SAQ

1. Case study of Google distributed file systems.

Google file system is a scalable distributed file system developed by Google to provide efficient and reliable access to data using large clusters of commodity hardware.
- It is designed to meet the rapidly growing demand of Google's data processing need.
- It provides performance, scalability, reliability and availability of data across distributed system for handling and processing big data.

Characteristics of GFS
1. Files are organised hierarchically in directories and identified by path name.
2. It supports all the general operations on files like read, writes, open, delete, etc.

Common goals of GFS
1. Performance
2. Reliability
3. Automation
4. Fault Tolerance
5. Scalability
6. Availability

2. Frame about the various faults and transparencies involved in distributed systems.

There are three main types of faults: transient, intermittent, and permanent

A transient fault is a fault that happens once, and then doesn't ever happen again.An intermittent fault is one that occurs once, seems to go away, and then occurs again! a permanent fault is one that just does not go away after it first occurs.

Types of Transparency in Distributed Systems:

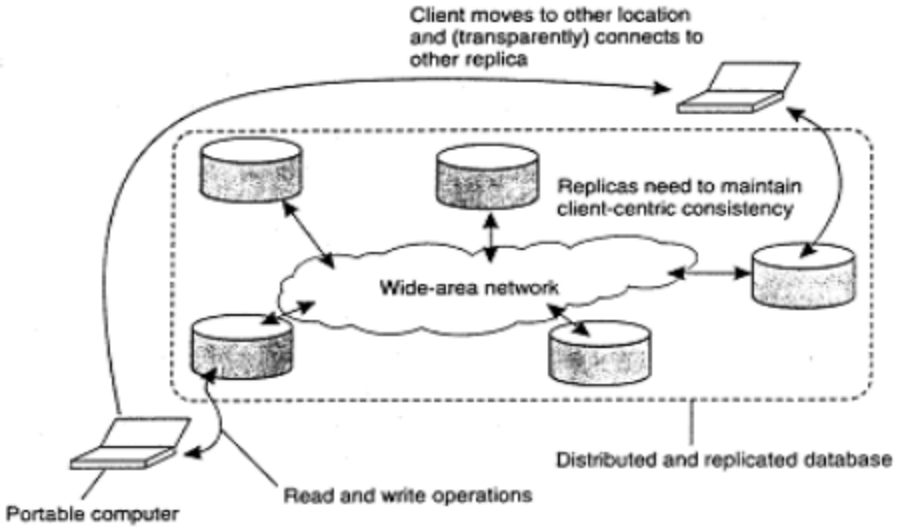
1. Access Transparency: Access Transparency allows the same operations to be used to access local and remote resources.

2. Location Transparency: Location Transparency permits access to resources regardless of their physical or network location.

3. Concurrency Transparency: Concurrency Transparency permits many processes to run in parallel using shared resources without interfering with one another.

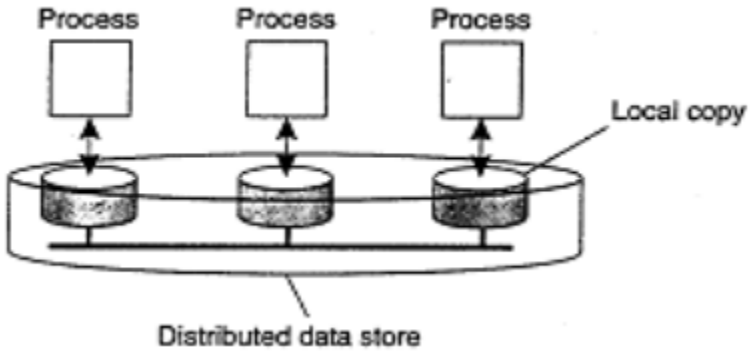
3. Negotiate between Client-Centric and Data-Centric Consistency models.

Client centric model:Client-centric Consistency Model defines how a data-store presents the data value to an individual client when the client process accesses the data value across different replicas. It is generally useful in applications where: one client always updates the data-store.



Client centric model

Data centric model:Data-centric consistency models describe how the replicated data is kept consistent across different data-stores, and what the process can expect from the data-store. Data-centric models are too strict when: most operations are read operations.



Data Centric model

4. List out any four CORBA services.

CORBA SERVICES are

- CORBA Naming Service.
- CORBA Event Service.
- CORBA Notification Service.
- CORBA Security Service.

