Nasigari Jitesh
2451-18-733-082

DS Assignment -2

1. List out any four CORBA services?

A. i. Naming Services.

ii. Event service and notification services.

iii. Security services.

iv. Trading services.

2. What are the useful requirements for distributed systems being fault tolerant?

A. Fault tolerance system is a vital issue in distributed computing; it keeps the system in a working condition subject to failure. For a system to be fault tolerant, it is related to dependable systems. Dependability covers some useful requirements in the fault tolerance system these requirements include: Availability, Reliability, Safety, and Maintainability.

Availability: This is when a system is in a ready state, and is ready to deliver its functions to its corresponding users. Highly available systems work at a given instant in time.

Reliability: This is the ability for a computer system to run continuously without a failure. Unlike availability, reliability is defined in a time interval instead of an instant in time. A highly reliable system works constantly for a long period of time without interruption.

Safety: This is when a system fails to carry out its corresponding processes correctly and its operations are incorrect, but no shattering event happens.

Maintainability: A highly maintainability system can also show a great measurement of accessibility, especially if the corresponding failures can be noticed and fixed mechanically.

Nasigari Jitesh
2451-18-733-082

3. What are the different types of failure models in distributed systems?

A. Figure 8-1. Diferent types of failures.

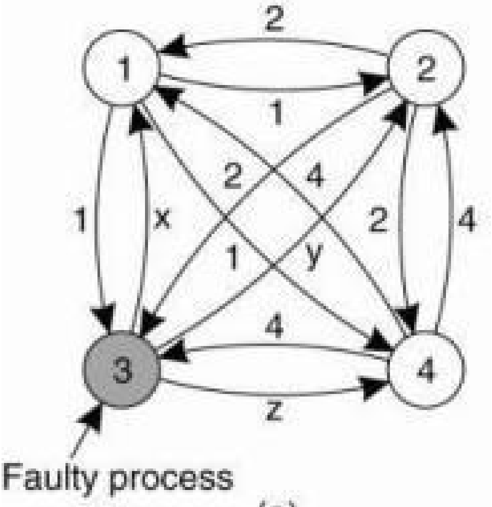
S.NO	Type of failure	Description
1	Crash failure	A server halts, but is working correctly until it halts
2	Omission failure	A server fails to respond to incoming requests
a	Receive omission	A server fails to receive incoming messages
b	Send omission	A server fails to send messages
3	Timing failure	A server's response lies outside the specifed time interval
4	Response failure	A server's response is incorrect
a	Value failure	The value of the response is wrong
b	State transition failure	The server deviates from the correct fow of control
5	Arbitrary failure	A server may produce arbitrary responses at arbitrary times

4. Explain about Byzantine agreement problem with 3 non faulty processes and one faulty process?

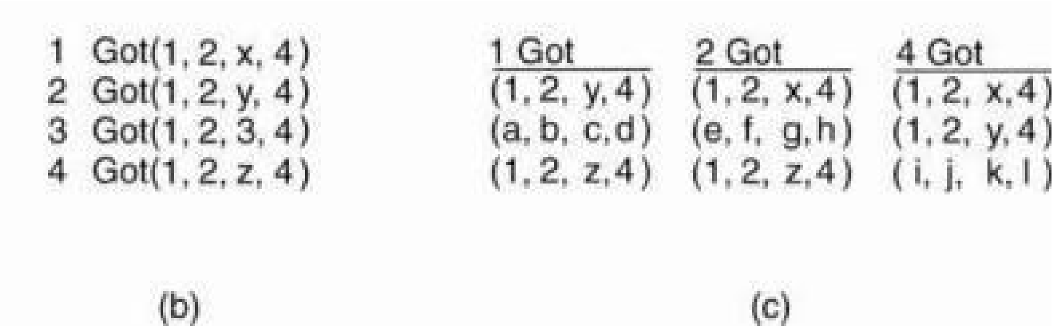
A. One of the fundamental problems in fault tolerant distributed computing is the Byzantine agreement problem. Byzantine agreement requires a set of parties in a distributed environment to agree on a value even if some of the parties are corrupted.

