Cloud Computing

Unit 2: Data Storage and Cloud Computing

Data Storage: Introduction to Enterprise Data Storage, Direct Attached Storage, Storage Area Network, Network Attached Storage, Data Storage Management, File System, Cloud Data Stores, Using Grids for Data Storage. **Cloud Storage:** Data Management, Provisioning Cloud storage, Data Intensive Technologies for Cloud Computing. **Cloud Storage from LANs to WANs:** Cloud Characteristics, Distributed Data Storage

Introduction to Enterprise Data Storage

Enterprise storage is a centralized repository for business information that provides common data management, protection and data sharing functions through connections to computer systems.

Because enterprises deal with heavy workloads of business-critical information, enterprise storage systems should be scalable for workloads of hundreds of terabytes or even petabytes without relying on excessive cabling or the creation of subsystems.

Other important aspects of an enterprise storage system are unlimited connectivity and support for multiple platforms.

Understanding storage system is an important point in building effective storage system. This will yield cost effective, high performance and ease in managing the systems. The various types of storage subsystems are:

- Direct Attached Storage (DAS)
- Storage Area Network (SAN)
- Network Attached Storage (NAS)

DAS is the basic in a storage system and employed in building SAN and NAS either directly or indirectly. NAS is the top most layer, having SAN and DAS as its base. SAN lies between a DAS and a NAS.

Direct Attached Storage

Direct-attached storage (DAS) is hard disk drives (HDDs) or solid-state drives (SSDs) connected directly inside or outside (in a storage enclosure) to a single computer or server that cannot be accessed by other computers or servers. Unlike NAS and SAN, DAS is not networked through Ethernet or FC switches.

Direct-attached storage (DAS) is a form of data storage accessed through an external drive's direct connection to a computer. Examples include external hard disk drives (HDDs) connected by cable to a desktop or laptop, and solid state drives (SSDs) connected by cable or an M.2 port on the motherboard.

Direct-attached storage is a quick way for users to access storage on a computer, but it's one of the less flexible methods of data storage. Hard drive data can only be accessed on the computer it's plugged into, unlike network-based storage, which can be accessed from multiple locations. It's not as scalable, because you can only connect as many drives as the computer has ports. Depending on the drive used, storage space can be limited.

While HDDs may be considered too slow for many enterprises, they're more affordable and have longer lifespans. SSDs using flash instead of mechanical parts are faster and more durable. They cost more, but the significant performance improvements they offer justify the price for many enterprise needs. Non-volatile memory express (NVMe), a protocol for solid-state drives, uses PCI express buses to connect storage to computer memory, enabling high sequential read/write speeds for stored data.

Following are the two types of Direct Attached Storage (DAS):

1. **Internal DAS:** Internal DAS is a DAS in which the storage device is attached internally to the server or PC by the HBA. In this DAS, HBA is used for high-speed bus connectivity over a short distance.
2. **External DAS:** External DAS is a DAS in which the external storage device is directly connected to the server without any device. In this type of DAS, FCP and SCSI are the protocols which act as an interface between server and the storage device.

Advantage of DAS:

1. DAS is ideal for localized file sharing in environments with a single server or a few servers.
2. DAS devices can offer block-level access or file-level access.
3. DAS also offers ease of management and administration.
4. DAS can still be used locally to store less critical data.
5. Initial cost of DAS is lower than NAS.

Disadvantage of DAS:

1. DAS is limited in its scalability.
2. DAS is limited to dedicated servers.
3. Unused resources cannot be reallocated easily.
4. If the server the device is attached to it is down for any reason, the data stored on attached DAS device is inaccessible.

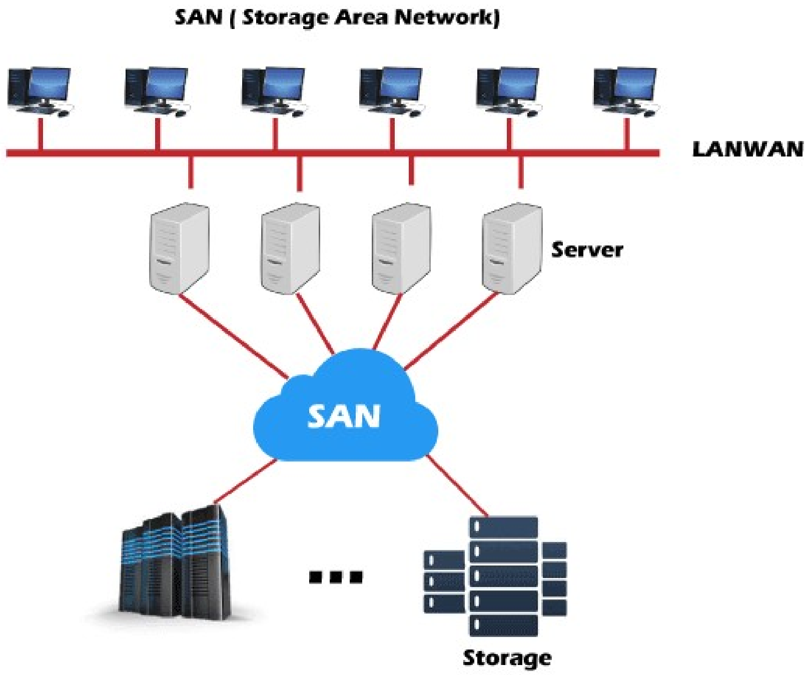
Storage Area Network (SAN)

A **Storage Area Network (SAN)** is a dedicated, independent high-speed network that interconnects and delivers shared pools of storage devices to multiple servers. Each server can access shared storage as if it were a drive directly attached to the server.

A SAN is typically assembled with cabling, host bus adapters, and SAN switches attached to storage arrays and servers. Each switch and storage system on the SAN must be interconnected.

A Storage Area Network (SAN) is a specialized network architecture that provides block-level storage access to servers and applications. SANs are typically used to provide high-speed, scalable storage for mission-critical applications, such as databases, email servers, and virtualized environments.

SANs use specialized hardware and software to provide storage connectivity between servers and storage devices. SANs typically use Fibre Channel (FC) or iSCSI protocols to provide high-speed, low-latency storage access. SANs can be configured in several topologies, including switched fabric, arbitrated loop, and point-to-point connections.



Some of the advantages of SANs include:

- **High performance:** SANs can provide high-speed storage access, with low latency and high throughput, which can be critical for mission-critical applications.
- **Scalability:** SANs can be scaled up to meet growing storage demands, by adding additional storage devices or expanding the network infrastructure.
- **Data protection:** SANs can provide built-in data protection features, such as RAID, replication, and snapshotting, which can help ensure data availability and protect against data loss in case of hardware failures or other disasters.
- **Centralized management:** SANs provide centralized storage management, which can simplify storage administration, backup and restore operations, and data migration.

However, SANs also have some potential disadvantages, including:

- **Complexity:** SANs can be complex to configure and manage, and may require specialized expertise. Careful planning and configuration is required to ensure that the SAN solution is optimized for the specific workload and data access patterns.

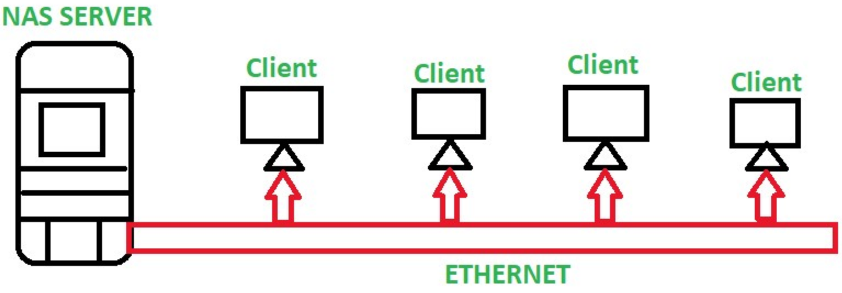
- **Cost:** SANs can be expensive, especially when high-performance or high-availability features are required. Organizations should carefully evaluate the cost of SAN solutions against the benefits they provide before implementing them as a storage solution.
- **Network dependency:** SANs rely on network connectivity to access storage devices, which can make them vulnerable to network congestion or failures. This can result in reduced performance or data unavailability.
- **Security:** SANs can be vulnerable to network-based attacks, such as data interception or unauthorized access. Implementing strong security measures, such as network encryption and access control, is critical to protecting data stored on SANs

Features of Storage Area Networks (SAN):

- **Centralized Storage:** SANs provide a centralized storage architecture that enables multiple servers to access shared storage resources. This centralized storage approach simplifies management and reduces the need for redundant storage devices.
- **High-Speed Data Access:** SANs provide high-speed data access to storage devices through specialized hardware such as Fibre Channel switches and Host Bus Adapters (HBAs). This high-speed data access ensures that data is available to servers quickly and efficiently.
- **Scalability:** SANs are highly scalable and can accommodate a growing number of servers and storage devices. Administrators can add new storage devices and servers to the SAN without disrupting the existing storage environment.
- **Redundancy:** SANs provide high levels of redundancy through features such as RAID and mirroring. This ensures that data is always available, even in the event of a hardware failure.
- **High Availability:** SANs are designed to provide high availability, which ensures that critical applications and services are always accessible. SANs use features such as load balancing and failover to ensure that data is always available to servers.
- **Backup and Recovery:** SANs provide efficient backup and recovery capabilities. Data can be backed up to a centralized storage location, which eliminates the need to back up data from each server individually.
- **Data Protection:** SANs provide advanced data protection features such as encryption and access control. This ensures that sensitive data is protected from unauthorized access and ensures compliance with industry regulations.
- **Remote Data Replication:** SANs provide remote data replication capabilities that enable data to be replicated to a remote location. This ensures that data is available in the event of a disaster or outage.