5. Distinguish between CODA and Andrew File system

CODA	AFS
Coda is a distributed file system developed as a research project at Carnegie Mellon University in 1987	Developed by Carnegie Mellon University as part of Andrew distributed computing environments (in 1986)
Coda uses a local cache to provide access to server data when the network connection is lost.	A research project to create a campus-wide file system.
Continued operation during partial network failures in server network	Public domain implementation is available on Linux (LinuxAFS)
Well-defined semantics of sharing, even in the presence of network failure	AFS is implemented as two software components that exist at UNIX processes called Vice and Venus.

6. List out any three differences between Map Reduce and Pig

Map Reduce	Pig
It is a Data Processing Language.	It is a Data Flow Language.
It converts the job into map-reduce functions.	It converts the query into map-reduce functions.
It is a Low-level Language.	It is a High-level Language

7. What is Grid computing?

Grid computing is a computing infrastructure that combines computer resources spread over different geographical locations to achieve a common goal. All unused resources on multiple computers are pooled together and made available for a single task. Organizations use grid computing to perform large tasks or solve complex problems that are difficult to do on a single computer.

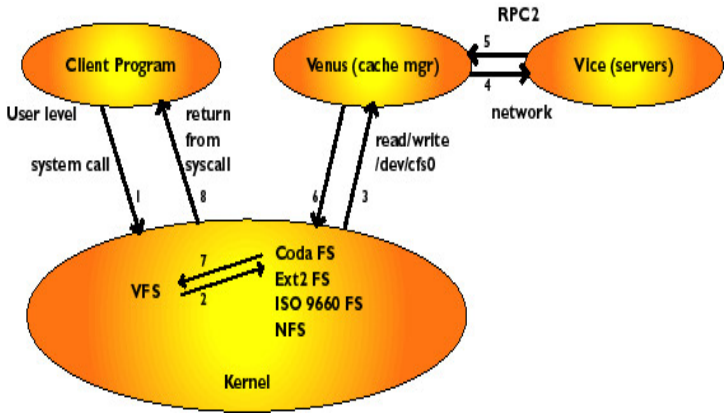
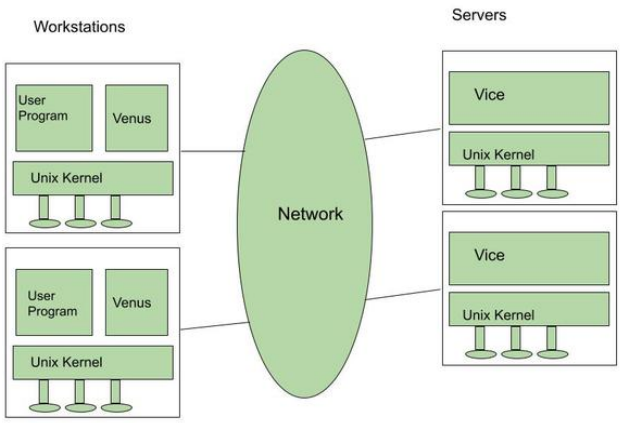
8.Discuss REST and Web Services

REST is a style of architecture with a set of constraints which allows data transfer over a standardized interface, such as HTTP. REST relies on a simple URI to make a request

A web service is a set of functions that are published. to a network for use by other programs. Many people regard web services. as a technology only for publishing software services on the Internet via.

LAQ

1. Differentiate between CODA and Andrew file systems with necessary diagrams.

CODA	AFS
	
Coda is an advanced networked filesystem. It has been developed at CMU since 1987 by the systems group of M. Satyanarayanan. in the SCS department.	AFS is a distributed file system. It uses the client/server model, where all the files are stored on file server machines.