The Byzantine agreement problem for three non faulty and one faulty process. (a) Each process sends their value to the others. (b) The vectors that each process assembles based on (a). (c) The vectors that each process receives in step 3.

We illustrate the working of the algorithm for the case of N = 4 and k = 1. For these parameters, the algorithm operates in four steps.



(a)



1 Got(1, 2, x, 4)
2 Got(1, 2, y, 4)
3 Got(1, 2, 3, 4)
4 Got(1, 2, z, 4)

(b)

1 Got	2 Got	4 Got
(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x, 4)
(a, b, c, d)	(e, f, g, h)	(1, 2, y, 4)
(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k, l)

(c)

Step-1: every non faulty process i sends vi to every other process using reliable unicasting. Faulty processes may send anything. Moreover, because we are using multicasting, they may send different values to different processes. Let vi =i. In Fig (a) we see that process 1 reports 1, process 2 reports 2, process 3 lies to everyone, giving x, y, and z, respectively, and process 4 reports a value of 4.

Step-2: the results of the announcements of step 1 are collected together in the form of the vectors of Fig (b).

Step-3: consists of every process passing its vector from Fig. 8-5(b) to every other process. In this way, every process gets three vectors, one from every other process. Here, too, process 3 lies, inventing 12 new values, a through l. The results of step 3 are shown in Fig. 8-5(c). Step-4: each process examines the ith element of each of the newly received vectors. If any value has a majority, that value is put into the result vector. If no value has a majority, the corresponding element of the result vector is marked UNKNOWN. From Fig. 8-5(c) we see that 1, 2, and 4 all come to agreement on the values for v1, v2, and v4, which is the correct result.

The goal of Byzantine agreement is that consensus is reached on the value for the non faulty processes only.

5. Explain about RPC's semantics in the presence of failures?

A. The goal of Remote Procedure Calls(RPC) is to hide communication by making remote procedure calls look just like local ones. With a few exceptions, so far we have come fairly close. Indeed, as long as both client and server are functioning perfectly, RPC does its job well. The problem comes about when errors occur. It is then that the differences between local and remote calls are not always easy to mask.

Distinguish between five different classes of failures that can occur in RPC systems, as follows:

1. The client is unable to locate the server.

2. The request message from the client to the server is lost.

3. The server crashes after receiving a request.

4. The reply message from the server to the client is lost.

5. The client crashes after sending a request.

Each of these categories poses different problems and requires different solutions.

7. Explain about different types of ordering multicast with examples?

A. Ordered Multicast

• FIFO ordering: If a correct process issues multicast(g,m) and then multicast(g,m'), then every correct process that delivers m' will have already delivered m.

-Preserving the process order. The message delivery order at each process should preserve the message sending order from every process.

For example,
– P1: m0, m1, m2
– P2: m3, m4, m5
– P3: m6, m7, m8
FIFO? (m0, m3, m6, m1, m4, m7, m2, m5, m8) – Yes!
FIFO? (m0, m4, m6, m1, m3, m7, m2, m5, m8) – No!

• Causal ordering: If multicast(g,m) à multicast(g,m') then any correct process that delivers m' will have already delivered m. – Typically, à defined in terms of multicast communication only.

-Preserving the happened-before relations.The message delivery order at each process should preserve the happened-before relations across all processes.

For example,
– P1: m0, m1, m2
– P2: m3, m4, m5
– P3: m6, m7, m8

– Cross-process happened-before: m0 à m4, m5 à m8
Causal? (m0, m3, m6, m1, m4, m7, m2, m5, m8) – Yes!
Causal? (m0, m4, m1, m7, m3, m6, m2, m5, m8) – No!

• Total ordering: If a correct process delivers message m before m’ (independent of the senders), then any other correct process that delivers m’ will have already delivered m.