Network-attached Storage

Network-attached Storage (Commonly known as NAS) is a file storage device which is connected to the network and enables multiple users to access data from the centralized disk capacity. The users on a LAN access the shared storage by the ethernet connection.



(Network Attached Storage Architecture)

This storage is fast, low-cost and offers all the advantages of a public cloud on the site. It uses file access protocols such as NFS, SMB, NCP, or AFP.

NFS is a file-based protocol which is popular on Unix systems. SMB stands for Server Message Block, which is used with the Microsoft Windows systems. AFP is also a file access protocol that is used with the Apple computers.

It is basically designed for those network systems, which may be processing millions of operations per minute. It supports the storage device for the organization, which need a reliable network system. It is more economical than the file servers and more versatile than the external disks.

If you have both UNIX and windows users on your network and you want both groups to be able to share files, NAS devices are most suitable. NAS devices can make use of existing directories of user accounts from Windows, Netware or UNIX server.

Components of a NAS device

NAS devices are typically made up of several components.

1. **Physical storage drives:** NAS devices can contain from two to five hard drives, which gives them high-volume storage capacity. Multiple physical drives are arranged logically as redundant storage containers (RAID). RAID is a virtualization technology that combines multiple physical storage components into one or more logical units. This helps back up data and improve performance.
2. **Central processing unit (CPU):** NAS devices have a CPU that provides computing intelligence and power to manage the file system. The CPU reads and writes data to process and serve files, manage multiple users, and integrate with the cloud if desired.
3. **Operating system:** An operating system is a software interface between the storage device hardware and its user. Although complex network-attached storage devices come with their own operating systems, some simpler devices may not have one.

4. **Networking interface:** The NAS unit connects to the network by using the networking interface. The network connection can be an ethernet cable or Wi-Fi. Many NAS devices also have USB ports for charging or to connect other devices to the NAS device.

Basic storage principle of NAS devices

NAS is network attached storage for file based data. There are three main storage methodologies:

1. **File storage:** In file storage, you can store data in files, organize files into folders, and place them under a hierarchy of directories and subdirectories. It is a popular and familiar storage technique.
2. **Block storage:** Block storage breaks a file into smaller chunks (or blocks) and stores each block separately under a unique address. The computer can store blocks anywhere on the device. The server's operating system uses the unique address to reassemble the blocks into the file. This is faster than looking through hierarchies to access a file.
3. **Object storage:** Objects are discrete units of data that are stored without a hierarchy or structure. Each object includes the data, descriptive information about the data (metadata), and a unique identifying number. Using this information, system software can find and access the object.
4. **File vs. block vs. object storage:** Each storage type can be used in different ways. For example, file storage for local file sharing, and block storage for high-performance applications. On the other hand, you can use object storage to store unstructured data like email, videos, image files, web pages, and sensor data produced by the Internet of Things (IoT).

SAN vs NAS vs DAS Storage: What's the Difference

SAN, NAS, and DAS have the same purpose- storage and they complement each other. However, they are different in terms of original intent, path, and outcome.

- Storage mechanism:
  The main difference between SAN, NAS, and DAS storage is the storage mechanism. SAN uses block storage, NAS uses shared files, and DAS uses hard-drive storage with sectors. The storage is shareable and scalable in terms of performance and capacity for both SAN and NAS, but not DAS or server-centric storage architecture.

- Data transmission:
  The technologies for data transmission are also different. SAN deployment types are IP SAN and fibre channel SAN, NAS uses Ethernet and TCP/IP, and DAS uses IDE/SCSI.

- Advanced features:
  SAN or information-centric architecture offers low latency and high throughput. It works in highly resilient environments and has features like synchronous replication.

NAS offers features like replication, thin provisioning, and snapshots. It is highly flexible and compatible with various operating systems. It can abstract storage management from the server and handle unstructured data like video, audio, text files, and so on.
DAS is less expensive than SAN/ NAS and does not have features like snapshots.

- Storage devices and servers:
  In SAN, the storage devices are managed centrally and can be shared between several computer systems. They can exist independently of the servers.
  In NAS, the storage devices are connected to the network directly for file sharing. It has a dedicated OS for file serving that uses standard protocols such as FTP, NFS, and CIFS.
  In DAS, there are limited storage devices for each server and they only exist in relation to these servers. It uses the host's OS for implementing data backup, management, and other tasks.

	SAN	**NAS**	**DAS**
Type of storage	Blocks	Shared files	Sectors
Transmission of data	Fiber Channel	Ethernet, TCP/IP	IDE/SCSI
Speed	5-10 ms	20-50 ms	5-10 ms
Complexity	High	Moderate	Easy
Mode of Access	Servers	Clients or Servers	Clients or Servers
Capacity	> 10^12 bytes	10^9-10^12 bytes	10^9 bytes
Usage	Application data	Unstructured, Shared data	OS

Data Storage Management

Data storage is expensive; therefore, storage administrators are trying to use tiered storage. Using fibre channel for storing data for a network user gives better performance but storage devices used are small and are expensive. SAS or DAS is cost effective performance-wise it is of lower grade. Today IT organizations are implementing tiered storage as a mix of storage technologies that meet the performance needs and are cost effective.
Data storage management involves the monitoring of software and hardware assets, such as storage arrays, physical servers, and cloud storage services. Data storage management can

involve resolving performance issues like potential bottlenecks and analyzing real-time storage capacity to help improve the end-user experience. With this information, admins can reallocate storage resources to fulfill business storage needs.

Data storage management can also include traffic analysis, process automation, memory management, network virtualization, replication, and storage provisioning. Using reliable data storage management software, organizations can more easily configure and track storage and report related storage activities.

Functions of data storage management

Common functionalities of data storage management include:

1. Performance and reliability: The objective of data storage management is to manage data to have it readily available for business operations. Easy and quick access to data increases the performance, efficiency, and productivity of employees and improves the end-user experience. To streamline the process, teams can use media and automatic tiering to optimize different storage tiers.
2. Security and data protection: While using cloud storage, it’s important to understand the importance of data protection. Protect business-critical data using data backup solutions, encryption for both in-transit and stored data, multifactor authentication to restrict unauthorized access, and more.
3. Control and compliance: Leverage various levels of tiering or automatic tiering to store the most valuable data assets. This helps manage and store data and helps organizations demonstrate compliance with regulations.

Data Storage Management Tools

Maintaining storage devices is a tedious job for storage administrators. They adopt some utilities to monitor and manage storage devices. Management level tasks are configuration, migration, provisioning, archiving and storage monitoring/reporting. Storage Resource Management (SRM) tools include configuration tools, provisioning tools and measurement tools.

- **Configuration tools** handle the set-up of storage resources. These tools help to organize and manage RAID devices by assigning groups, defining levels or assigning spare drives.
- **Provisioning tools** define and control access to storage resources for preventing a network user from being able to use any other user’s storage.
- **Measurement tools** analyze performance based on behavioral information about a storage device. An administrator can use that information for future capacity and upgrade planning.

Storage Management Process

Data storage management tools must rely on policies which governs the usage of storage devices and its procedures. Storage management encompasses three areas—change management, performance and capacity planning and tiering (tiered storage).

The process used to request, schedule, implement and evaluate adjustments to the storage infrastructure is called **change management.** The change management process defines the way a request is made and approved and documents the steps used to configure and provision the requested space on a storage array or server. Change management may also document processes such as data migration and maintains the integrity and availability of that data for network users.
Performance and capacity planning are used to measure the performance of system in-terms of storage and utilization. The result of performance and consumption analysis is used to make sensible decisions about subsequent storage purchases.

Data Storage Challenges

In depth, understanding of storage devices will minimize the risks, and an administrator can easily handle challenges like finding out the reason for performance degrading, cost check, etc. Managing traditional storage devices is a complicated task because of high operations cost, performance and scalability issues. Some challenges are massive data demand, performance barrier, power consumption and cost.

Massive Data Demand

An industry survey estimates the digital world to increase by 45 zettabytes by 2020, that is, one terabyte is equal to 1024 gigabytes, one petabytes is equal to 1024 terabytes, one exabytes is equal to 1024 petabytes and one zettabytes is equal to 1024 exabytes

Performance Barrier

Rapid growth in data has caused a parallel increase in the size of databases. In the traditional storage method, the response time taken for queries is slow and it should be increased. Be it a social networking site, an enterprise database or a web application, all requires faster disk access to read and write data.

Power Consumption and Cost

Because of increase in storage demands, IT organizations and data centres need larger storage with minimal cost. Performance lags with minimal cost but has other expenses like licensing and maintenance. Apart from this, other factors such as power consumed by storage devices, cooling systems, man power for managing it and space for data centres are to be considered.