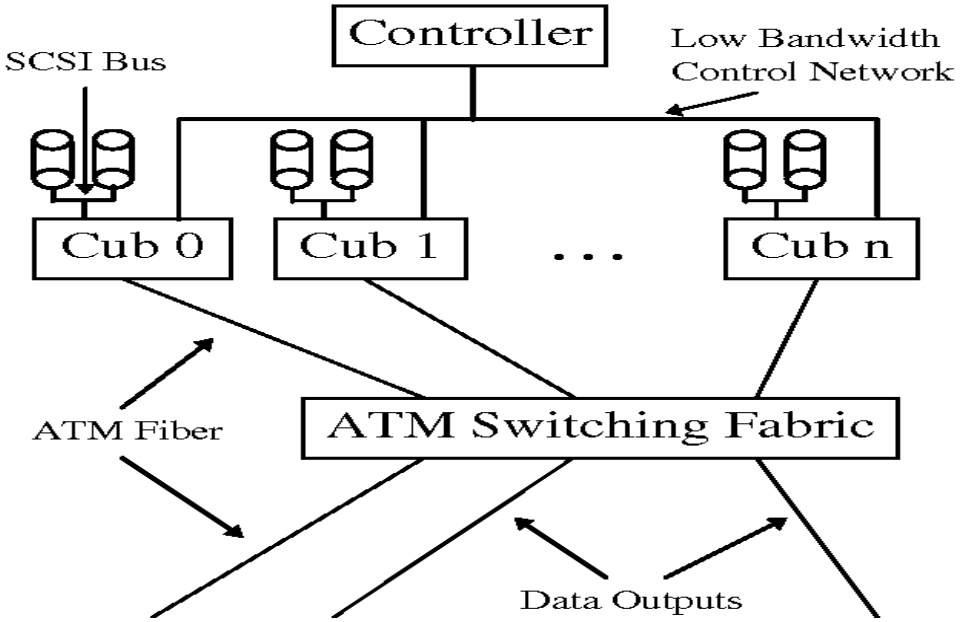
disconnected operation for mobile computing	Files are transferred to client machines as necessary and cached on local disk. The server part of AFS is called the AFS File Server,
is freely available under a liberal license	Its efficiency can be attributed to the following practical assumptions (as also seen in UNIX file system)
high performance through client side persistent caching	AFS uses the Weak Consistency model
server replication	AFS volumes can be replicated to read-only cloned copies.
security model for authentication, encryption and access control	Volumes are created by administrators and linked at a specific named path in an AFS cell.
continued operation during partial network failures in server network	Read and write operations on an open file are directed only to the locally cached copy.
network bandwith adaptation	a client system will retrieve data from a particular read-only copy. If at some point, that copy becomes unavailable
good scalability well defined semantics of sharing, even in the presence of network failures	It only contains temporary files needed for workstation initialization and symbolic links to files in the shared name space.

Fault tolerance refers to the ability of a system (computer, network, cloud cluster, etc.) to continue operating without interruption when one or more of its components fail.

2. Summarize briefly about the following with neat diagrams:

i) Case study of Tiger file server

Tiger is a distributed, fault-tolerant real-time fileserver. It provides data streams at a constant, guaranteed rate to a large number of clients, in addition to supporting more traditional filesystem operations.



It is intended to be the basis for multimedia (video on demand) fileservers, but may also be used in other applications needing constant rate data delivery. The fundamental problem addressed by the Tiger design is that of efficiently balancing user load against limited disk, network and I/O bus resources. Tiger accomplishes this balancing by striping file data across all disks and all computers in the (distributed) system, and then allocating data streams in a schedule that rotates across the disks. This paper describes the Tiger design and an implementation that runs on a collection of personal computers connected by an ATM switch.

ii) Characteristics of multimedia data.

Multimedia applications are characterized by three features: the combination of content (information) and software (how to control the presentation of the information) ; the possibility of many kinds of media for presenting information (text, graphic, images, sound, video), and the possibility of presenting information …Very High Processing Power: To deal with large amount of data, very high processing power is used.

File System:

File system must be efficient to meet the requirements of continuous media. These media files requires very high-disk bandwidth rates. Disks usually have low transfer rates and high latency rates.

File formats that support multimedia:

Multimedia data consists of a variety of media formats or file representation including ,JPEG, MPEG, AVI, MID, WAV, DOC, GIF,PNG, etc. AVI files can contain both audio and video data in a file container that allows synchronous audio-with-video playback.

Input/Output:

In multimedia applications, the input and output should be continuous and fast. Real-time recording as well as playback of data are common in most of the multimedia applications which need efficient I/O.

