PART-B
ESSAY QUESTIONS WITH SOLUTIONS

1.1 FEATURES OF CLOUD AND GRID PLATFORMS

Q9. Explain the capabilities of cloud and platform features.

Ans:

Cloud Capabilities and Platform Features: The commercial clouds require huge number of capabilities. They provide computing of less cost with flexibility. This will in addition provides extra capabilities together called as "Platform as a Service". The present platform features for the azure are table, queues, web, worker roles, database SQL, blobs. The platform features for Amazon are "Infrastructure as a Service" (IaaS) all together other than queues, notification, monitoring, content delivery network, relational database and map reduce. The capabilities of cloud platform are listed in the below table.

	Capability	Description
1.	Physical or virtual computing platform	The cloud environment incorporates both physical as well as virtual platform. The virtual platforms posses the unique capabilities in order to isolate environments for various applications and users.
2.	Huge data storage service, distributed file system	The cloud data storage services after a wide range of disk capacity for heavy data sets. In addition to this service interface and store data the distributed file system provides heavy data storage services.
3.	Huge database storage service	The clouds require this service similar to DBMS so that the developers can store the data in semantic way.
4.	Huge data processing method and programming model	The infrastructure of cloud after several nodes for simple applications. So the programmers must manage the issues such as network failure or scaling of running code etc in order to use all the services provided by platform.
5.	Support workflow and data query language	The programming model wides the cloud infrastructure. The workflow language and data query language is provided for application logic.
6.	Programming interface and services deployment	Cloud applications need web interfaces or special API's such as J2EE, PHP, ASP or Rails. They can make use of AJax technologies for improving user experience while using web browsers for function access.
7.	Runtime support	The runtime support is open for the users as well as applications. it incorporates the distributed monitoring services, distributed task scheduler, distributed locking etc.
8.	Support services	Various important services are data and computing services.

The infrastructure cloud features as illustrated in below table.

	Feature	Description
1.	Accounting	It has economies and acts as an active area for commercial clouds.
2.	Appliances	It configures virtual machine image that supports multi faceted tasks like Message passing interface clusters.
3.	Authentication and authorization	It requires all the systems to have single sign in.
4.	Data transport	It supports the data transfer among job components within and between the clouds and grids.
5.	Registry	It provides registry as information resource for system.
6.	Operating systems	Supports OS such as Android, Linear, Windows and Apple
7.	Program library	It stores images as well as program material.

8.	Scheduling and gang scheduling	Provides scheduling similar to that of Azure worker role in addition to condor, platform, oracle grid engine etc. The gang scheduling will assign multiple tasks scalably.
9.	Software as a Service (SaaS)	This service is shared among clouds and grids. It is successful and is used in clouds as distributed systems.
10.	Virtualization	It is a basic feature and supports elastic feature. It also incorporates virtual networking.

Traditional features in cluster, grid and parallel computing environments are illustrated in the below table.

	Feature	Description
1.	Cluster management	Clusters are developed using the tools provided by rocks and packages.
2.	Data management	A metadata support called RDR triple stores is provided. In addition to these SQL and NOSQL are also provided.
3.	Portals	It is also termed as gateways that has transformation in technology from portlets to HUB zero.
4.	Virtual organizations	Organizations range from specialized grid solutions to popular web 2.0.
5.	Grid programming environment	The programming environment differs from link together services as in open grid services architecture to grid RPC and SAGA.
6.	Open MP/Threading	It incorporates parallel computers like click and roughly shared memory technologies in addition to transactional memory and fine grained data flow.

The platform features supported by clouds and grids are as follows,

	Feature	Description
1.	Blob	It provides the basic storage concept that is typified by Azure, Blob, and Amazon S3
2.	DPFS	It provides support for file systems like Google, HDFS, cosmos along with compute data affinity that is optimized for data processing.
3.	Map reduce	It supports map reduce programming model with Hadoop on Linux, Dryad on windows HPCS and thirster on windows and Linex.
4.	Fault tolerance	It is a major feature of clouds.
5.	Notification	It is the basic function of publish subscribe systems.
6.	Monitoring	It provides many grid solutions like Inca which is be based on publish subscribe.
7.	Programming model	The cloud programming models are developed with other platform features. It relates to web and grid models.
8.	Queues	The queue system depends on publish subscribe.
9.	SQL	It is a relational database.
10.	Table	It supports the table data structures that are modeled on apache Hadoop or Amazon Simple DB/Azure table.
11.	Web role	It is used in Azure for determining the link tosser.
12.	Worker role	It is used in Amazon and Grids.
13.	Scalable synchronization	It is a apache 200 keeper or Google clubtry. It supports distributed locks and is used by big table.

Q10. Discuss various traditional features common to grids and clouds.

Ans: Various features that are common to grids and clouds are as follows,

1. **Workflow:** In US and Europe the workflow has created various projects such as pegasur, Taverna and Kepler. The commercial systems include Pipeline, Pilot, AVS and LIMS environment. Trident is the latest entry which will run the workflow proxy services on external environments if working on Azure or Windows.

2. **Data Transport:** The data transfer is the major issue in commercial clouds. The links of high bandwidth can be allowed among clouds and tera grid if at all commercial clouds are made the major components of cyber infrastructure. The cloud data can be structured into tables and blocks in order to make the high performance parallel algorithms in addition to HTTP mechanisms for data transfer among academic systems/ Teragrid and commercial clouds.