Every process delivers all messages in the same
order. For example,
– P1: m0, m1, m2
– P2: m3, m4, m5
– P3: m6, m7, m8
Total? – P1: m7, m1, m2, m4, m5, m3, m6, m0, m8
– P2: m7, m1, m2, m4, m5, m3, m6, m0, m8
– P3: m7, m1, m2, m4, m5, m3, m6, m0, m8

– P2: m7, m2, m1, m4, m5, m3, m6, m0, m8

– P3: m7, m1, m2, m4, m5, m3, m6, m8, m0

8. Explain about 2 phase commit protocol. How does 3 phase commit protocol differ from 2 phase commit protocol?

A. A two-phase commit is a standardized protocol that ensures that a database commit is implemented in the situation where a commit operation must be broken into two separate parts.

In database management, saving data changes is known as a commit and undoing changes is known as a rollback. Both can be achieved easily using transaction logging when a single server is involved, but when the data is spread across geographically-diverse servers in distributed computing (i.e., each server being an independent entity with separate log records), the process can become more tricky.The difference between 2PC and 3PC protocols is:

1. For 2PC, the coordinator may abort the transaction globally or resend the global decision. For a participant, it can leave the process blocked until communication with the coordinator is re-established such as sending an abort message to the coordinator or invoke the cooperative termination protocol.

2. For 3PC, the coordinator can abort the transaction globally, send global-commit message to the participants or simply send the global decision to all sites that have not acknowledged. The participant can unilaterally abort a transaction, follow an election protocol, or elect a new coordinator.

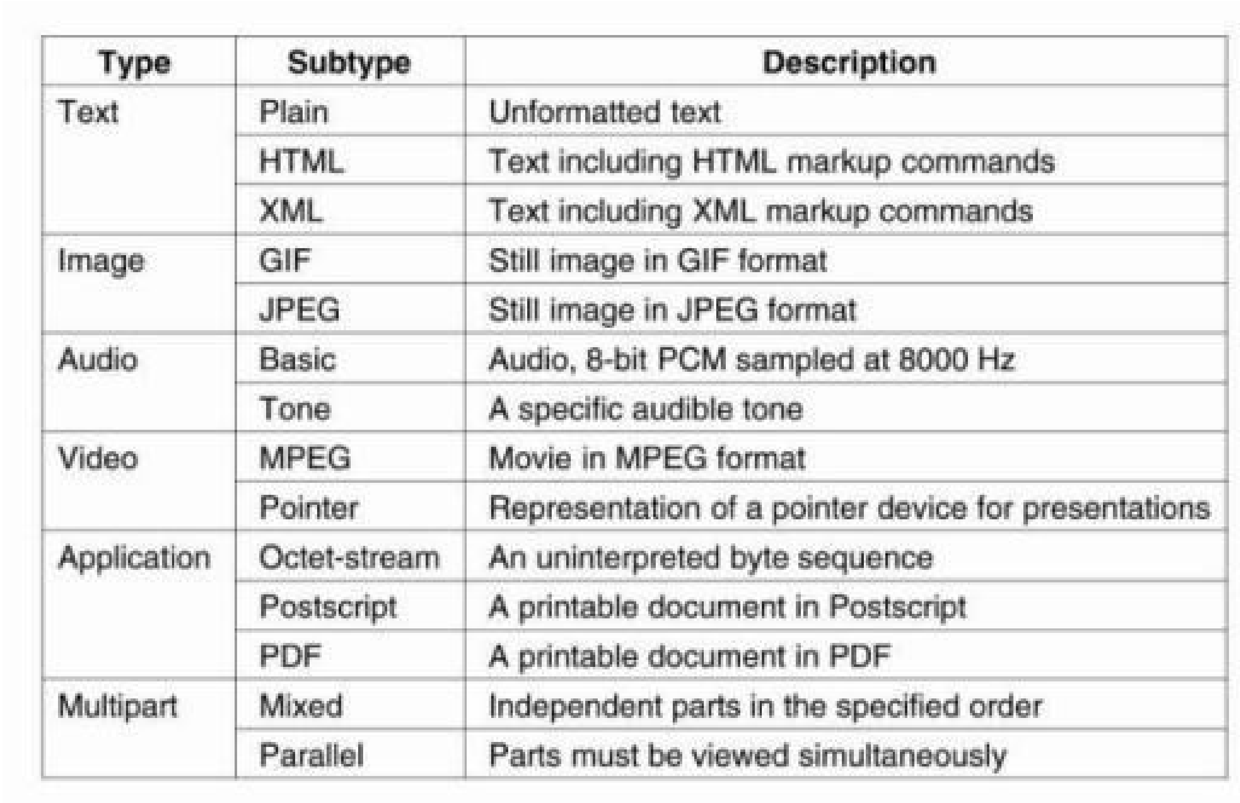
The 2PC protocol is a blocking Two-Phase commit protocol. The 3PC protocol is a non-blocking Three-Phase commit protocol. However, the 3PC protocol does not recover in the event the network is segmented.