Unified Storage

A new innovative solution ‘Unified Storage’ is developed and addresses the issues discussed earlier. Basically this type of storage solution is a combination of NAS and SAN and termed as NUS (network unified storage). This type of storage system handles both file and block level accessing and hence storage devices can be accessed by single and multiple hosts. The main advantage of this system is reduced cost and it supports fibre channel and iSCSI

File Systems

File system is an interface between secondary storage device like hard disk and user application. Hard disk is a block oriented device.

The purpose of file systems is to maintain a consistent view of storage so that we can effectively manage it. This is done in a way that allows the users to create files and directories as well as delete, open, close read, write and/or extend the files on the device.

File systems also maintain security over the files by using access control lists for a file.

A file system is a structure used in computer to store data on a hard disk. When we install a new hard disk, we need to partition and format it using a file system before storing data. There are three file systems in use in Windows OS; they are NTFS, FAT32 and rarely-used FAT.

A cloud file system is a hierarchical storage system in the cloud that provides shared access to file data. Users can create, delete, modify, read, and write files, as well as organize them logically in directory trees for intuitive access.

FAT File System

File Allocation Table (FAT) file system is a simple file system originally designed for small disks and simple folder structures.

FAT was planned for systems with very small RAM and small disks. It required much less system resources compared to other file systems like UNIX. It is also found on in flash memory, digital cameras and portable devices. It is used to store file information and extend the life of a hard drive.

Essentially, the FAT system has made a comeback. Thumb or flash drives have become very common and have smaller size that makes the FAT system useful. The smaller sizes are even formatted in FAT16.

Usually the file system operates blocks, not sector. File system blocks are groups of sectors that optimize storage addressing. Modern file systems generally use block sizes from 1 upto 128 sectors.

The number in FAT12, FAT16 and FAT32 stands for

NTFS

NTFS is New Technology File System, is a process that the Windows NT operating system uses for storing, organizing, and finding files on a hard disk efficiently.

In the 1990s, Microsoft recognized that DOS based Windows was inadequate because of demands in business and industry. They started working for better software which can suit larger systems.

NTFS is much simpler than FAT. While files are used, the system areas can be customized, enlarged, or moved as required. NTFS has much more security incorporated. NTFS is not apt for small-sized disks.

The benefits of NTFS are that, compared to other similar file systems like File Allocation Table (FAT) and High-Performance File System (HPFS), NTFS focuses on:

- Performance: NTFS allows file compression so your organization can enjoy increased storage space on a disk.
- Security access control: NTFS will enable you to place permissions on files and folders so you can restrict access to mission-critical data.
- Reliability: NTFS focuses on the consistency of the file system so that in the event of a disaster (such as a power loss or system failure), you can quickly restore your data.
- Disk space utilization: In addition to file compression, NTFS also allows disk quotas. This feature enables businesses to have even more control over storage space.
- File system journaling: This means that you can easily keep a log of—and audit—the files added, modified, or deleted on a drive. This log is called the Master File Table (MFT).

Cloud File System

A cloud file system is a file system that creates a hub and spoke method of distributing data.

The "hub" is the central storage area, typically located at a public cloud provider like Amazon AWS, Microsoft Azure or Google Cloud.

The "spokes" are the organization's local locations (data centers, branch offices, remote offices). At each spoke a software or hardware appliance is installed and it acts as a cache for that location's most active data.

In cloud file systems, the considerations are:

- It must sustain basic file system functionality.
- It should be an open source.
- It should be grown-up enough that users will at least think about trusting their data to it.
- It should be shared, i.e., available over a network.
- It should be paralleling scalable.
- It should provide honest data protection, still on commodity hardware with only internal storage.

Organizations that use cloud computing outsource massive amounts of data and workloads to cloud providers. Due to its low cost, lower management overhead and elasticity, organizations move towards using cloud computing.

In cloud storage, systems host or consumers can find only corruption or loss of data from their service provider's report, when a system failure occurs. This consumer–provider gap creates business risk and complicates compliance SLAs.

A cloud file system should be scalable enough to adopt large organizations file systems under different workloads with good performance requirements. Cloud file systems should have high throughputs then local file systems. Cloud file system should have minimal operation latency. The system should also be scalable to multiple hosts operating in parallel. Transparency and backwards compatibility is important to facilitate migration to the cloud with less effort. Following are some of the cloud file systems.

1. *Ghost File System*

Ghost cloud file system is used in Amazon Web Services (AWS). It gives high redundant elastic mountable, cost-effective and standards-based file system. A fully featured scalable and stable cloud file systems is provided by ghost cloud file system. GFS (Ghost File System) run over Amazon's S3, EC2 and SimpleDB web services. When using GFS, user can have complete control of the data and can be accessed as a standard network disk drive.

Benefits of Ghost CFS

- Elastic and cost efficient: Pay for what you use from 1 GB to hundreds of terabytes.
- Multi-region redundancy: Aiming to take advantage of AWS's 99.99% availability
- Highly secure: Uses your own AWS account (ghost cannot access your data).
- No administration: Scales elastically with built in redundancy—no provisioning or backup.
- Anywhere: Mount on a server or client or access files via a web page or from a mobile phone.

Features of Ghost CFS

- Mature elastic file system in the cloud.
- All files and metadata duplicated across multiple AWS availability regions.
- WebDav for standard mounting on any Linux, Windows or Mac server or client in the world.
- FTP access.
- Web interface for user management and for file upload/download.
- File name search.
- Side-loading of files from torrent and from URL.

2. *Gluster File System*

GlusterFS is an open source, distributed file system capable of handling multiple clients and large data. GlusterFS clusters storage devices over network, aggregating disk and memory resources and managing data as a single unit. GlusterFS is based on a stackable user space design and delivers good performance for even heavier workloads.

GlusterFS supports clients with valid IP address in network. Users no longer locked with legacy storage platforms which are costly and monolithic. GlusterFS gives users the ability to deploy scale-out, virtualized storage, centrally managed pool of storage.

Attributes of GlusterFS include scalability and performance, high availability, global namespace, elastic hash algorithm, elastic volume manager, gluster console manager, and standards-based.

3. *Hadoop File System*

A distributed file system designed to run on commodity hardware is known as Hadoop Distributed File System (HDFS). In HDFS, files are stored in blocks ranging from 64 MB to 1024 MB. The default size is 64 MB. The blocks will be distributed across the cluster and replicated for fault tolerance.

4. *XtreemFS: A Distributed and Replicated File System*

XtreemFS is a distributed, replicated and open source. XtreemFS allows users to mount and access files via WWW. Engaging XtreemFS a user can replicate the files across data centres to reduce network congestion, latency and increase data availability. Installing XtreemFS is quite easy, but replicating the files is bit difficult.

5. *Kosmos File System*

Kosmos Distributed File System (KFS) gives high performance with availability and reliability. For example, search engines, data mining, grid computing, etc. It is deployed in C++ using standard system components such as STL, boost libraries, aio, log4cpp. KFS is incorporated with Hadoop and Hypertable.

6. *CloudFS*

CloudFS is a distributed file system to solve problems when file system is itself provided as a service. CloudFS is based on GlusterFS, a basic distributed file system, and supported by Red Hat and hosted by Fedora.

Google File System (GFS)

Google Inc. developed the Google File System (GFS), a scalable distributed file system (DFS), to meet the company's growing data processing needs.

GFS is a file system designed to handle batch workloads with lots of data. The system is distributed: multiple machines store copies of every file, and multiple machines try to read/write the same file.

- **Fault Tolerance:** Google uses commodity machines because they are cheap and easy to acquire, but the software behind GFS needs to be robust to handle failures of machines, disks, and networks.
- **Large Files:** It's assumed most files are large (i.e. ≥ 100 MB). Small files are supported, but not optimized for.
- **Optimize for Reads + Appends:** The system is optimized for reading (specifically large streaming reads) or appending because web crawling and indexing heavily relied on these operations.
- **High and Consistent Bandwidth:** It's acceptable to have slow operations now and then, but the overall amount of data flowing through the system should be consistent. Again, this stems from Google's crawling and indexing purposes.

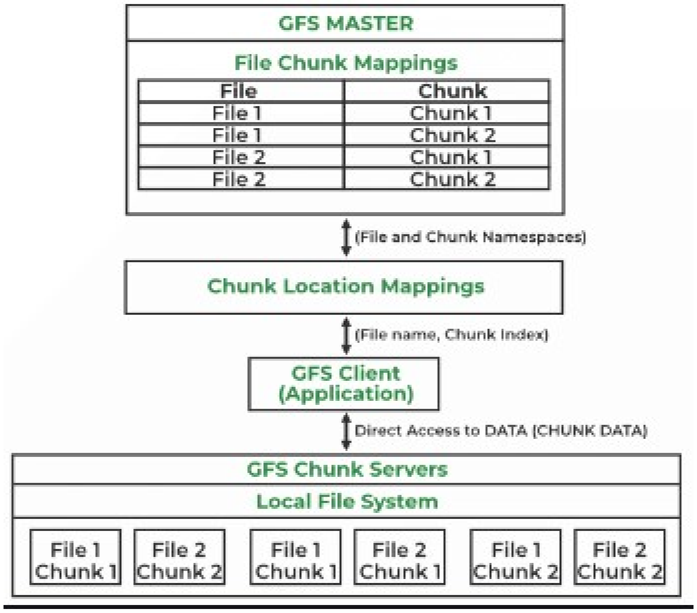
GFS is made up of several storage systems built from low-cost commodity hardware components. It is optimized to accommodate Google's different data use and storage needs, such as its search engine, which generates huge amounts of data that must be stored.