Operating System:

The operating system must provide a fast response time for interactive applications. High throughput for batch applications, and real-time scheduling,

Storage and Memory:

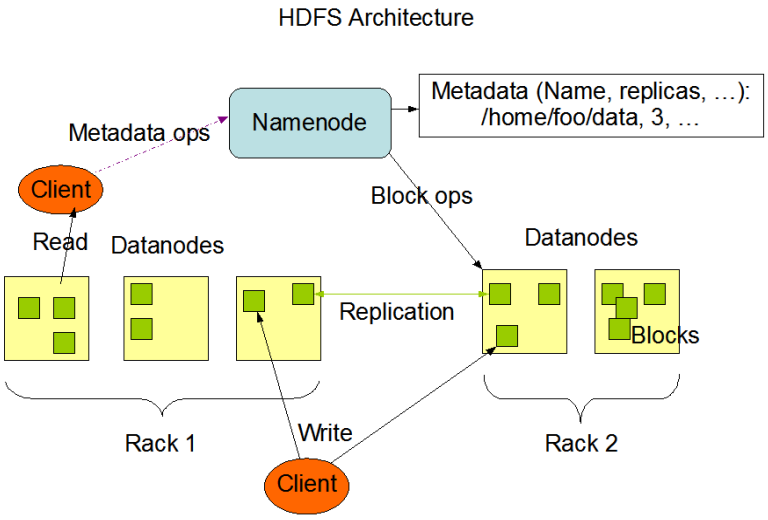
Multimedia systems require storage for large capacity objects such as video, audio, animation and images. Depending on the compression scheme and reliability video and audio require large amount of memory.

Network Support:

It includes internet, intranet, LAN, WAN, ATM, Mobile telephony and others. In recent years, there has been a tremendous growth of multimedia applications on the internet like streaming video, IP telephony, interactive games, teleconferencing, virtual world, distance learning and so on.

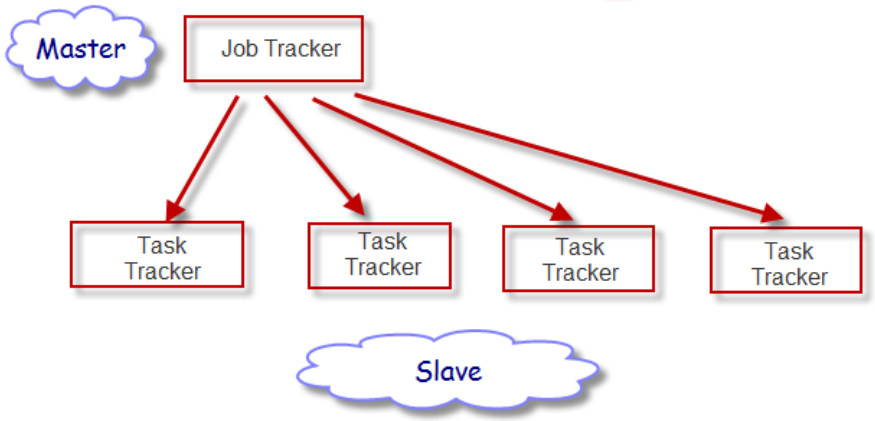
3.Justify the role of the Name node, Job Tracker and Test tracker services of HDFS using a neat diagram.

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept.
The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients.



Job Tracker

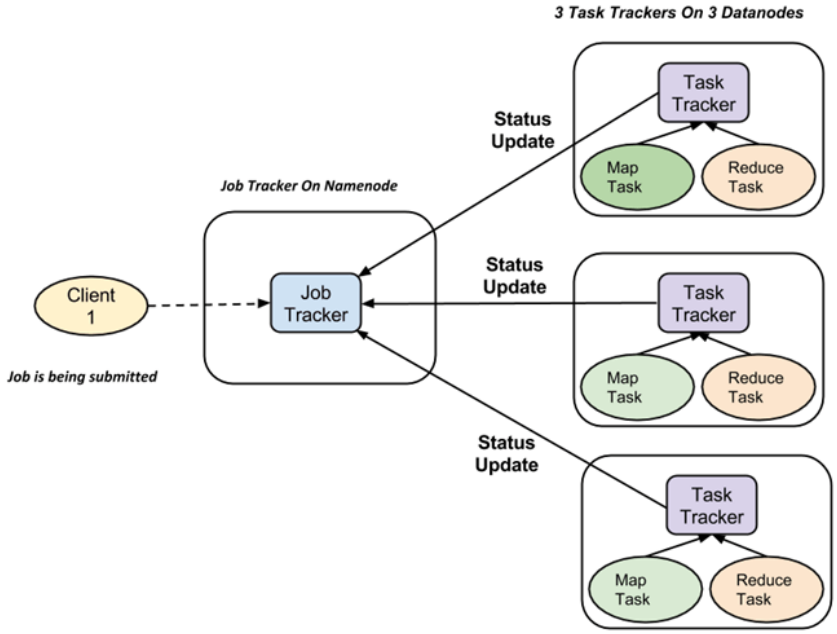
. Client applications (2) submit jobs to the Job tracker. The JobTracker submits the work to the chosen TaskTracker nodes The JobTracker submits the work to the chosen TaskTracker nodes. The TaskTracker nodes are (3) monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker.