9. Is a file server implementing NFS version 3 required to be stateless?

A. The NFS version 3 protocol assumes a stateless server implementation. Statelessness means that the server does not need to maintain state about any of its clients in order to function correctly. Stateless servers have a distinct advantage over stateful servers in the event of a crash. With stateless servers, a client need only retry a request until the server responds; the client does not even need to know that the server has crashed. NFS version 3 protocol server can't maintain noncritical state. In many cases, servers will maintain state (cache) about previous operations to increase performance.

10. Write down six top-level MIME types and some common sub-types?

A.



Type	Subtype	Description
Text	Plain	Unformatted text
	HTML	Text including HTML markup commands
	XML	Text including XML markup commands
Image	GIF	Still image in GIF format
	JPEG	Still image in JPEG format
Audio	Basic	Audio, 8-bit PCM sampled at 8000 Hz
	Tone	A specific audible tone
Video	MPEG	Movie in MPEG format
	Pointer	Representation of a pointer device for presentations
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in Postscript
	PDF	A printable document in PDF
Multipart	Mixed	Independent parts in the specified order
	Parallel	Parts must be viewed simultaneously

Nasigari Jitesh
2451-18-733-082

11) Write down six top-level MIME types and some common sub-types.?

Ans:

.net

.edu

.org

.com

.mil

.gov

a) .net

Network shorthand was developed specifically for institutions participating in-network systems such as a web service provider or an infrastructure firm. As with .com, the restrictions were never upheld to limit.net to networking purposes, and it was one of the most popular top-class fields, and many saw it as close to the top-class domain.com.

b).edu

.edu was produced for educational organizations by shorthand for education. While the TLD.edu was designed for universities worldwide, only instructional centres in America were linked to it. Schools in other nations use .edu together with their national domain, which is discussed in the next chapter below.

c) .org

.org was developed for Nonprofits, shorthand for the organization. As we saw with the other domains at the top, these intentions were often not maintained or implemented over time. Today, the company’s non-profits, profit enterprises, schools, and communities are used as high-level domain.

d) .com

The first high-level domain in popular use was for business shorthand, .com. Whilst .com was originally developed for use by businesses, this was not restricted strictly. .com was the most common type of high-level domain for companies, websites, and emails by the mid-1990s.

e) .mil

Military shorthand,.mil was specifically developed for the US military. This limitation is still maintained as opposed to other top-level domain kinds. In combination with the .mil TLD, it is common for .mil to use second and third-level domains.

f) .gov

Shorthand was restricted for the government, like .mil to federal government agencies only and only staff use. Today,.gov has been used in the town, cities, area, cities, counties and Native American tribes by government agencies.

12. Explain the difference between a plug-in, an applet, a servlet and a CGI program.?

Ans:

Sr. No.	Key	Applets	Servlets
1	Execution	Applets are executed on client-side i.e applet runs within a Web browser on the client machine.	Servlets on other hand executed on the server-side i.e servlet runs on the web Page on server.
2	Parent packages	Parent package of Applet includes java.applet.* and java.awt.*	Parent package of Servlet includes javax.servlet.* and java.servlet.http.*

Sr. No.	Key	Applets	Servlets
3	Methods	Important methods of applet includes init(), stop(), paint(), start(), destroy().	Lifecycle methods of servlet are init( ), service( ), and destroy( ).
4	User interface	For the execution of the applet, a user interface is required such as AWT or swing.	No such interface is required for the execution of servlet.
5	Required Bandwidth	The applet requires user interface on the client machine for execution so it requires more bandwidth.	On the other hand, Servlets are executed on the servers and hence require less bandwidth.
6	Secure	Applets are more prone to risk as execution is on the client machine.	Servlets are more secure as execution is under server security.