The Google File System capitalized on the strength of off-the-shelf servers while minimizing hardware weaknesses. GFS is also known as GoogleFS.

The GFS node cluster consists of a single master and several chunk servers that various client systems regularly access. On local discs, chunk servers keep data in the form of Linux files. Large (64 MB) pieces of the stored data are split up and replicated at least three times around the network. Reduced network overhead results from the greater chunk size.

Without hindering applications, GFS is made to meet Google's huge cluster requirements. Hierarchical directories with path names are used to store files. The master is in charge of managing metadata, including namespace, access control, and mapping data. The master communicates with each chunk server by timed heartbeat messages and keeps track of its status updates.

More than 1,000 nodes with 300 TB of disc storage capacity make up the largest GFS clusters. This is available for constant access by hundreds of clients.



Components of GFS

A group of computers makes up GFS. A cluster is just a group of connected computers. There could be hundreds or even thousands of computers in each cluster. There are three basic entities included in any GFS cluster as follows:

- **GFS Clients:** They can be computer programs or applications which may be used to request files. Requests may be made to access and modify already-existing files or add new files to the system.
- **GFS Master Server:** It serves as the cluster's coordinator. It preserves a record of the cluster's actions in an operation log. Additionally, it keeps track of the data that describes

chunks, or metadata. The chunks' place in the overall file and which files they belong to are indicated by the metadata to the master server.

- **GFS Chunk Servers:** They are the GFS's workhorses. They keep 64 MB-sized file chunks. The master server does not receive any chunks from the chunk servers. Instead, they directly deliver the client the desired chunks. The GFS makes numerous copies of each chunk and stores them on various chunk servers in order to assure stability; the default is three copies. Every replica is referred to as one.

Features of GFS

- Namespace management and locking.
- Fault tolerance.
- Reduced client and master interaction because of large chunk server size.
- High availability.
- Critical data replication.
- Automatic and efficient data recovery.
- High aggregate throughput.

Advantages of GFS

1. High accessibility Data is still accessible even if a few nodes fail. (replication) Component failures are more common than not, as the saying goes.
2. Excessive throughput. many nodes operating concurrently.
3. Dependable storing. Data that has been corrupted can be found and duplicated.

Disadvantages of GFS

1. Not the best fit for small files.
2. Master may act as a bottleneck.
3. Unable to type at random.
4. Suitable for procedures or data that are written once and only read (appended) later.

Cloud Data Stores

A data store is a digital repository that stores and safeguards the information in computer systems. A data store can be network-connected storage, distributed cloud storage, a physical hard drive, or virtual storage.

It can store both structured data like information tables and unstructured data like emails, images, and videos. Organizations use data stores to retain, share, and manage information across business units.

A data store is a data repository for storing, managing and distributing data sets on an enterprise level.

Data stores can be of different types:

- Relational databases (Examples: MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database)
- Object-oriented databases
- Operational data stores
- Schema-less data stores, e.g. Apache Cassandra or Dynamo
- Paper files
- Data files (spread sheets, flat files, etc)

Distributed Data Store

A distributed data store is a computer network where information is stored on more than one node, often in a replicated fashion. It is usually specifically used to refer to either a distributed database where users store information on a number of nodes, or a computer network in which users store information on a number of peer network nodes.

These kinds of data store are non-relational databases that searches data quickly over a large multiple nodes. Examples for this kind of data storage are Google's BigTable, Amazon's Dynamo and Windows Azure Storage.

Distributed data storage becomes even more essential when there are complex tasks involved. This is because complex tasks involve complex networks and take a lot of time in operation as well as implementation.

The distributed data storage purpose is to not let all your resources to concentrate on a single task. Rather it distributes the resources evenly across all the channels. Depending on the past observations, the distributed data storage approach has proved to be more powerful and resourceful than stand-alone systems.

Types of Data Stores

Established IT organizations have started using advanced technologies for managing large size data, which come from social computing and data analysis applications.

1. *BigTable*

BigTable is a compressed, high performance and proprietary data storage system construct on Google File System. Cloud Bigtable is a sparsely populated table that can scale to billions of rows and thousands of columns, enabling you to store terabytes or even petabytes of data. A single value in each row is indexed; this value is known as the row key.

Bigtable is ideal for storing large amounts of single-keyed data with low latency. It supports high read and write throughput at low latency, and it's an ideal data source for MapReduce operations.

BigTable was developed in 2004 and is used in number of Google applications such as web indexing, Google Earth, Google Reader, Google Maps, Google Book Search, MapReduce, Blogger.com, Google Code hosting, Orkut, YouTube and Gmail.

Advantage for developing BigTable includes scalability and better performance control. BigTable charts two random string values (row and column key) and timestamp into an associated random byte array. BigTable is designed to scale into the petabyte range across multiple machines and easy to add more machines and automatically start using resources available without any configuration changes.

i) Row Key:
   - The row keys in a table are arbitrary strings. Every read or write of data under a single row key is atomic.
   - Bigtable maintains data in lexicographic order by row key. The row range for a table is dynamically partitioned.
   - Each row range is called a tablet, which is the unit of distribution and load balancing.

ii) Column key:
   - Column keys are grouped into sets called column families, which form the basic unit of access control. All data stored in a column family is usually of the same type.
   - A column family must be created before data can be stored under any column key in that family; after a family has been created, any column key within the family can be used.

Other similar softwares are as follows:

- Apache Accumulo: Construct on top of Hadoop, ZooKeeper and economy. Server-side programming mechanism deployed in Java environment.
- Apache Cassandra: Dynamo's distributed design and BigTable's facts and numbers form adds simultaneously in Apache Cassandra, which uses Java.
- Hbase: Supports BigTable and Java programming language.
- Hypertable: Designed for cluster of servers especially for storage and processing.
- KDI: Kosmix stab to make a BigTable clone and is written in C++.

2. *Dynamo: A Distributed Storage System*

Dynamo is a vastly offered, proprietary key-value structured storage system or a dispersed data store. It can act as databases and also distributed hash tables (DHTs). It is used with parts of Amazon web services such as Amazon S3.

Dynamo is the most powerful relational database available in World Wide Web. Relational databases have been used a lot in retail sites, to make visitors browse and search for products easily.

It is difficult to create redundancy and parallelism with relational databases which is a single point failure. Replication is also not possible. Dynamo is a distributed storage system and not a relational database.

Similar to a relational database it stores information to be retrieved; however, it stores the data as objects and not as tables. The advantage of using Dynamo is responsive and consistent in creating a distributed storage solution.

Dynamo is a completely decentralized system with minimal need for manual administration. Storage nodes can be added and removed from Dynamo without requiring any manual partitioning or redistribution.

Compared to Bigtable, Dynamo target applications that require only key/value access with primary focus on high availability where updates are not rejected even in the wake of network partitions or server failures.

Dynamo stores objects associated with a key through a simple interface; it exposes two operations: get() and put().

Dynamo treats both the key and the object supplied by the caller as an opaque array of bytes. It applies a MD5 hash on the key to generate a 128-bit identifier, which is used to determine the storage nodes that are responsible for serving the key.

Dynamo's portioning scheme relies on consistent hashing to distribute the load across multiple storage hosts. In consistent hashing, the output range of a hash function is treated as a fixed circular space or "ring".

Dynamo provides eventual consistency, which allows for updates to be propagated to all replicas asynchronously.

In dynamo, when a client wishes to update an object, it must specify which version it is updating. This is done by passing the context it obtained from an earlier read operation, which contains the vector clock information.

In Dynamo, each storage node has three main software components: request coordination, membership and failure detection, and a local persistence engine. All these components are implemented in Java.

Using Grids for Data Storage

Grid Computing can be defined as a network of computers working together to perform a task that would rather be difficult for a single machine. All machines on that network work under the same protocol to act as a virtual supercomputer.

Grid computing established its stand as an understood architecture, as it provides users and applications to use shared pool of resources. The compute grid connects computers both desktops and servers and storage across an organization. It virtualizes heterogeneous and remotely located components into a single system. Grid computing allows sharing of computing and data resources for multiple workloads and enables collaboration both within and across organizations.

Demand for storage requirement prevails in gird computing. Storage for grid computing requires a common file system to present as a single storage space to all workloads. Presently grid computing system uses NAS type of storage. NAS provides transparency but limits scale and storage management capabilities.

To set the unique demands of the compute grid on its storage infrastructure, storage for the grid must be abnormally flexible. DAS is basically not an option. Virtualization is a start, providing the single unit behavior where the global filing system requires data compute grid. Due to this, SAN architectures are used. However, the scale of these SANs is beyond the capabilities of fibre channel.