(1) Job tracker's role is resource management, tracking resource availability and tracking the progress of fault tolerance.

Job tracker communicates with the Namenode to determine the location of data. Finds the task tracker nodes to execute the task on given nodes. It tracks the execution of MapReduce from local to the Slave

A TaskTracker is a node in the cluster that accepts tasks - Map, Reduce and Shuffle operations - from a JobTracker. Every TaskTracker is configured with a set of slots, these indicate the number of tasks that it can accept



here are the five core **Hadoop services**:
namenode, secondary namenode, datanode, jobtracker, and tasktracker, each a separate daemon. Services are run on nodes of the cluster

4 i) Write short notes on Distributed-coordination based systems.

In distributed coordination-based systems, the focus is on how coordination between the processes takes place.that assume that the various components of a system are inherently distributed and that the real problem in developing such systems lies in coordinating the activities of different components. In other words, instead of concentrating on the transparent distribution of components, emphasis lies on the coordination of activities between those components.

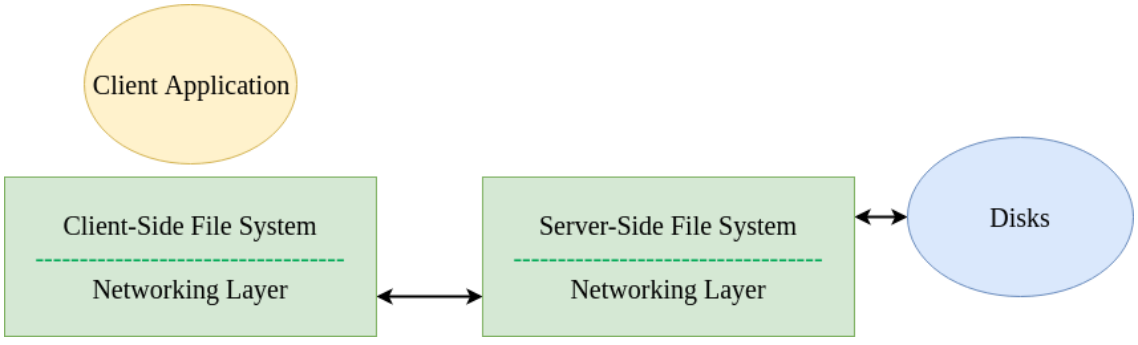
coordination-based systems is the clean separation between computation and coordination. If we view a distributed system as a collection of (possibly multithreaded) processes, then the computing part of a distributed system is formed by the processes, each concerned with a specific computational activity, which in principle, is carried out independently from the activities of other processes.

ii) Describe briefly about Trends in Distributed systems.

- Trends in Distributed systems
- Ethernet - the name given to the popular local area packet-switched network technology invented by Xerox PARC. ...
- FDDI - the Fiber Distributed Data Interface. ...
- HiPPI - the high-performance parallel interface. ...
- SONET - Synchronous Optical Network. ...
- ATM - Asynchronous Transfer Mode.

5.Explain distributed file systems? Explain SUN NFS in detail

A distributed file system (DFS) is a file system that spans across multiple file servers or multiple locations, such as file servers that are situated in different physical places. Files are accessible just as if they were stored locally, from any device and from anywhere on the network.



The architecture consists of a client-side file system and a server-side file system. A client application issues a system call (e.g. read(), write(), open(), close() etc.) to access files on the client-side file system, which in turn retrieves files from the server

The main reason enterprises choose a DFS is to provide access to the same data from multiple locations. For example, you might have a team distributed all over the world, but they have to be able to access the same files to collaborate.

The Sun Network Filesystem (NFS) provides transparent, remote access to filesystems. Unlike many other remote filesystem implementations under UNIX, NFS is designed to be easily portable to other operating systems and machine architectures.