3. **Security, Privacy and Availability:** These techniques related to security, privacy and availability used for developing good and dependable cloud programming environment are illustrated as follows,

- Using virtual clustering for achieving dynamic resource support at less overhead cost.
- Using special API's for user authentication and e-mail sending through commercial accounts.
- Accessing cloud resources using security protocols like ATTPS and SSL.
- Using stable and persistent data storage through quick queries for data access.
- Including the features for improving availability and disaster recovery with file migration of VM's.
- Using fine gained access control for protecting the data integrity and deterring intruders and hackers.
- Protecting shared data form Malicious alternation, deletion of copy right Molations.
- Using popular systems for protecting data centres from stopping the privates and authorizing the trusted clients.

Q11. Write about data features and databases.

Ans: The features of data and databases are illustrated as follows,

1. **Program Library:** An attempt is made for developing a VM image library for managing the images to be used in academic and commercial clouds.

2. **Bolbs and Drives:** The containers of azure are responsible for arranging the storage in clouds such as blobs for azure and S3 for Amazon. The users are allowed to attach directly for computing instances like Azure drives and Elastic Book Store for amazon. The cloud storage is found to be fault tolerant when tera grid require backup storage.

3. **DPFS:** It supports the file systems like Google File System, HDFS and cosmos with the features of optimized compute data affinity for processing the data. The DPFS can linked with blob and drives based architecture but it is better to use if application centric storage model with optimized compute data affinity. With this the blob and drives must be used as repository centric view. The DPFS file systems are developed for executing the data intensive applications efficiently.

4. **SQL and Relational Databases:** The relational databases are offered by the amazon and azure clouds. As done earlier a new private computing model is built on future grid for the observational medical outcomes partnership for patient related data that makes use of oracle and SAS. This is where the future grid includes hadoop to scale various analysis methods. The databases are predicated to be used for determing the methods that deploy capabilities. The database software can be added on to the disk. It can be executed through the database instance. The database on amazon and azure can be installed on different VM with this "SQL as a service" gets implemented.

5. **Table and NOSQL Non-related Databases:** A large number of developments took place related to a simple database structure called "NOSQL". This emphasizes distribution and scalability. The clouds like Google, big table and simple DB in amazon and azure make use of. The tables are used in service that is illustrated by VO table standard in astronomy and excel popularity. Non-relational database are used in many terms of triple stores depending upon the mapreduce and tables or hadoop file system with good success.

The tables of the cloud can be classified into azure table and amazon simple DB which support lightweight storage for document stores. They are found to be schema free and will soon gain importance in scientific computing.

6. **Queuing Services:** The amazon as well as azure provide the robust and scalable queueing services for the components to interact with each other in an application. These messages are short and contain a REST (Representational State Transfer) interface which has 'deliver at least once' semantics. Time outs are used for controlling them in order to post the amount of processing time assigned for the client.

Q12. Write a short note on programming and runtime support.

Ans: The programming and runtime support after parallel programming and runtime support of major functions in grids and clouds.

1. **Worker and Web Roles:** Azure provides roles for facilitating nontrivial functionality and also for preserving better affinity support in non-virtualized environment. They are the schedulable processes that can be launched automatically. Queues are considered to be complex since they after natural method for assigning tasks in fault-tolerant and distributed fashion. The web roles offer an significant method for the portal.

2. **MapReduce:** The data parallel languages are found to have great interest in loosely coupled computations that execute among the data samples. The grid applications are provided with efficient execution by language and runtime. The map reduce is found to be more advantageous than traditional implementations of the task problems.

This is because it supports dynamic execution, strong fault tolerance and easy to use high level interface. Hadoop and dryad are the mapreduce implementations that can be executed with or without VM's. Hadoop is provided by amazon and azure is provided by Azure.

3. **Cloud Programming Models:** The GAC and manjrasoft aneka environment are the two basic programming models that are applied on clouds. But these models are not specific to this architecture model that provides portability between the cloud, HPC and cluster environments is Iterative MapReduce.

4. **SaaS:** Services are used similarly in both commercial clouds and latest distributed systems. The users can package their programs as required. Hence, SaaS services can be enabled without any additional support because of this reason. SaaS environment is expected to provide various useful tools for developing the cloud applications over the huge data sets. In addition to this various protection features are also offered by SaaS for achieving scalability, security, privacy and availability.

4.2 PARALLEL AND DISTRIBUTED PROGRAMMING PARADIGMS

Q13. Discuss in brief about parallel computing and programming paradigms. Also write about the motivation for programming paradigms.

Ans: The distributed and parallel programs are assumed as the parallel programs running on set of computing engines or distributed computing systems. The distributed computing denote the computational engines interconnected in a network intended to run a job or application. The parallel computing denotes the usage of one or more computational engine intended to run a job or application. The parallel programs are allowed to be run on distributed computing systems. But it has certain issues described below,

1. **Partitioning:** Partitional is done in the below two ways,

(i) **Computation Partitioning:** The give program or job is divided into various tasks depending upon the portion identification that is capable for concurrent transaction. Various parts of a program can process different data or share same data.

(ii) **Data Partitioning:** The input or intermediate data is divided into various partitions that can be processed on various workers. A copy of the program or various parts of it is responsible for processing the pieces of data.

2. **Mapping:** The process of assigning parts of program of data pieces to the respective resources is called mapping. It is handled by the system resources allocators.