Grid Oriented Storage (GOS)

Grid Oriented Storage (GOS) is a dedicated data storage architecture connected directly to a computational grid. It supports and acts as a data bank and reservoirs for data, which can be shared among multiple grid clients. GOS is a successor of Network-Attached Storage (NAS) products in the grid computing era. GOS accelerates all kinds of applications in terms of performance and transparency. A GOS system contains multiple hard disks, arranged into logical, redundant storage containers like traditional file servers. GOS deals with long-distance, heterogeneous and single-image file operations. GOS acts as a file server and uses file-based GOS-FS protocol. Similar to GridFTP, GOS-FS integrates a parallel stream engine and Grid Security Infrastructure (GSI). GOS-FS can be used as an underlying platform to utilize the available bandwidth and accelerate performance in grid-based applications.

Cloud Storage

Cloud storage is a cloud computing model that enables storing data and files on the internet through a cloud computing provider that you access either through the public internet or a dedicated private network connection. The provider securely stores, manages, and maintains the storage servers, infrastructure, and network to ensure you have access to the data when you need it at virtually unlimited scale, and with elastic capacity. Cloud storage removes the need to buy and manage your own data storage infrastructure, giving you agility, scalability, and durability, with anytime, anywhere data access.

Standards and services pertaining to cloud storage have to be understood before its implementation. Resources that are exposed to clients are called as functional interfaces, that is, data paths. Resources maintained by the service providers are called as management interfaces, that is, control paths. A standard model is to be developed and proposed for both interfaces, that is, consumers and providers. That standard should be mapped to various services rendered by the provider. This standard should act as a base for cloud storage interfaces.

Cloud storage came under the limelight because of the following attributes available in cloud computing: pay-as-you-use, elasticity and simplicity (management). It is important that any provider providing storage as a service should also provide these attributes to the consumer. Following are some additional cloud storage attributes:

- Resource pooling and multi-tenancy: Multiple consumers can use shared single storage device. Storage resources are pooled and consumers can be assigned and unassigned resources according to their needs.
- Scalable and elastic: Virtualized storage can be easily expanded on need basis.
- Accessible standard protocols including HTTP, FTP, XML, SOAP and REST.
- Service-based: Consumers no need to invest, that is, no CAPEX (Capital Expenditure) and only pay for usage, that is, OPEX (Operational Expenditure).
- Pricing based on usage
- Shared and collaborative
- On-demand self-service

Cloud storage can be accessible through web-based applications or through web services Application Programming Interfaces (APIs), and using this data are stored. IT organizations have started developing personalized web applications for easy access of cloud storage services.

Data Management for Cloud Storage

Data management encompasses acquiring, storing, protecting and processing data across an organization or business unit. It helps ensure that data is validated and fully accessible to stakeholders when needed.

Data is considered to be a valuable resource for modern organizations. With access to large volumes and different data types, organizations invest significantly in data storage and management infrastructure. They use data management systems to run business intelligence and data analytics operations more efficiently.

Some benefits of data management below.

- *Increase revenue and profit*

  Data analysis gives deeper insights into all aspects of a business. You can action these insights to optimize business operations and reduce costs. Data analysis can also predict the future impact of decisions, improving decision making and business planning. Hence, organizations experience significant revenue growth and profits by improving their data management techniques.

- *Reduce data inconsistency*
  A data silo is a collection of raw data within an organization that only one department or group can access. Data silos create inconsistencies that reduce the reliability of data analysis results. Data management solutions integrate data and create a centralized data view for improved collaboration between departments.
- *Meet regulatory compliance*
  Laws like the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) give consumers control over their data. Individuals can seek legal recourse if they perceive that organizations:
  - Capture data without consent
  - Exercise poor control over data location and use
  - Store data in spite of erasure requests

Hence, organizations require a data management system that is fair, transparent, and confidential while still maintaining accuracy.

For cloud storage, a standard document is placed by SNIA (Storage Networking Industry Association) Storage Industry Resource Domain Model (SIRDM). It states the importance of simplicity for cloud storage.

Figure below shows the SIRDM model which uses CDMI (Cloud Data Management Interface) standards. SIRDM model adopts three metadata: system consisting of storage metadata, data metadata and user metadata. By using these metadata, cloud storage interface can offer services without adding unnecessary complexity in managing the data.

Storage system and data system metadata are used to meet the requirements of the data and the simplicity required is maintained.

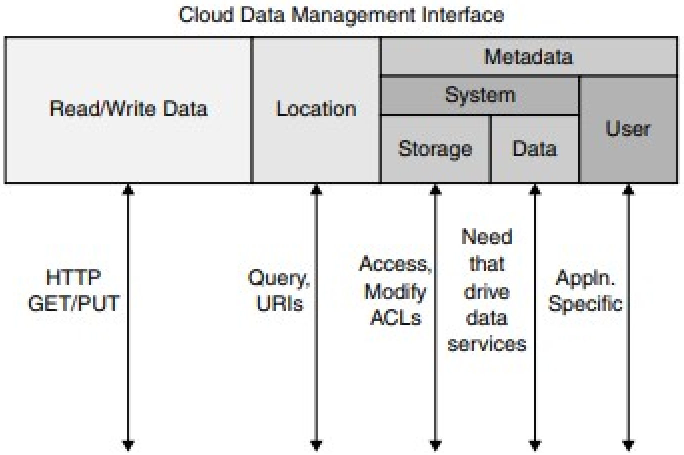


Figure 12.1 Cloud Storage Usage of SIRDM Model

User metadata is used by the cloud to find the data objects and containers.

Storage system metadata is used by the cloud to offer basic storage functions like assigning, modifying and access control.

Data system metadata is used by the cloud to offer data as a service based on user requirements and controls the operation based on that data.

Cloud Data Management Interface (CDMI)

To create, retrieve, update and delete objects in a cloud the cloud data management interface (CDMI) is used. The functions in CDMI are:

- Cloud storage offerings are discovered by clients
- Management of containers and the data
- Sync metadata with containers an objects

CDMI is also used to manage containers, domains, security access and billing information. CDMI standard is also used as protocols for accessing storage.

CDMI defines how to manage data and also ways of storing and retrieving it. ‘Data path’ means how data is stored and retrieved. ‘Control path’ means how data is managed. CDMI standard supports both data path and control path interface.

Cloud Storage Requirements

Multi-tenancy

In a multi-tenancy model, resources provided are pooled, so that it may be shared by multiple customers based on their needs. Due to the elasticity property in cloud computing, shared pool of storage model makes the provider cost effective and billing is made easy.

Security

Secure cloud storage requires a secure transmission channel and methods. Securing data can be done using encryption, authentication and authorization.

- Encryption is the process of scrambling data in such a manner as to make it unreadable without special information, called a key, to make it readable again.
- Authentication is the process of determining their identity. Authentication can employ passwords, biometrics, identifying tokens and other means.
- Authorization determines access rights on the data and the levels of authorization. To provide secure cloud storage, access must be restricted for the communication channel, the data source and the cloud storage sites.

Secure Transmission Channel

The four primary methods used to secure network communications are as follows:

1. Transport Layer Security (TLS) and Secure Sockets Layer (SSL)
2. Hypertext Transfer Protocol Secure (HTTPS)
3. Private Networks
4. Virtual Private Networks (VPNs)

Performance

Cloud storage performance can be categorized into two: speed and latency. Factors that affect cloud storage performance are: available network bandwidth, types of systems available in provider's end, method adopted for compression and caching.

Quality of Service (QoS)

Quality of service (QoS) refers to levels of performance and efficiency of the system that they can provide.

Data Protection and Availability

To ensure that data is protected from loss and theft, providers must take some precautionary measures:

- Physical site security
- Protection against power loss
- Protection against loss of network access
- Data redundancy
- Server redundancy and server fail-over
- Redundant data sites
- Levels of redundancy
- Versioning and data retention
- Accessibility of cloud storage as live data
- Backup to tape or other media
- Data availability, when contract disputes

Metering and Billing

Metering and billing in cloud storage are done based on: data uploaded, data downloaded, data stored and depends on requests and types of request.

Provisioning Cloud Storage

Cloud provisioning means allocating a cloud service provider's resources to a customer. It is a key feature of cloud computing. It refers to how a client gets cloud services and resources from a provider. The cloud services that customers can subscribe to include infrastructure-as-a-service (IaaS), software-as-a-service (SaaS), and platform-as-a-service (PaaS) in public or private environments.

Provisioning in cloud computing involves allocating, configuring, and enabling access to IT resources to address the dynamic needs of an organization. Cloud provisioning aims to ensure that an organization can seamlessly access the required resources in an optimized and efficient way. It also configures various components, such as operating systems, middleware, and applications. Another critical aspect of cloud provisioning is the implementation of security initiatives, such as firewalls, threat detection, and encryption, to ensure the safety, confidentiality, and integrity of critical information and data.

Cloud means sharing third party resources via the Internet. This sharing can be done on need basis and there is no need to invest any infrastructure at consumers end.