13. Explain the difference between a plug-in, an applet, a servlet and a CGI program.?

Ans:

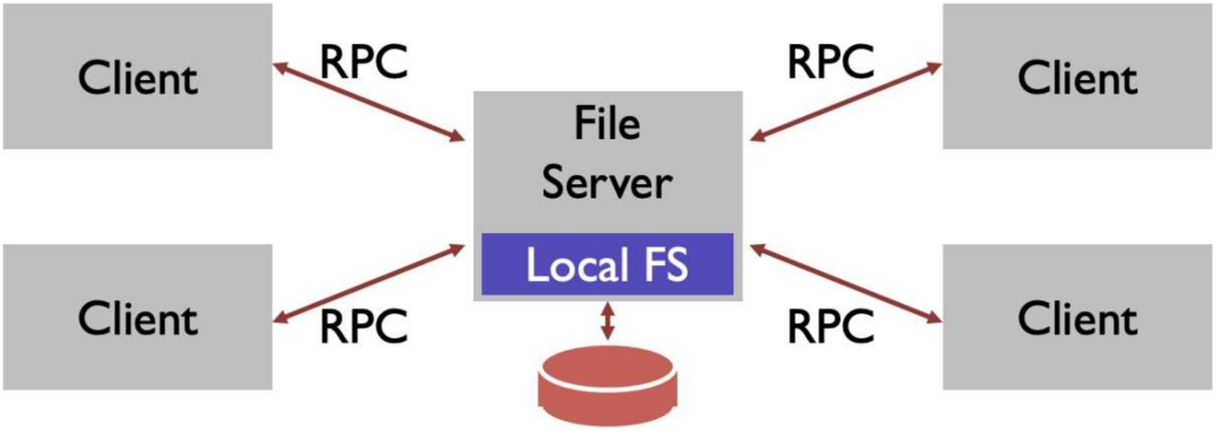
Sr. No.	Key	Applets	Servlets
1	Execution	Applets are executed on client-side i.e applet runs within a Web browser on the client machine.	Servlets on other hand executed on the server-side i.e servlet runs on the web Page on server.

Sr. No.	Key	Applets	Servlets
2	Parent packages	Parent package of Applet includes java.applet.* and java.awt.*	Parent package of Servlet includes javax.servlet.* and java.servlet.http.*
3	Methods	Important methods of applet includes init(), stop(), paint(), start(), destroy().	Lifecycle methods of servlet are init( ), service( ), and destroy( ).
4	User interface	For the execution of the applet, a user interface is required such as AWT or swing.	No such interface is required for the execution of servlet.
5	Required Bandwidth	The applet requires user interface on the client machine for execution so it requires more bandwidth.	On the other hand, Servlets are executed on the servers and hence require less bandwidth.
6	Secure	Applets are more prone to risk as execution is on the client machine.	Servlets are more secure as execution is under server security.

14. Draw the architecture of SUN NFS and explain the flow control of SUN NFS.?

Ans: Sun’s Network File System:
The earliest successful distributed system could be attributed to Sun Microsystems, which developed the Network File System (NFS). NFSv2 was the standard protocol followed for many years, designed with the goal of simple and fast server crash recovery. This goal is of utmost importance in multi-client and single-server based network architectures because a single instant of server crash means that all clients are unserviced. The entire system goes down.

Stateful protocols make things complicated when it comes to crashes. Consider a client A trying to access some data from the server. However, just after the first read, the server crashed. Now, when the server is up and running, client A issues the second read request. However, the server does not know which file the client is referring to, since all that information was temporary and lost during the crash.