3. **Synchronization:** Synchronization is required since various workers perform various tasks. Coordination is also important among the workers. With this, race conditions can be prevented and data dependency can also be managed.

4. **Communication:** Communication is considered as the major concept when the intermediate data is sent to workers. This is because data dependency is a major reason for communication among the workers.

5. **Scheduling:** A scheduler is responsible for picking a set of jobs or programs and for running them on distributed computing system. It is required when the resources are not sufficient to run various jobs or programs simultaneously. Scheduling is done based on the scheduling policy.

Motivation for Programming Paradigms: Handling of complete data flow of parallel and distributed programming is observed to be time consuming. It also needs special programming knowledge. These issues affect programmer productivity and programs time to market. For this purpose the parallel and distributed programming paradigms or models are use in order to hide the data flow part from the users.

These models after abstraction layer to the users in order to hide the implementation details of data flow that requires the users to code. An important metric to be considered is simple coding for parallel programming with respect to parallel and distributed programming paradigms. Motivation behind parallel and distributed programming model are as follows,

1. Improve the program productivity
2. Decrease programs time to market
3. Leverage underlying resources efficiently
4. Increasing system throughput
5. Support for higher levels of abstraction.

Q14. Explain in detail about MapReduce function and framework.

Ans: Model Paper-I, Q5(a)

MapReduce: A software framework that allows to perform parallel and distributed computing on huge data sets is called MapReduce. It hides the flow of data of a parallel program on distributed computing system. For this purpose two interfaces Map and Reduce are provided to the users as functions.

The data flow in a program is manipulated through these functions. The below figure illustrates the flow of data from Map to Reduce.

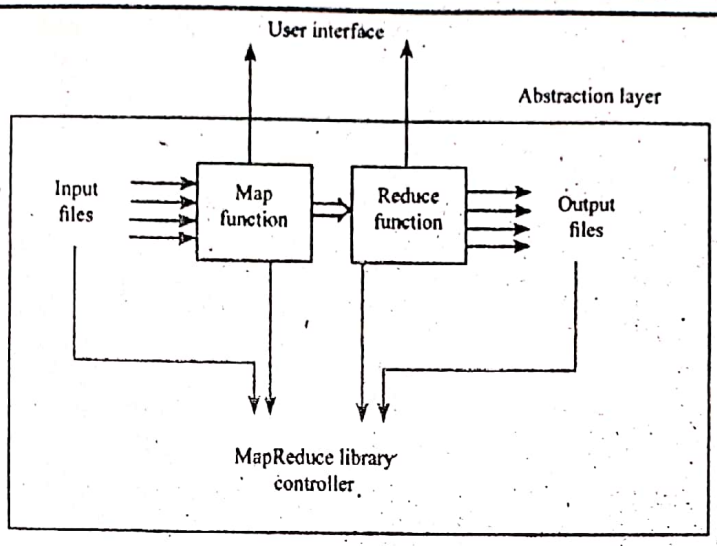


Figure: MapReduce Software Framework

In the above figure the abstraction layer abstracts the data flow steps such as partitoning, mapping, synchronization, communication and scheduling to the users. The Map and Reduce functions can be overriden by the user in order to achieve their respective goals. These functions can be passed with the required parameters such as spec, results etc. The program structure containing Map and Reduce subroutine is illustrated below,

```
Map function(---)
{
≡
}
Reduce function(--)
{
≡
}
Main function(--)
{
initialize spec object
≡
MapReduce(spec & results)
}
```

The input data to map function is a (key, value) pair where key indicates the line offset in a input file and value is the line content. The output returned from map function is also a (key, value) pair called as intermediate pair. The Reduce function is responsible for receiving the intermediate(key, value) pairs as a set of values as (key, [set of values]) by sorting and grouping the same value keys. It processes and generates a group of (key, value) pairs as output. The formal notation of map function is,

$$(key_1, val_1) \xrightarrow{MapFunction} List\ (key_2, val_2)$$

The result obtained is a intermediate (key, value) pairs. They are gathered by the MapReduce library and sorted based on the keys.

The various occurrences of same key are gathered and reduce function is applied on them to produce another list.

$$(key_2, List(val_2) \xrightarrow{Reduce\ Function} List\ (val_2)$$

MapReduce Actual Data and Control Flow

For answer refer Unit-IV, Q15.

Compute Data Affinity: The MapReduce is capable of transforming to GFS, where GFS is a distributed file system in which the files are segregated into fixed-size blocks. These blocks are distributed and stored on cluster nodes. The map function is applied on every block by just sending a copy of the program to nodes containing data blocks. The size of every block is 64 MB.

Q15. Write about the MapReduce actual data and control flow.

Ans: MapReduce framework is responsible for running the program on distributed computing system efficiently. This process is detailed as follows,

1. **Data Partitioning:** The input data is retrieved from GFS and divided into Mpieces by the MapReduce library. These partitions correspond to number of map tasks.
2. **Computation Partitioning:** The obliging users perform computation partitioning for coding as Map and Reduce functions. The result will be a user program containing Map and Reduce functions. They are distributed and initiated on number of computation engines that are available.
3. **Determining the Master and Workers:** The architecture of MapReduce depends upon master workers model. Here a copy of user programs becomes the master and the remaining become the workers. The master is responsible for assigning the map and reduce tasks to the idle workers. And the worker is responsible to run the map/reduce task through Map/ Reduce function execution.
4. **Retrieving Input Data (Data Distribution):** The respective input data is read by the worker and sent to the map function after dividing it.
5. **Map Function:** The input data is retrieved by the map function in the form of (key, value) pairs in order to process and produce the intermediate (key, value) pairs.
6. **Combiner Function:** This function is applied on (key, value) pairs and invoked in the user program. It merges the local data of the map worker and sends it on networks. With this the communication cost decreases.
7. **Partitioning Function:** The intermediate (key, value) palrs are partitioned using partitioning function. All the similar keys are stored in same region through hash function. The data later sent to master which inturn forward to the workers.