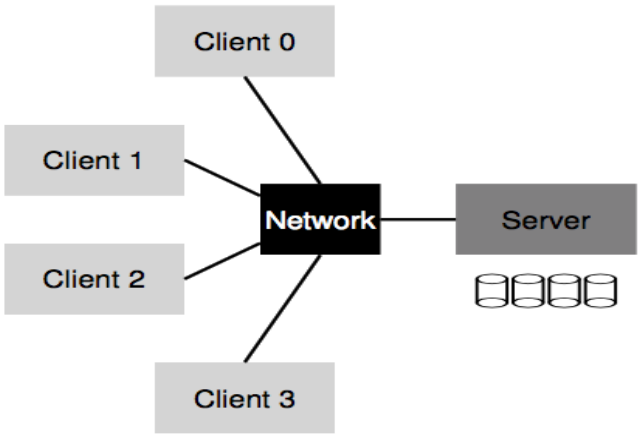


Figure 49.1: **A Generic Client/Server System**

The need for distributed file systems: The primary motivation for a distributed file system is sharing, i.e., the ability to access a file from multiple machines called clients. These clients access the files that are stored on one (or a few) servers.

Another advantage of storing files on a centralized server (as opposed to on each individual client) is the ease of administration. Administration includes things like backing up the data, enforcing quota across users, and security (the chapter notes security as a different aspect than administration, but I consider it to be a subset of administration in general)

NFS is an abbreviation of the Network File System. It is a protocol of a distributed file system. This protocol was developed by the Sun Microsystems in the year of 1984.It is an architecture of the client/server, which contains a client program, server program, and a protocol that helps for communication between the client and server.

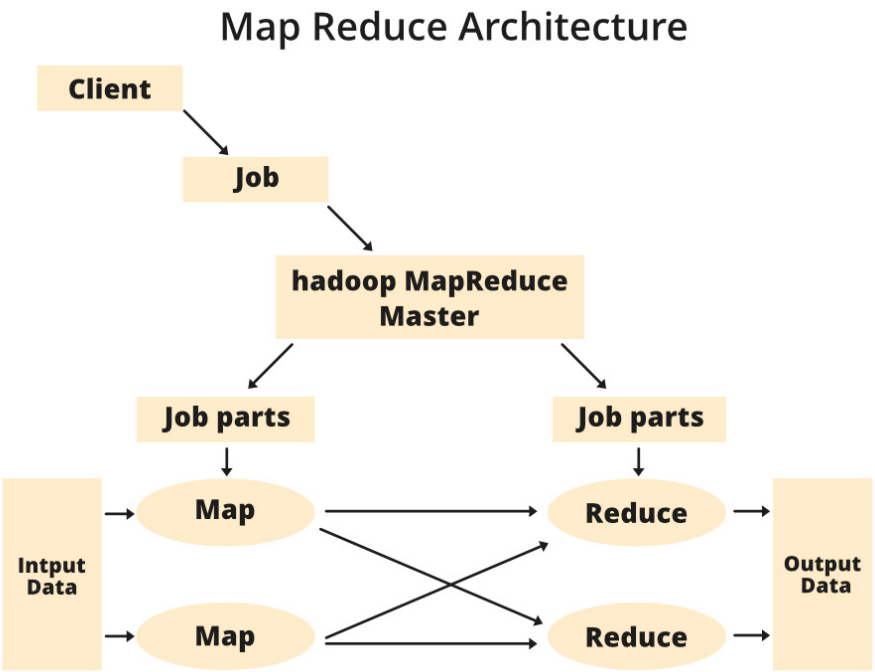
It is that protocol which allows the users to access the data and files remotely over the network. Any user can easily implement the NFS protocol because it is an open standard.

This protocol is mainly implemented on those computing environments where the centralized management of resources and data is critical. It uses the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) for accessing and delivering the data and files.

Network File System is a protocol that works on all the networks of IP-based. It is implemented in that client/server application in which the server of NFS manages the authorization, authentication, and clients. This protocol is used with Apple Mac OS, Unix, and Unix-like operating systems such as Solaris, Linux, FreeBSD, AIX.

6.Discuss the Map Reduce Model?

MapReduce is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster. As the processing component, MapReduce is the heart of Apache Hadoop.



The term "MapReduce" refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.

MapReduce programming offers several benefits to help you gain valuable insights from your big data:

- Scalability. Businesses can process petabytes of data stored in the Hadoop Distributed File System (HDFS).
- Flexibility. Hadoop enables easier access to multiple sources of data and multiple types of data.
- Speed. With parallel processing and minimal data movement, Hadoop offers fast processing of massive amounts of data.
- Simple. Developers can write code in a choice of languages, including Java, C++ and Python.