Storage clouds increase the efficiency of storing data in remote places, by sharing the storage devices provided by the service providers. Capacity of storage can be increased on need basis and can be done using multi-tenancy methods.

Private storage clouds reside at the back of an organization's firewall that is deployed for in-house customers and is designed for providing elasticity and simplicity in cloud model.

Types of Provisioning in Cloud Computing

Provisioning incorporates the procedures and policies involved in sourcing cloud services. It also amalgamates a client's objectives in sourcing such solutions and services.

The following are the significant types of provisioning in cloud computing.

1. Server Provisioning

Server provisioning is allocating a particular server in a network the specific resources it requires to be operational.

The work the user is designed to handle primarily determines the resources they get. Thus, the parties involved must collect information about the server's intended purpose before the provisioning.

Numerous servers are grouped according to their purpose, each with unique provisioning needs.

2. Cloud Provisioning

Cloud provisioning involves the creation of the basic infrastructure needed by an organization's cloud environment. This includes tasks like installing the essential networking components, services, etc.

After laying down the infrastructure, provisioning cloud service providers ensure that the resources, services, and applications within the cloud are ready for use.

3. User Provisioning

User provisioning is an identity management process that involves giving permissions to applications and services in a corporate environment.

Essentially, it monitors the authentication and authorization of privileges for things like emails, a database, or the entire network.

User provisioning hinges on a user's specific responsibilities or job titles. The act of denying a particular user access is referred to as de-provisioning.

4. Network Provisioning

Network provisioning, with reference to IT infrastructure, involves setting up components such as firewalls, routers, and switches. It also includes allocating IP addresses to these devices and the routine fact-gathering and performance of operational health checks.

5. Service Provisioning

Service provisioning is the setting up IT-dependent services for a user and handling the data involved. A great example of this form of provisioning is when an employee gets access to a SaaS platform.

This stage typically involves system privileges and credentials to limit access to certain activities or data types.

The Benefits of Cloud Provisioning

There are numerous benefits that cloud positioning adds to a business that is difficult to achieve through conventional provisioning methods. Some benefits of provisioning in cloud computing include the following.

1. Scalability: By incorporating cloud provisioning, companies can work more efficiently. They have to make substantial initial investments with conventional IT provisioning models, where they rely on on-site infrastructure. With a cloud provisioning model in place, they can scale down and up.
2. *Costs Savings:* When used correctly, cloud provisioning allows organizations to spend less on managing and maintaining IT systems. Rather than spending vast amounts on expensive systems and equipment, companies get only the resources they need. Some of the pointers that make cloud provisioning an excellent option for cost savings include the following:
    - There are fewer employees required, leading to more cost-cutting.
    - The duty of upgrading cloud provisioning software and hardware falls to the cloud service providers.
    - With limited IT hardware around, organizations consume less energy.
    - Employees execute projects faster and spend more of their company time working on productive things.

Challenges of Cloud Provisioning

Provisioning in cloud computing presents many challenges in any organization. These include:

1. *Complex Management and Monitoring:* Organizations may have to rely on multiple provisioning tools to customize how they use cloud resources and services. Most companies even have to deploy workloads on more than one cloud platform. This makes it more challenging to have one central console where they can manage everything.

2. *Service Enforcement:* A self-service provisioning structure is ideal for streamlining how users solicit and control cloud resources. It, however, needs strict rules to ensure that no one provisions resources they are not supposed to.
3. *Service and Resource Dependencies:* Cloud workloads and applications frequently tap into essential cloud infrastructure services like storage, networking, and computing. Besides, public cloud providers' most significant selling point is higher-level ancillary services such as big-data capabilities, machine learning, and serverless functions. These could cause unanticipated overuse and unforeseen costs.
4. *Cost Controls:* Other than provisioning policies, automated alerts and monitoring about pricing thresholds and usage are important. Such alerts are not always delivered on time. A warning about an upcoming cloud service budget overrun can arrive after a few hours or days.

Data-Intensive Technologies for Cloud Computing

Data-intensive computing is concerned with production, manipulation, and analysis of large-scale data in the range of hundreds of megabytes (MB) to petabytes (PB) and beyond.

The rapid growth of the Internet and World Wide Web led to vast amounts of information available online. In addition, business and government organizations create large amounts of both structured and unstructured information, which need to be processed, analyzed, and linked.

The storing, managing, accessing, and processing of this vast amount of data represents a fundamental need and an immense challenge in order to satisfy needs to search, analyze, mine, and visualize this data as information. Data-intensive computing is intended to address this need.

Parallel processing approaches can be generally classified as either compute-intensive, or data-intensive.

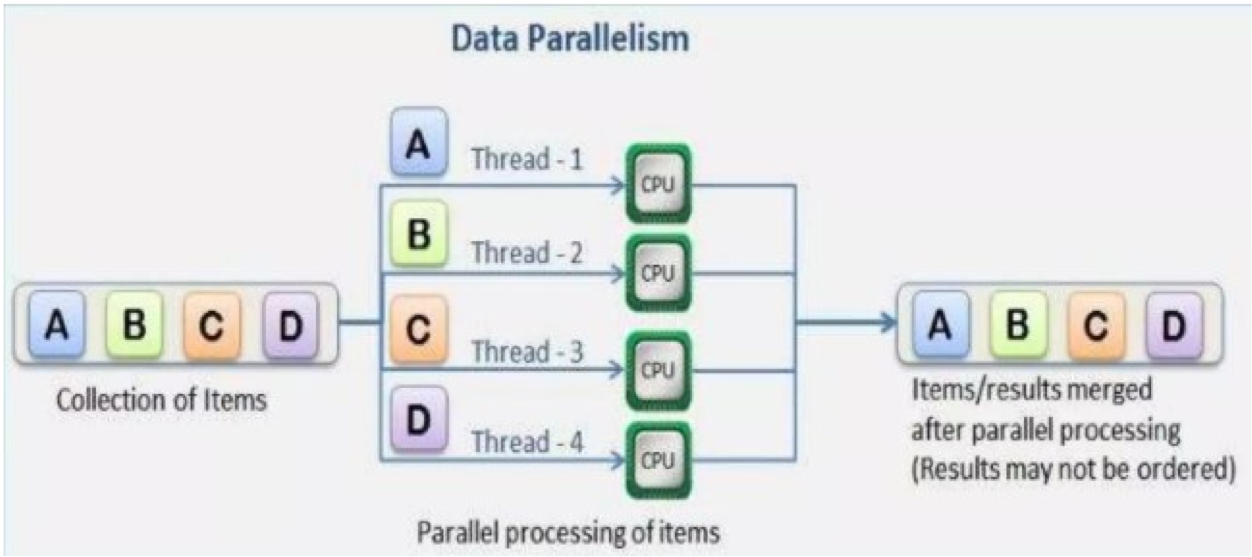
Compute-intensive is used to describe application programs that are compute-bound. Such applications devote most of their execution time to computational requirements as opposed to I/O, and typically require small volumes of data. While Parallel processing of data-intensive applications typically involves parallelizing individual algorithms within an application process, and decomposing the overall application process into separate tasks, which can then be executed in parallel on an appropriate computing platform to achieve overall higher performance than serial processing.

Processing Approach

Data-intensive is used to describe applications that are I/O bound or with a need to process large volumes of data. Such applications devote most of their processing time to I/O and movement and manipulation of data.

Parallel processing of data-intensive applications typically involves partitioning or subdividing the data into multiple segments which can be processed independently using the same executable application program in parallel on an appropriate computing platform, then reassembling the results to produce the completed output data.

The fundamental challenges for data-intensive computing are managing and processing exponentially growing data volumes, significantly reducing associated data analysis cycles to support practical, timely applications, and developing new algorithms which can scale to search and process massive amounts of data.



System Architecture

A variety of system architectures have been implemented for data-intensive computing and large-scale data analysis applications. A number of solutions have come out, one among them is MapReduce concept which is developed by Google and available as open-source implementation known as Hadoop. This project is used by Yahoo, Facebook and others. LexisNexis Risk Solutions also developed and implemented a scalable platform for data-intensive computing which is used by LexisNexis.

MapReduce

The MapReduce architecture and programming model is an example for data-intensive computing, pioneered by Google.