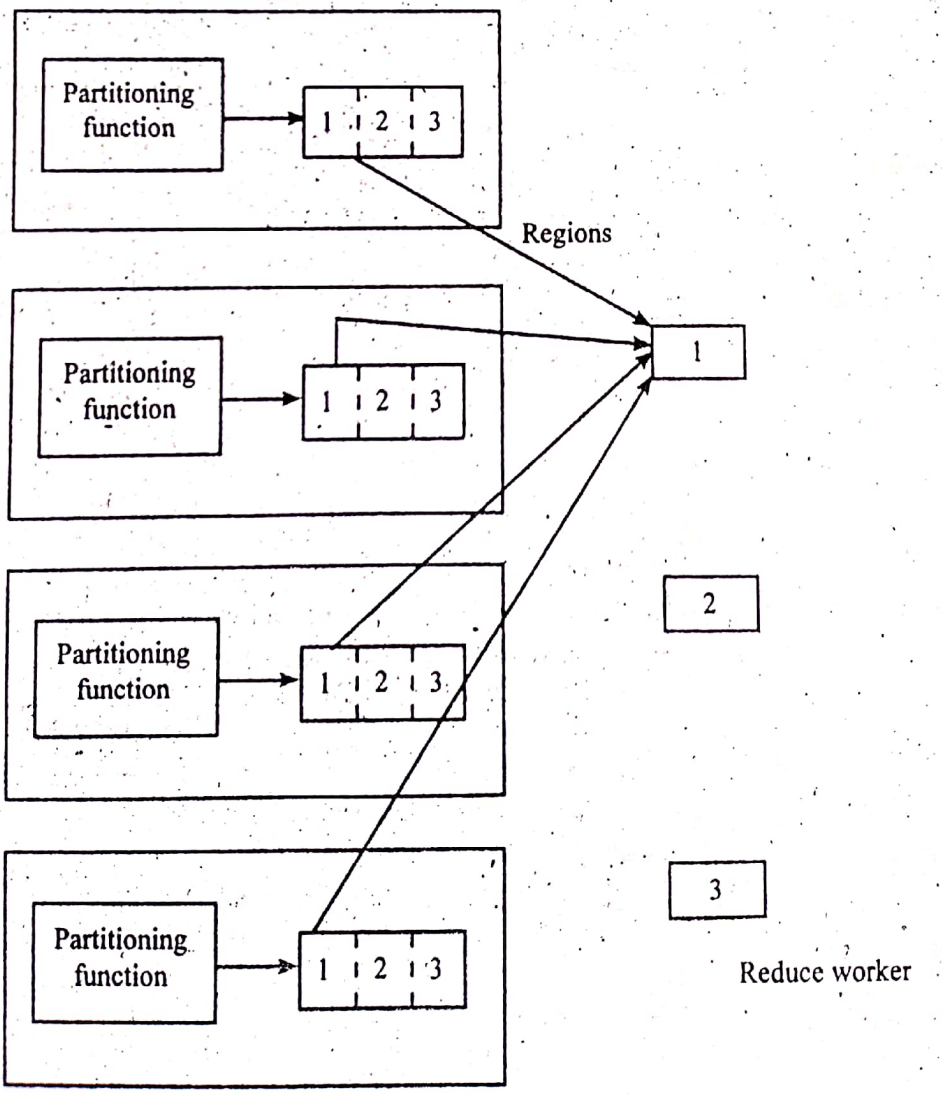


Figure: Linking Map and Reduce Workers Through MapReduce Partitioning Function

8. **Synchronization:** Synchronization policy of MapReduce allow coordination between map workers and reduce workers and provides interaction among them after task completion.

9. **Communication:** A remote procedure call is used by the reduce worker for reading the data from the map workers. A all-to-all communication occurs between the map and reduce workers giving rise to network congestion. For this purpose a data transfer module is developed for scheduling the data transfer.

10. **Sorting and Corresponding:** A reduce worker decides the reading process of input data and groups the intermediate (key, value) pairs by sorting the data according to the keys. All the occurrences of similar keys are grouped and unique is produced.

11. **Reduce Function:** The reduce worker is responsible for iterating the grouped (key, value) pairs for all the unique keys and the set of key and values are sent to reduce function. This function will process the received data and stores the output in predetermined files of user program.

Q16. Discuss about twister and iterative MapReduce.

Ans:

Twister and Iterative MapReduce: The performance of any runtime requires to be checked and the MPI and MapReduce also need to be compared. The communication and load imbalance are the important sources of parallel overhead. The overhead of communication can be high in MapReduce because of the following reasons.

- The MapReduce performs read and write through files and MPI allows data transfer between the nodes in the network.
- MPI will not transfer the complete data rather it only does the required data for updation. The MPI flow is called flow and MapReduce flow is called full data flow.