15. Draw the layered architecture of JINI system?

Ans:

	Infrastructure	**Programming Model**	**Services**
Base Java	Java VM	Java APIs	JNDI
	Java RMI	JavaBeans	Enterprise Beans
	Java Security	...	JTS

Java + **Jini**	Discovery/Join	Leasing	Printing
	Distributed Security	Transactions	Transaction Manager
	Lookup	Events	JavaSpaces™ Service

16. Explain about Hadoop?

Hadoop is a popular open-source Java implementation of MapReduce and is used to build cloud environments in a highly fault tolerant manner. Hadoop will process web-scale data of the order of terabytes or petabytes by connecting many commodity computers together to work in parallel. Hadoop includes a complete

distributed batch processing infrastructure capable of scaling to hundreds or thousands of computing nodes, with advanced scheduling and monitoring

capability. Hadoop is designed to have a 'very fat scalability curve' meaning that once a program is created and tested on a small number of nodes, the same program can then be run on a huge cluster of machines with minimal or no further programming required. Reliable performance growth should then be in proportion to the number of machines available.

The Hadoop File System (HDFS) splits large data files into chunks which are managed by different nodes on the cluster so that each node is operating on a subset of the data. This means that most data is read from the local disk directly into the CPU, thus reducing the need to transfer data across the network and therefore improving performance. Each chunk is also replicated across the cluster so that a single machine failure will not result in data becoming inaccessible.

Fault tolerance is achieved mainly through active monitoring and restarting tasks when necessary. Individual nodes communicate with a master node known as a 'JobTracker'. If a node fails to communicate with the job tracker for a period of time (typically 1 min), the task may be restarted. A system of speculative execution is often employed such that once most tasks have been completed, the remaining tasks are copied across a number of nodes. Once a task has completed, the job tracker is informed, and any other nodes working on the same tasks can be terminated.

17. Explain about coordination models along with temporal and referential dimensions?

Ans. The coordination part of a distributed system handles the communication and cooperation between processes. It forms the glue that binds the activities performed by processes into a whole.
When processes are temporally and referentially coupled, coordination takes place in a direct way, referred to as direct coordination. The referential coupling generally appears in the form of explicit referencing in

communication. For example, a process can communicate only if it knows the name or identifier of the other processes it wants to exchange information with. Temporal coupling means that processes that are communicating will both have to be up and running. This coupling is analogous to the transient message-oriented communication we discussed in Chap. 4.

A different type of coordination occurs when processes are temporally decoupled, but referentially coupled, which we refer to as mailbox coordination. In this case, there is no need for two communicating processes to execute at the same time in order to let communication take place. Instead, communication takes place by putting messages in a (possibly shared) mailbox. This situation is analogous to persistent message-oriented communication as described in Chap. 4. It is necessary to explicitly address the mailbox that will hold the messages that are to be exchanged. Consequently, there is a referential coupling.

The combination of referentially decoupled and temporally coupled systems form the group of models for meeting-oriented coordination. In referentially decoupled systems, processes do not know each other explicitly. In other words, when a process wants to coordinate its activities with other processes, it cannot directly refer to another process. Instead, there is a concept of a meeting in which processes temporarily group together to coordinate their activities. The model prescribes that the meeting processes are executing at the same time.

18. Write short notes on Windows AZURE.?

Ans: Azure is Microsoft’s cloud platform, just like Google has it’s Google Cloud and Amazon has it’s Amazon Web Service or AWS.000. Generally, it is a platform through which we can use Microsoft’s resource. For example, to set up a huge server, we will require huge investment, effort, physical space and so on. In such situations, Microsoft Azure comes to our rescue. It will provide us with virtual machines, fast processing of data, analytical and monitoring tools and so on to make our work simpler. The pricing of Azure is also simpler and cost-effective. Popularly termed as “Pay As You Go”, which means how much you use, pay only for that.

Azure can help in our business in the following ways-

Capitaless: We don’t have to worry about the capital as Azure cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that’s kind to your cash flow. Also, to set up an Azure account is very easy. You simply register in Azure Portal and select your required subscription and get going.