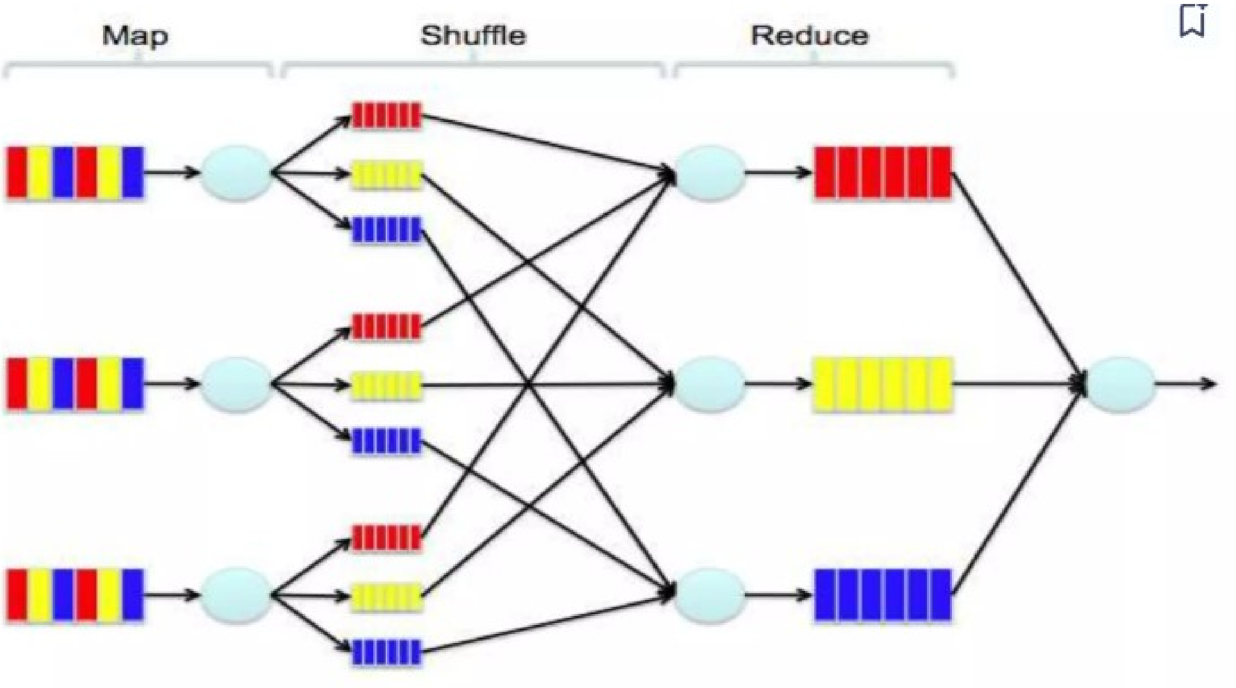
The MapReduce architecture allows programmers to use a functional programming style to create a map function that processes a key–value pair associated with the input data to generate a set of intermediate key–value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key.

It takes a set of input key–value pairs associated with the input data and produces a set of output key–value pairs.

System takes care of particulars like partitioning the input data, scheduling and executing automatically. Hence programmers who do not have experience in parallel programming can simply use a large distributed processing environment without any problem.

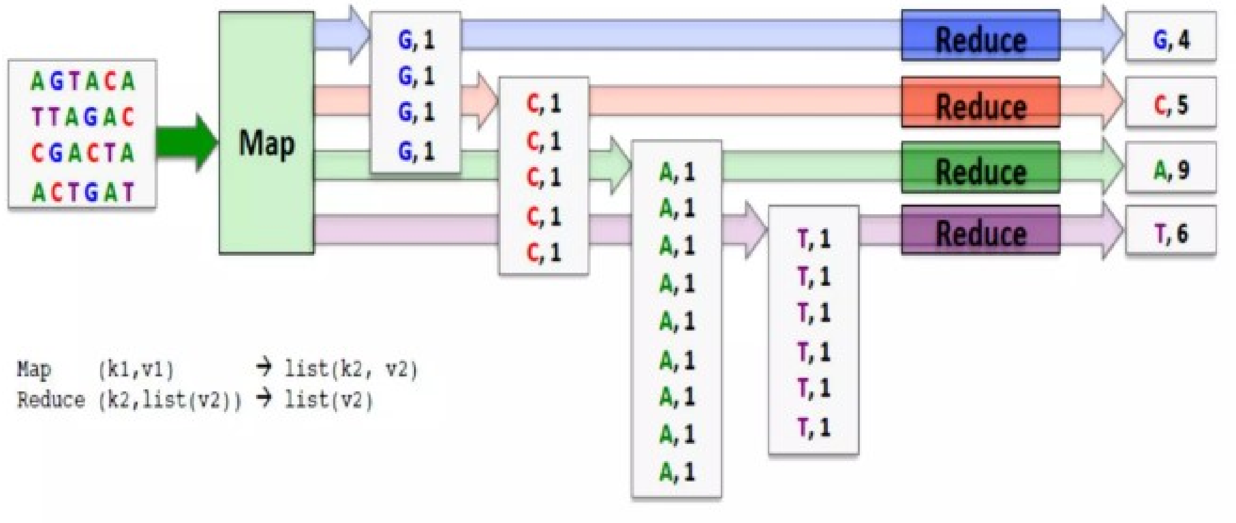
In the Map phase, the input data is partitioned into input splits and assigned to Map tasks associated with processing nodes in the cluster.

The Map task typically executes on the same node containing its assigned partition of data in the cluster. These Map tasks perform user-specified computations on each input key–value pair from the partition of input data assigned to the task, and generates a set of intermediate results for each key.



The shuffle and sort phase then takes the intermediate data generated by each Map task, sorts this data with intermediate data from other nodes, divides this data into regions to be processed by the reduce tasks, and distributes this data as needed to nodes where the Reduce tasks will execute.

The Reduce tasks perform additional user-specified operations on the intermediate data possibly merging values associated with a key to a smaller set of values to produce the output data. For more complex data processing procedures, multiple MapReduce calls may be linked together in sequence.



```
Map    (k1,v1)       → list(k2, v2)
Reduce (k2,list(v2)) → list(v2)
```

Hadoop

Apache Hadoop is an open source software project sponsored by The Apache Software Foundation which implements the MapReduce architecture.

Hadoop now encompasses multiple subprojects in addition to the base core, MapReduce, and HDFS distributed filesystem. These additional subprojects provide enhanced application processing capabilities to the base Hadoop implementation and currently include Avro, Pig, HBase, ZooKeeper, Hive, and Chukwa.

The Hadoop MapReduce architecture is functionally similar to the Google implementation except that the base programming language for Hadoop is Java instead of C++. The implementation is intended to execute on clusters of commodity processors.

Hadoop implements a distributed data processing scheduling and execution environment and framework for MapReduce jobs.

Hadoop includes a distributed file system called HDFS which is analogous to GFS in the Google MapReduce implementation.

The Hadoop execution environment supports additional distributed data processing capabilities which are designed to run using the Hadoop MapReduce architecture.

These include

i. HBase, a distributed column-oriented database which provides random access read/write capabilities;

ii. Hive, which is a data warehouse system built on top of Hadoop that provides SQL-like query capabilities for data summarization, ad hoc queries, and analysis of large datasets; and

iii. Pig – a high-level data-flow programming language and execution framework for data-intensive computing.

HPCC

HPCC (High-Performance Computing Cluster) was developed and implemented by LexisNexis Risk Solutions.

The HPCC approach also utilizes commodity clusters of hardware running the Linux operating system. Custom system software and middleware components were developed and layered on the base Linux operating system to provide the execution environment and distributed filesystem support required for data-intensive computing.

LexisNexis also implemented a new high-level language for data-intensive computing.

The ECL programming language is a high-level, declarative, data-centric, implicitly parallel language that allows the programmer to define what the data processing result should be and the dataflows and transformations that are necessary to achieve the result.

The ECL language includes extensive capabilities for data definition, filtering, data management, and data transformation, and provides an extensive set of built-in functions to operate on records in datasets which can include user-defined transformation functions.

ECL programs are compiled into optimized C++ source code, which is subsequently compiled into executable code and distributed to the nodes of a processing cluster.

Cloud Storage from LANs to WANs

Data management applications are promising for candidates who opt for deployment of the cloud. This is because an on-premises enterprise database system usually comes with a large, occasionally prohibitive up-front cost, both in hardware and in software.

For multiple businesses, the pay-as-you-go cloud computing form is very attractive. Thus, cloud computing is reminiscent of the Application Service Provider (ASP) and Database-as-a-Service (DaaS) paradigms.

Instead of owning, establishing and managing the database programs, cloud computing vendors normally maintain little more than the hardware and give their clients a set of virtual appliances to establish their own software.

Resource accessibility is normally elastic, with an apparently infinite allowance of compute power and storage accessible on demand, in a pay-only-for-what-you-use model.

Cloud Characteristic

There are three characteristics of a cloud computing natural environment that are most pertinent to be considered before choosing storage in cloud.

1. Computer power is elastic, when it can perform parallel operations. In general, applications conceived to run on the peak of a shared-nothing architecture are well matched for such an environment.
2. Data is retained at an unknown host server. In general, letting go off data is a threat to many security issues and thus suitable precautions should be taken.
3. Data is duplicated often over distant locations. Data accessibility and durability is paramount for cloud storage providers, as data tampering can be impairing for both the business and the organization's reputation. Data accessibility and durability are normally accomplished through hidden replications.

Distributed Data Storage

A distributed data store is a computer network where information is stored on more than one node, often in a replicated fashion. It is usually specifically used to refer to either a distributed database where users store information on a number of nodes or a computer network in which users store information on a number of peer network nodes.

There are some reasons for distributed storage means to be favored over traditional relational database systems encompassing scalability, accessibility and performance. The new generation of applications requires processing of data to a tune of terabytes and even peta bytes. This is accomplished by distributed services. Distributed services means distributed data. This is a distinct giant compared to traditional relational database systems.

Emerging examples of distributed data storage are Amazon Dynamo, CouchDB and ThruDB.