This phenomenon can be observed in all the classic parallel loosely synchronous applications that show off the iteration off structure in the compute phases and communication phases. The performance issues can deposited with below changes.

- Transfer of data between the steps without expanding the steps internally to disk.
- Usage of long-running threads or processors for communicating flow.

The above changes give rise to increase in performance at the cost of fault tolerance and also supports dynamic changes like available nodes. The below figure depicts the twister programming paradigm along with its architecture at run time. The twister illustrates the difference of static data that can never be reloaded from dynamic flow that is communicated.

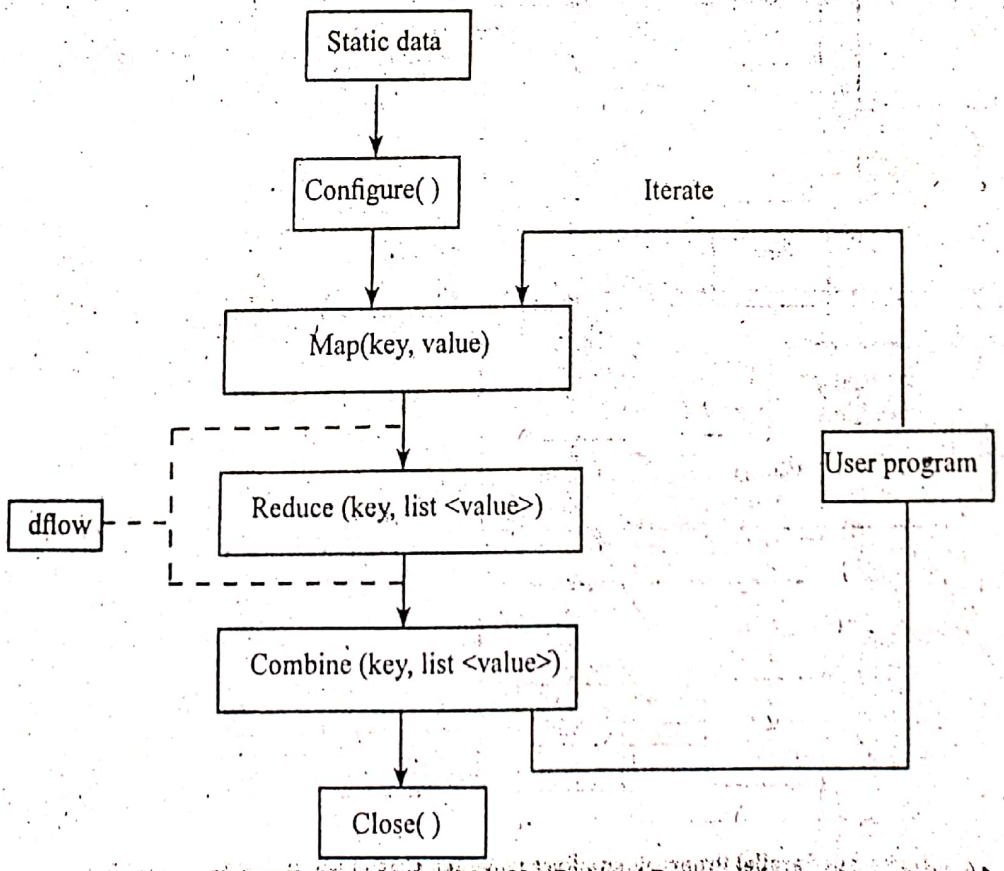


Figure: Twister for Iterative MapReduced Programming

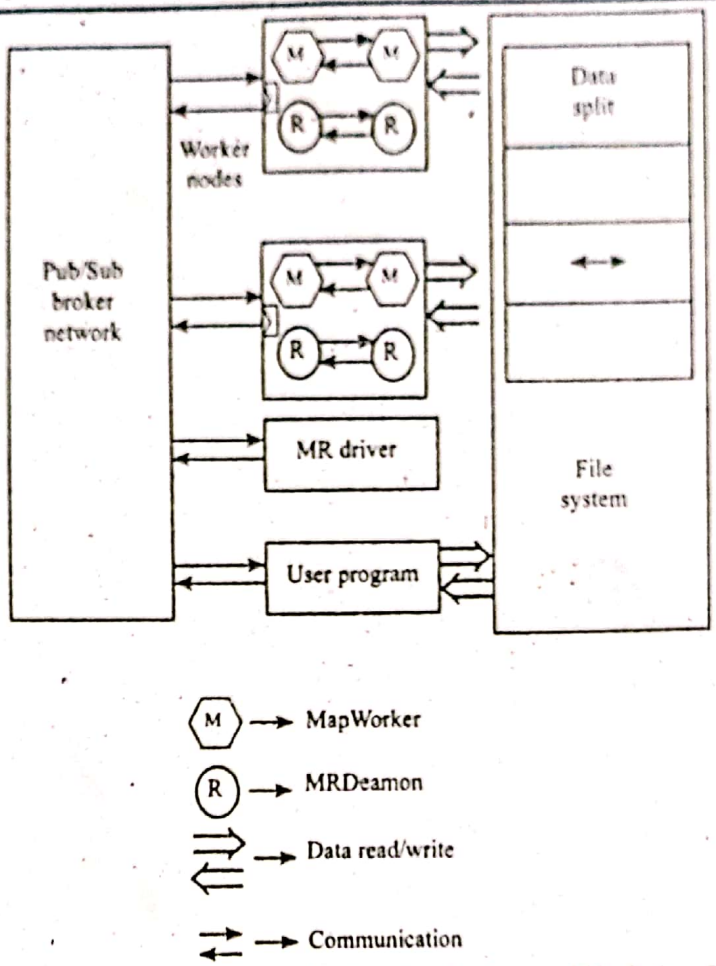


Figure: Iterative MapReduce Programming Paradigm for Repeated MapReduce Executions

The pair of map and reduce is executed iteratively in thread that are long running. The below figure shows the comparison of thread and process structures of parallel programming paradigms like hadoop, dryad, twister and MPI.