Less Operational Cost: Azure has low operational cost because it runs on its own servers whose only job is to make the cloud functional and bug-free, it's usually a whole lot more reliable than your own, on-location server.

Cost Effective: If we set up a server on our own, we need to hire a tech support team to monitor them and make sure things are working fine. Also, there might be a situation where the tech support team is taking too much time to solve the issue incurred in the server. So, in this regard is way too pocket-friendly.

Easy Back Up and Recovery options: Azure keep backups of all your valuable data. In disaster situations, you can recover all your data in a single click without your business getting affected. Cloud-based backup and recovery solutions save time, avoid large up-front investment and roll up third-party expertise as part of the deal.

Easy to implement: It is very easy to implement your business models in Azure. With a couple of on-click activities, you are good to go. Even there are several tutorials to make you learn and deploy faster.

19. Explain in detail about MAp-reduce programming approach with suitable example?

A. MapReduce was originally introduced by Google as a distributed programming model using large server clusters to process massive (multi-terabyte) data sets. The model can be applied to many large-scale computing problems and offers a number of attractive features such as automatic parallelisation, load balancing, network and disk transfer optimisation and robust handling of machine failure. MapReduce MapReduce runs on top of a specialised file system such as the Google File System (GFS) or the Hadoop File System (HDFS). Data is loaded, partitioned into chunks works by breaking a large problem into smaller parts, solving each part in parallel and then combining results to produce the final answer.

(commonly 64 MB) such that each chunk can replicate. A key feature of MapReduce is that data processing is collocated with data storage, and as a distributed programming paradigm, a number of advantages are evident when compared to the traditional approach of moving the data to the computation including:

1. Scalability

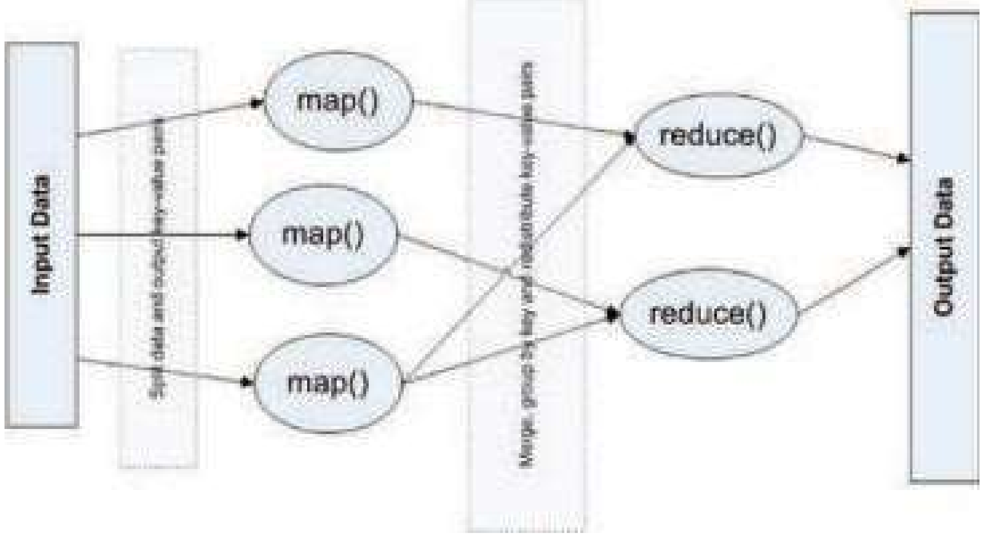
2. Reliability

3. Fault tolerance

4. Simplicity

5. Efficiency

MapReduce was inspired by the map and reduce functions which are commonly found in functional program languages like LISP.