Generally MapReduce paradigm is based on sending the computer to where the data resides!

MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

- Map stage – The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
- Reduce stage – This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

Shuffle Stage: Shuffle phase in Hadoop transfers the map output from Mapper to a Reducer in MapReduce.

7.Explain about Two-phase commit Protocol

Two-Phase Commit Protocol: This protocol we have multiple distributed databases which are operated from different servers(sites) let's say S1, S2, S3, ...Sn. Where every Si made to maintains a separate log record of all corresponding activities and the transition T has also been divided into the subtransactions T1, T2, T3, ..., Tn and each Ti are assigned to Si. This all maintains by a separate transaction manager at each Si. We assigned anyone site as a Coordinator.

Phase-1st–

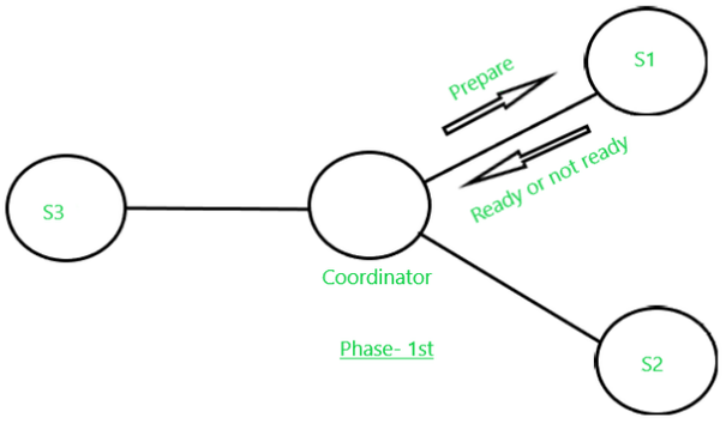
a) Firstly, the coordinator(Ci) places a log record <Prepare T> on the log record at its site.

b) Then, the coordinator(Ci) sends a Prepare T message to all the sites where the transaction(T) executed.

c) Transaction manager at each site on receiving this message Prepare T decides whether to commit or abort its component(portion) of T. The site can delay if the component has not yet completed its activity, but must eventually send a response.

d) If the site doesn't want to commit, so it must write on log record <no T>, and local Transaction manager sends a message abort T to Ci.

e) If the site wants to commit, it must write on log record <ready T>, and local Transaction manager sends a message ready T to Ci. Once the ready T message at Ci is sent nothing can prevent it to commit its portion of transaction T except Coordinator(Ci).



Messaging in Phase- 1st

Phase- 2nd–

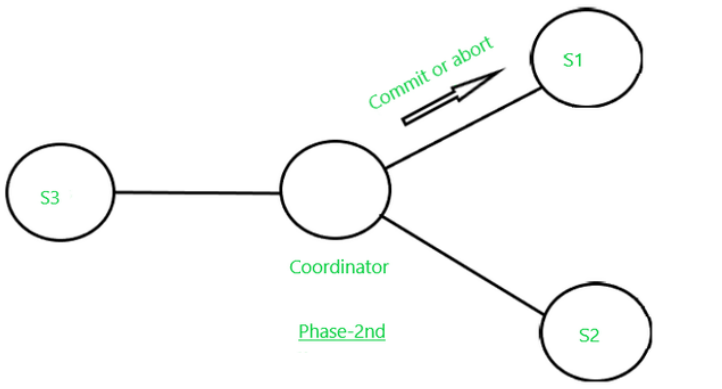
The Second phase started as the response abort T or commit T receives by the coordinator(Ci) from all the sites that are collaboratively executing the transaction T. However, it is possible that some site fails to respond; it may be down, or it has been disconnected by the network. In that case, after a suitable timeout period will be given, after that time it will treat the site as if it had sent abort T. The fate of the transaction depends upon the following points:

a) If the coordinator receives ready T from all the participating sites of T, then it decides to commit T. Then, the coordinator writes on its site log record <Commit T> and sends a message commit T to all the sites involved in T.

b) If a site receives a commit T message, it commits the component of T at that site, and write it in log records <Commit T>.

c) If a site receives the message abort T, it aborts T and writes the log record <Abort T>.

d) However, if the coordinator has received abort T from one or more sites, it logs <Abort T> at its site and then sends abort T messages to all sites involved in transaction T.