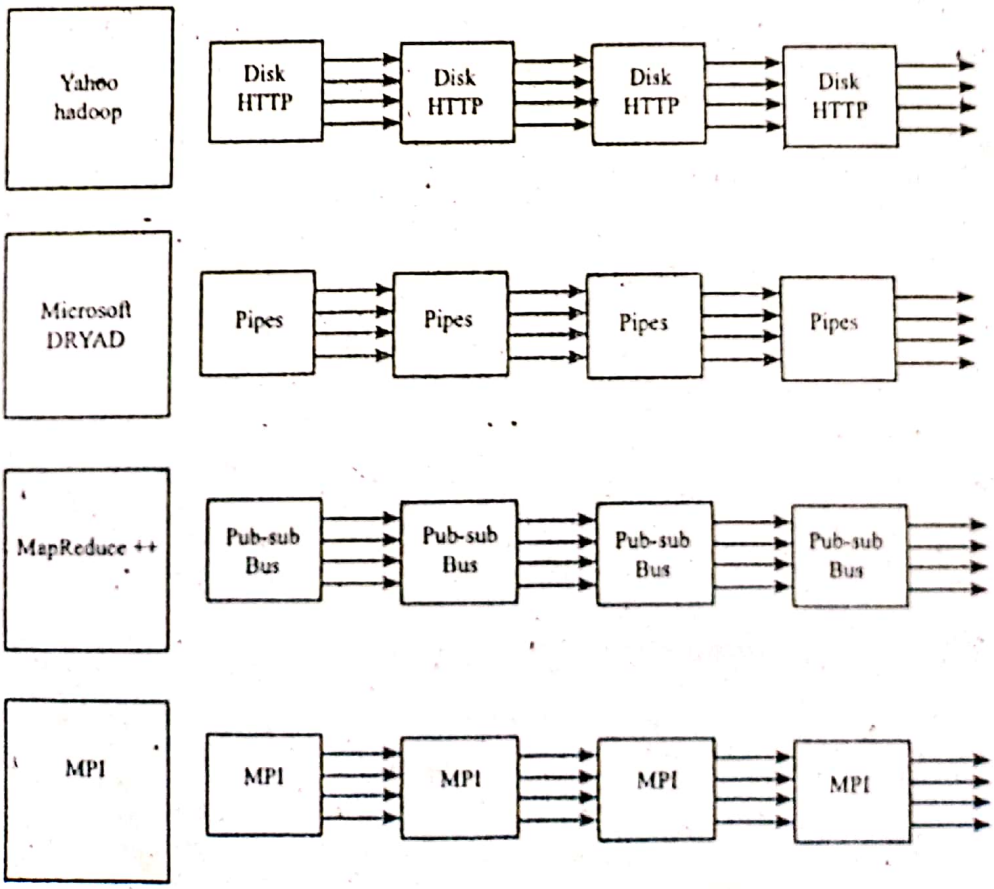


Figure: Four Parallel Programming Paradigms for Thread and Process Structure

Yahoo Hadoop: It is used for short running processes communication through disk and tracking process.

Microsoft Dryad: It is used for short running processes communication through pipes, disk or shared memory between cores.

MapReduce ++: It is used for long running processing with asynchronous distributed rendezvous synchronization.

MPI: It is used for long running process with rendezvous for message exchange synchronization.

Q17. Discuss about hadoop library from apache.

Ans:

Hadoop Library from Apache: MapReduce has an open sources implementation called hadoop. It is coded in Java by apache and makes use of Hadoop Distributed File System (HDFS) as internal layer. The core of hadoop has two layers called MapReduce engine and HDFS. The MapReduce engine is the top layer and acts as computations engine and data storage manager.

MapReduce in Hadoop

For answer refer Unit-IV, Q18.

HDFS: HDFS acts as a distributed file system and stores the data on distributed computed system after organizing. Architecture of it contains a master and slave with single NameNode and number of DataNodes respectively. The files are divided into fixed sized blocks by this architecture are stored on workers. Mapping will be done based on NameNode. The NameNode is responsible for managing the file systems metadata and namespace. Moreover it maintains the metadata in the area and meta data is the data of file. System accessible to the file management.

Features of HDFS are as follows,

1. **Fault Tolerance:** Fault tolerance is an important characteristic of HDFS. As hadoop is to be deployed on low-cost hardware, it frequently comes across hardware failure. For this reason hadoop considers the below issues to comes the reliability requirements.

(i) **Block Replication:** To ensure data reliability, replications of file blocks are maintained and distributed across the cluster.

(ii) **Replica Placement:** Replica placement is one of issue in building the fault tolerance. It is reliable to store the replicas on nodes of other racks in the cluster. But this technique is not much considered because it is of high cost. So, reliability is compromised to make HDFS cost effective.

