system should be designed to scale easily as data volume grows.

Data Consistency and Integrity: Ensure data remains consistent and uncorrupted, even in high-traffic or failure-prone environments.

Backup and Recovery: Ensure data can be restored in the event of loss or corruption.

Challenges of Centralized Storage in system design

Single Point of Failure: If the centralized storage system goes down, it can cause widespread outages, impacting all connected services and applications.

Scalability Limitations: As data grows, scaling centralized storage can become more complex and expensive, particularly when handling massive volumes of data or users.

Latency and Bandwidth Issues: Accessing centralized storage from geographically distributed locations can introduce latency and increase bandwidth costs, impacting performance.

Challenges of Centralized Storage in system design

High Availability and Redundancy Costs: Ensuring continuous availability requires costly failover systems, redundancy mechanisms, and backup solutions to avoid data loss.

Performance Bottlenecks: Centralized storage systems may experience bottlenecks under heavy loads, especially if they are not designed to handle concurrent access.

Security Risks: Centralizing data in one location can make it an attractive target for cyberattacks, and a breach could expose a large volume of sensitive information.

Complex Disaster Recovery: Recovering from a failure in centralized storage may require complex procedures, particularly if the storage handles mission-critical data across systems.

What is Distributed Storage?

In our modern, data-centric society, handling and securing large volumes of information is paramount. Distributed repository, implied by its name, denotes a framework where data isn't limited to just one place or gadget. Rather, it's dispersed across several locations, typically through a network of linked computers or repository mechanisms.

What is Distributed Storage?

Decentralized repository is a framework where information is stored across multiple devices or locations, rather than being confined to a single device or location. This approach utilizes a network of interconnected computers or storage devices to manage, access, and disseminate data. It enhances information availability, redundancy, and access speed by allowing simultaneous information retrieval from different nodes or locations.

How Distributed Storage Works

At its core, decentralized storage divides information into chunks and distributes them across different servers or devices. This decentralization:

- **Ensures redundancy**, which boosts data recovery and backup.
- **Balances the load**, spreading the information access requests among multiple nodes.
- **Enhances access speed**, as multiple users can retrieve different pieces of data simultaneously.

There are two main types of distributed repository frameworks:

File-based decentralized repository frameworks: These systems store information in files, which are then decentralized across the nodes. Each node stores a portion of the files, and the framework ensures that the files are always consistent.

Object-based decentralized repository frameworks: These frameworks store information in objects, which are uniquely identified by a key. Each object is stored on one or more nodes, and the framework ensures that the objects are always accessible.

How Distributed Storage Works

Here are some of the key components of a decentralized storage framework:

Nodes: The nodes are the individual repository devices that make up the framework. Each node stores a portion of the data, and the framework ensures that the data is always consistent.

Network: The network connects the nodes together. The network must be reliable and high-performance to ensure that the data can be accessed quickly and efficiently.

Storage management software: The repository management software is responsible for managing the information on the nodes. The software ensures that the data is stored in a way that is efficient and secure.

Replication: Replication is the process of copying information to multiple nodes. Replication is used to ensure that the data is always available, even if one or more nodes fail.

Coordination: Coordination is the process of ensuring that the information on all of the nodes is consistent. Coordination is necessary to prevent data corruption and ensure that users always see the same information.

How Distributed Storage Works

Decentralized repository frameworks are used in a variety of applications, including:

Web hosting: Distributed storage frameworks are used to store the static content of websites, such as images and JavaScript files.

Content delivery networks (CDNs): CDNs use decentralized repository frameworks to deliver content to users from the closest possible location.

Backup and disaster recovery: Distributed storage frameworks can be used to store backups of information, which can be used to restore the data in the event of a disaster.

How Distributed Storage Works

Distributed storage systems are a complex technology, but they offer a number of advantages over traditional centralized storage systems. Distributed storage systems are highly available, scalable, and cost-effective, making them a good choice for a variety of applications.

Why is the Distributed Storage System Becoming So Important?

There are a number of reasons why decentralized repository frameworks are becoming so important:

- The amount of information is growing exponentially.
- The amount of information that is being generated and stored is growing at an exponential rate. This growth is being driven by a number of factors, including:
 - the increasing use of the internet
 - the proliferation of mobile devices
 - the growth of big data analytics.

Why is the Distributed Storage System Becoming So Important?

Traditional centralized storage frameworks are not scalable. Traditional centralized storage systems are not designed to scale to the massive amounts of data that are being generated today.

This is because centralized frameworks have a single point of failure, and they can become bottlenecks as the amount of information increases.

Decentralized storage frameworks are scalable and fault-tolerant.

Decentralized storage systems are designed to scale to the massive amounts of data that are being generated today.

Why is the Distributed Storage System Becoming So Important?

This is because distributed frameworks have multiple nodes, and the information is replicated across the nodes. This replication ensures that the data is always available, even if one or more nodes fail.

Decentralized repository frameworks are cost-effective.

Distributed repository frameworks can be more cost-effective than traditional centralized storage systems.

This is because decentralized frameworks use commodity hardware, and they can be easily scaled up or down as needed.

Why is the Distributed Storage System Becoming So Important?

As the amount of information continues to grow, decentralized storage frameworks will become increasingly important. Distributed storage systems offer a scalable, fault-tolerant, and cost-effective way to store massive amounts of information.

Pros and Cons of Distributed Cloud Storage

Pros:

Cost-Efficiency: The cost per GB of repository often decreases as you store more information. For instance, cloud storage costs per TB might be significantly lower than the storage cost per GB for smaller data volumes.

Accessibility: Data can be accessed from anywhere, anytime.

Reliability: Even if one node fails, data remains accessible from other nodes.

Pros and Cons of Distributed Cloud Storage

Cons:

Complexity: Managing and setting up can be intricate.

Security Concerns: Storing data online poses potential threats if not adequately secure.

Variable Costs: While 1 TB cloud repository cost might seem low, costs can accumulate when accessing, sharing, or transferring large amounts of data frequently

Example of Distributed Storage?

A common example of decentralized repository is cloud storage services like Dropbox, Google Drive, or iCloud.

Users upload their information, which is then stored across multiple servers. They can then share a link, allowing others to download or access the content.

Example of Distributed Storage?

Amazon S3: Amazon S3 is a popular cloud storage service that offers object storage. Objects in S3 are uniquely identified by a key, and they can be stored in any region of the world.

Microsoft Azure Blob Storage: Microsoft Azure Blob Storage is another popular decentralized cloud repository service that offers object storage. Blob Storage is designed for storing large amounts of unstructured data, such as images, videos, and text files.

.

Example of Distributed Storage?

Google Cloud Storage: Google Cloud Storage is a cloud repository service that offers object storage. Cloud repository is designed for storing large amounts of data, and it can be used for a variety of purposes, such as web hosting, backup and disaster recovery, and big information analytics.

HDFS: Hadoop Distributed File System (HDFS) is a decentralized file framework that is designed to run on clusters of commodity hardware. HDFS is commonly used for storing large datasets for big information analytics.

Which is Better: Centralized or Distributed?

The answer depends on the specific needs of a user or business. Centralized frameworks are simpler and can be sufficient for smaller data loads. However, for organizations dealing with massive amounts of information and requiring high availability, decentralized repository systems often offer more advantages in terms of scalability, redundancy, and cost-efficiency.

What is the Difference Between Distributed Storage and Centralized Storage?

To conclude.....

Centralized Storage: All information is stored on a single device or server. It's easier to manage but has limited scalability and poses a risk of a single point of failure.

Distributed Storage: information is spread across several devices or servers, offering redundancy, scalability, and improved performance.



Thank You