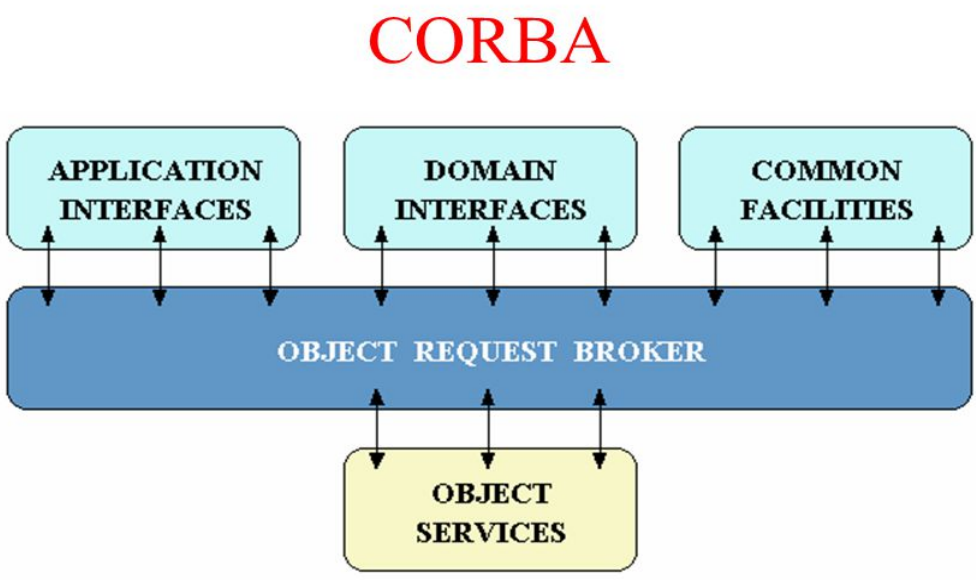


Messaging in Phase- 2nd

Disadvantages:

a) The major disadvantage of the Two-phase commit protocol is faced when the Coordinator site failure may result in blocking, so a decision either to commit or abort Transaction(T) may have to be postponed until coordinator recovers.

8.Illustrate the architecture of CORBA and its services in detail.



OMG Reference Model architecture

CORBA The Common Object Request Broker Architecture (CORBA) is an open distributed object computing infrastructure being standardized by the Object Management Group (OMG).CORBAOMG CORBA automates many common network programming tasks such as object registration, location, and activation; request demultiplexing; framing and error-handling; parameter marshalling and demarshalling; and operation dispatching.

CORBA OMG Reference Model architecture

CORBA Object Services - These are domain-independent interfaces that are used by many distributed object programs. For example, a service providing for the discovery of other available services is almost always necessary regardless of the application domain. Two examples of Object Services that fulfill this role are: –The Naming Service - which allows clients to find objects based on names; –The Trading Service - which allows clients to find objects based on their properties. There are also Object Service specifications for lifecycle management, security, transactions, and event notification, as well as many others.

CORBA Common Facilities - Like Object Service interfaces, these interfaces are also horizontally- oriented, but unlike Object Services they are oriented towards end-user applications. –An example of such a facility is the Distributed Document Component Facility (DDCF), a compound document Common Facility based on OpenDoc.

CORBA Domain Interfaces - These interfaces fill roles similar to Object Services and Common Facilities but are oriented towards specific application domains. –For example, one of the first OMG RFPs issued for Domain Interfaces is for Product Data Management (PDM) Enablers for the manufacturing domain.

CORBA Application Interfaces - These are interfaces developed specifically for a given application. –Because they are application-specific, and because the OMG does not develop applications (only specifications), these interfaces are not standardized.

Some examples of these generic services are:

Naming and trading services that allow objects to refer to and discover other objects on the network. The naming service is a directory service that allows objects to be named and discovered by other objects. This is like the white pages of a phone directory. The trading services are like the yellow pages. Objects can find out what other objects have registered with the trader service and can access the specification of these objects.

Notification services that allow objects to notify other objects that some event has occurred. Objects may register their interest in a particular event with the service and, when that event occurs, they are automatically notified.

Transaction services that support atomic transactions and rollback on failure. Transactions are a fault-tolerance facility that support recovery from errors during an update operation. If an object update operation fails then the object state can be rolled back to its state before the update was started.

-Vinaykumarshivakoti