(iii) **Heartbeat and Block Report Messages:** The periodic messages which the DataNode sends to the NameNode are Heartbeats and Blockreports. This implies the proper functioning of DataNode. The block report consists a list of blocks in DataNode.

2. **High Throughput Access to Large Data Sets:** The throughput of HDFS is important because it is designed and purposed for batch processing. In addition to this the applications that runs on HDFS contain heavy data sets and separate files. These files are divided into large blocks so that HDFS can decrease the storage of metadata required by a file. With this the block list decreases with the increase in block size and also fast streaming reads are provided by the HDFS.

Operations of HDFS: Operations of HDFS are depicted as follows,

1. **File Read:** To perform red operation the user will send the "open" request to NameNode for file block location. The response will be address of DataNode in which replica data is stored. The addresses depend upon the block replicas, After this read is performed to connect to the nearby DataNode. The connection will be terminated after streaming the connection. The complete process will iterative until the file is streamed completely to the user.

2. **File Write:** The user initially will send a create request to NameNode for new file creation. Then the data is written to it using write function. The data queue which is an internal queue first holds the first data block later it is written to DataNode while the data streamer monitors it. At parallely even replicas of the data blocks are also created accordingly.

Q18. Explain the architecture of MapReduce in hadoop and running a job in hadoop.

Ans: **Model Paper-I, Q5(b)**

Architecture of MapReduce in Hadoop: MapReduce engine is the upper layer of the hadoop. It is responsible for managing the data flow and control flow of the MapReduce jobs in distributed computing systems. MapReduce engine contains a master/slave architecture with single JobTracker (That acts as a master) and several TaskTrackers (Which act as slaves). The MapReduce job is managed by the JobTracker over the cluster. It also monitors and assigns the jobs and tasks to the TaskTrackers. The TaskTracker is responsible for managing the map/reduce tasks execution on a computation node within the cluster.

Every TaskTracker is assigned with various execution. slots to execute map or reduce task. A map task that is running on a slot will process a data block. A one-to-one correspondence is found between the map task and the data block of the DataNode.

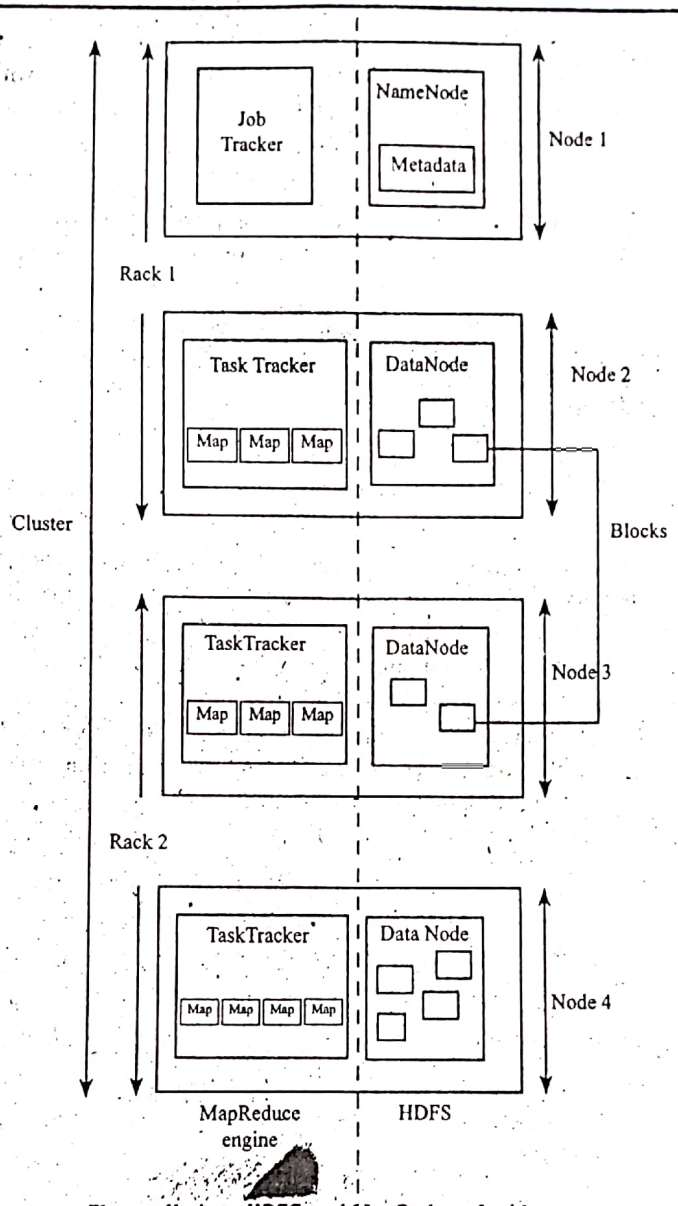


Figure: Hadoop HDFS and MapReduce Architecture

Running the Job in Hadoop: Components required to run a job in this system are user node, JobTracker and a set of TaskTrackers. The function runJob(conf) is called to begin the data flow in the user program. The conf is the parameter is an object for MapReduce framework and HDFs. This function is similar to MapReduce (spec & Results).