Input Data
Split data and output key-value pairs
map()
map()
map()
Merge, group by key and redistribute key-value pairs
reduce()
reduce()
Output Data

Ex:

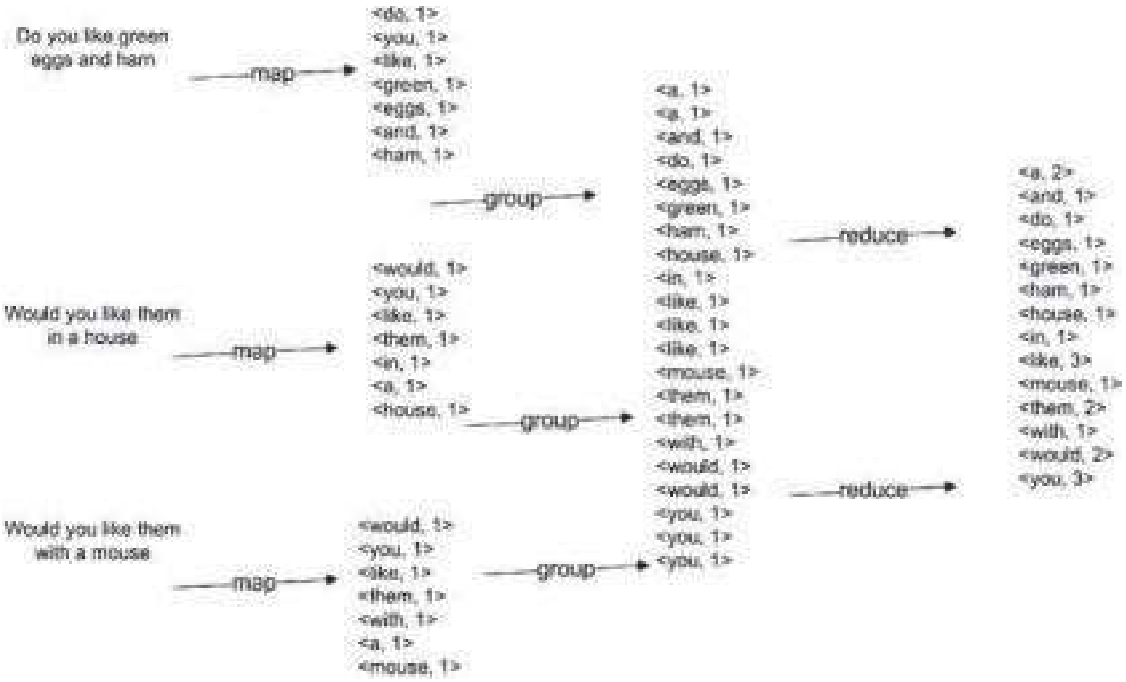


Figure shows a simple example with the three input fles shown on the left and the output word counts shown on the right. The same kind of process can be applied to many other problems and is particularly useful where we need to process a huge amount of raw data.

20) Explain about apache web?

Apache is a complex piece of software, and with the numerous enhancements to the types of documents that are now ofered in the Web, it is important that the server is highly confgurable and extensible, and at the same time largely independent of specifc platforms.

Making the server platform independent is realized by essentially providing its own basic runtime environment, which is then subsequently implemented for diferent operating systems.

This runtime environment, known as the Apache Portable Runtime (APR), is a library that provides a platform-independent interface for fle handling, networking, locking, threads, and so on. When extending Apache (as we will discuss shortly), portability is largely guaranteed provided that only calls to the APR are made and that calls to platform-specifc libraries are

avoided.

As we said, Apache is tailored not only to provide flexibility (in the sense that it can be configured to considerable detail), but also that it is relatively easy to extend its functionality.

For example, later in this chapter we will discuss adaptive replication in Globule, a home-brew content delivery network developed in the authors' group at the Vrije Universiteit Amsterdam.

Globule is implemented as an extension to Apache, based on the APR, but also largely independent of other extensions developed for Apache.

Nasigari Jitesh
2451-18-733-082

21) Explain about synchronization in Distributed coordination - based systems

Synchronization

Synchronization in coordination-based systems is generally restricted to systems supporting generative communication. Matters are relatively straightforward when only a single server is used. In that case, processes can be simply blocked until tuples become available, but it is also simpler to remove them. Matters become complicated when the shared dataspace is replicated and distributed across multiple servers, as we describe next.