Amazon Dynamo

Amazon DynamoDB is a fully managed NoSQL database service that allows to create database tables that can store and retrieve any amount of data. It automatically manages the data traffic of tables over multiple servers and maintains performance.

It also relieves the customers from the burden of operating and scaling a distributed database. Hence, hardware provisioning, setup, configuration, replication, software patching, cluster scaling, etc. is managed by Amazon.

With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

DynamoDB allows you to delete expired items from tables automatically to help you reduce storage usage and the cost of storing data that is no longer relevant.

Benefits of Amazon DynamoDB

- Managed service − Amazon DynamoDB is a managed service. There is no need to hire experts to manage NoSQL installation. Developers need not worry about setting up, configuring a distributed database cluster, managing ongoing cluster operations, etc. It handles all the complexities of scaling, partitions and re-partitions data over more machine resources to meet I/O performance requirements.

- Scalable − Amazon DynamoDB is designed to scale. There is no need to worry about predefined limits to the amount of data each table can store. Any amount of data can be stored and retrieved. DynamoDB will spread automatically with the amount of data stored as the table grows.

- Fast − Amazon DynamoDB provides high throughput at very low latency. As datasets grow, latencies remain stable due to the distributed nature of DynamoDB's data placement and request routing algorithms.

- Durable and highly available − Amazon DynamoDB replicates data over at least 3 different data centers' results. The system operates and serves data even under various failure conditions.

- Flexible: Amazon DynamoDB allows creation of dynamic tables, i.e. the table can have any number of attributes, including multi-valued attributes.

- Cost-effective: Payment is for what we use without any minimum charges. Its pricing structure is simple and easy to calculate.

CouchDB

Apache CouchDB is an open source NoSQL document database that collects and stores data in JSON-based document formats. Unlike relational databases, CouchDB uses a schema-free data model, which simplifies record management across various computing devices, mobile phones, and web browsers.

It is an open-source database that uses various different formats and protocols to store, transfer, and process its data. It uses JSON to store data, JavaScript as its query language using MapReduce, and HTTP for an API.

Documents are the primary unit of data in CouchDB and they also include metadata. Document fields are uniquely named and contain values of varying types and there is no set limit to text size or element count.

CouchDB aspires to persuade the Four Pillars of Data Management by these methods:

1. Save: ACID compliant, save efficiently
2. See: Easy retrieval, straightforward describing procedures, fulltext search
3. Secure: Strong compartmentalization, ACL, connections over SSL
4. Share: Distributed means

CouchDB actually has no apparent authentication scheme, i.e., it is in-built. The replication is distributed. A server can revise others once the server is made offline and data is changed. If there are confrontations, CouchDB will choose a survivor and hold that as latest. Users can manually suspend this surviving alternative later. Importantly, the confrontation tenacity yields identical results comprehensively double-checking on the offline revisions. This also promises to compose a storage motor for MySQL founded on CouchDB.

Features of CouchDB

Features of CouchDB includes the following:

1. Replication: It provides the simplest form of replication and no other database is so simple to replicate.
2. Document Storage: It is a NoSQL database that follows document storage where each field is uniquely named and contains values of various data types such as text, number, Boolean, lists, etc.
3. ACID Properties: The CouchDB file layout follows all the features of ACID properties.
4. Security: It also provides database-level security and the permissions are divided into readers and admins where readers can do both the read and write to the database.
5. Map/Reduce: The main reason for the popularity of CouchDB is a map/reduce system.
6. Authentication: CouchDB facilitates you to keep authentication open via a session cookie-like a web application.

7. Built for Offline: CouchDB can replicate to devices like smartphones that have a feature to go offline and handle data sync for you when the device is back online.
8. Eventual Consistency: CouchDB guarantees eventual consistency to provide both availability and partition tolerance.
9. HTTP API: All items have a unique URI(Unique Resource Identifier) that gets exposed via HTTP. It uses the HTTP methods like POST, GET, PUT, and DELETE for the four basic CRUD (Create, Read, Update, Delete) operations on all resources.

Advantages of CouchDB

Advantages of CouchDB includes the following:

1. HTTP API is used for easy Communication.
2. It is used to store any type of data.
3. ReduceMap allows optimizing the combining of data.
4. Structure of CouchDB is very simple
5. Fast indexing and retrieval.

Disadvantages of CouchDB

Disadvantages of CouchDB includes the following:

1. CouchDB takes a large space for overhead, which is a major disadvantage as compared to other databases.
2. Arbitrary queries are expensive.
3. There's a bit of extra space overhead with CouchDB compared to most alternatives.
4. Temporary views on huge datasets are very slow.
5. It doesn't support transactions
6. Replication of large databases may fail.

ThruDB

ThruDB is an open source database built on Apache's Thrift framework and is a set of simple services such as scaling, indexing and storage which is used for building and scaling websites.

It provides flexible, fast and easy-to-use services that simplify the management of the modern web data layer and provides developers with features and tools most web developers need. These features can be easily configured or turned off.

ThruDB aspires to be universal in simplifying the administration of the up-to-date WWW data level (indexing, caching, replication, backup) by supplying a reliable set of services:

- Thrucene for indexing
- Throxy for partitioning and burden balancing
- Thrudoc for article storage

Thrift is a structure for effective cross-language data serialization, RPC and server programming. Thrift is a programs library and set of code-generation devices conceived to expedite development and implementation of effective and scalable backend services. Its prime aim is to enhance effective and dependable connection over programming languages. This is finished by abstracting the portions of each dialect that are inclined to need the most customization into a widespread library that is applied in each language. Specifically, Thrift permits developers to characterize data types and service interfaces in a sole language-neutral document and develop all the essential cipher to construct RPC purchasers and servers.

There are some more systems out in the untamed as well as appearing systems. Prominent amidst them are:

- Amazon Simple Storage Service is a straightforward data storage scheme with a hash-table like API. It is a hosted service with interior architecture minutia not available. It is proclaimed that the conceive obligations of S3 are scalable, reliable, fast, inexpensive and simple.
- Amazon SimpleDB is a hosted WWW service for running queries on organized data in real time. It has the prime functionality of a database, real-time lookup and straightforward querying of organized data.
- MemcacheDB is a distributed key-value storage scheme conceived for persistence. It conforms to the memcache protocol. Memcachedb values Berkeley DB as a saving backend, so allotments of characteristics encompassing transaction and replication are supported.

Features

- Multi-master replication
- Built for horizontal scalability
- Incremental backups and redo logging
- Multiple storage back-end client libraries for most languages
- Simple and powerful search API

Services

ThruDB provides web-scale data management by providing these services:

- *Thrucene* - For Lucene-based indexing
- *Throxy* - For partitioning and load balancing
- *Thrudoc* - For document storage
- *Thruqueue* - For a persistent message queue service
- *Thrift* - For cross-language services framework

Compared to SimpleDB, ThruDB removes many of the former's restrictions- no 1024 byte limit per attribute, ability to use proprietary data formats, no query time limits, and more. Since it also runs on EC2 and public Thrift/ThruDB *Amazon Machine Images* (AMIs) exist, ThruDB is in a sense a competitor to SimpleDB.

Thrudb is an attempt to simplify the process of the modern web data layer which comprises indexing, caching, replication & backup. It provides a consistent set of services. Thrucene provides indexing, Throxy provides partitioning and load balancing, and Thrudoc provides document storage.

To get a great overview of the design philosophy and API samples we can refer to Jake's whitepaper. The entire stack is implemented on top of Thrift, which has a great access for easy language interoperability and API access.

Applications Utilizing Cloud Storage

1. Online File Storage

Being capable of accessing documents from any location and from any computer is one of the large conveniences of the Internet. Online document storage has been around for a while now, but the latest generation of services is so simple to use. Most online storage providers moreover give us the proficiency to share these documents with associates and colleagues.

1. DropBox
2. Box.net
3. Live Mesh
4. Oosah
5. Jungle Disk

2. Cloud Storage Companies

It seems like there is a new cloud business popping up every day, and the variety of services they offer are endless. Following are some cloud storage companies that are worth noting. Most of these service providers have a free test or offer some sort of free storage space.

1. Box Cloud
2. Amazon Cloud
3. SugarSync online backup
4. Hubic online storage
5. Google cloud drive

3. Online Book Marking Service

Social bookmarking is the best method in which client, use bookmarks and organize the sheets they wish to recall or share with their friends. These collective bookmarks are generally community-based and can be kept in confidence, only with specific persons or assemblies, distributed only to internal reliable systems, or another grouping of public and individual domains. Only the approved persons can observe these communal bookmarks in succession, by class or tags, or by a search engine.

4. Online Photo Editing Service

Cloud computing is a phase that encounters a wide range of services and programs applications that share one common fact that they all run on the Internet and not on a user's PC. Cloud computing has been around for years: instant messaging and web posted letters are only two examples. New applications and services emerge almost every day in the form of new community networking sites, SaaS (Software as a Service), web posted letters and many others.

Cloud computing contributes numerous valuable and helpful services, and applications and for many of these, it is a flawless venue.

1. Photoshop Express Editor
2. Picnik
3. Splashup
4. FotoFlexer
5. Pixer.us