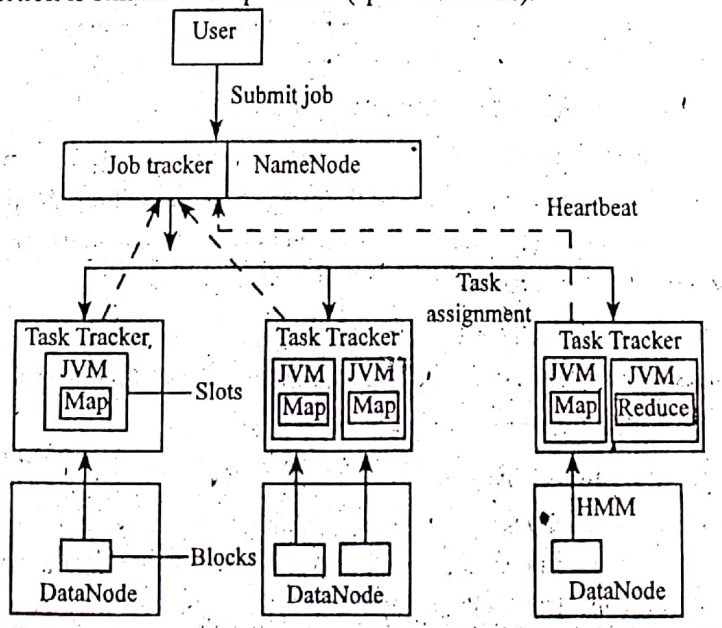


Figure: Using Hadoop Library for Data Flow to Run MapReduce Job at Different Task Trackers

4.3 PROGRAMMING SUPPORT OF GOOGLE APP ENGINE

Q21. Write how programming of GoogleAppEngine is done.

Ans: Programming Google App Engine: The key features of GAE(Google App Engine) programming model for languages such as Java and python is illustrated in the below figure.

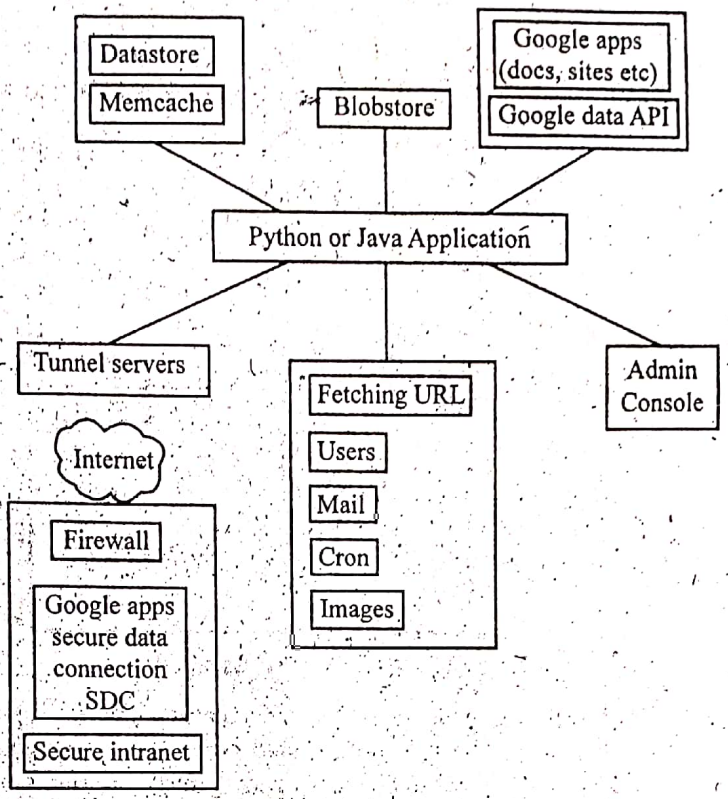


Figure: Google App Engine Programming Environment

The GAE is allowed to debugged on the local machine by the client environment that contains eclipse plug-in for Java. The Java web applications developers are provided with GWT(Google Web Toolkit). It can be used even for JavaScript or Rub. The language python is used with frameworks like Django and Cherrypy. Instead a webapp python environment is provided by Google. The data is stored and accessed using various constructs from the NOSQL data storage. The entities can be retrieved by queries through filtering and sorting the values. The JDO(Java Data Object) and JPA(Java Persistence API interfaces are offered by Java and implemented by open source Data Nucleus Access Platform. The python is provided with SQL-like query language called GQL.

The application is capable of executing various data store operations in one transaction the succeed or fail all together. The entities can be assigned to groups by GAE application. Google apended a new feature blobstore for heavy files.

The Google SDC(Secure Data Connection) can tunnel Using Internet and connect the Intranet to external GAE application. The URL Fetch operation will make the applications capable to fetch the resources and to interact with others on Internet through HTTP and HTTPs requests. It also accesses the web resources through high speed Google infrastructure to get the web pages for various products of Google.

Q22. Explain about Google File System.

Ans: Model Paper-II, Q5(b)

Google File System: Google File System(GFS) was designed as a storage service for Google's search Engine. It was basically designed to store and process huge amount of data needed by Google.

Google File System is a distributed file system that was developed to support Google applications. The reason for employing Google file system is that it is capable of holding a file of about 100MB. It basically partitions a file into fixed size segments called chunks. Each chunk provides a data block of about 64KB. Besides this it also ensures reliability of data by distributing replicate copies of data across multiple chunk server.

It also allow multiple append operations concurrently. It make use of a single master in order to provide access to metadata and simultaneously store the data. It provides an accessing interface similar to POSIX file system. This feature allow application to view the physical location of a file blocks. It also make use of customized API inorder to capture the append operation and also to record them.

The architecture of Google file system is shown below,

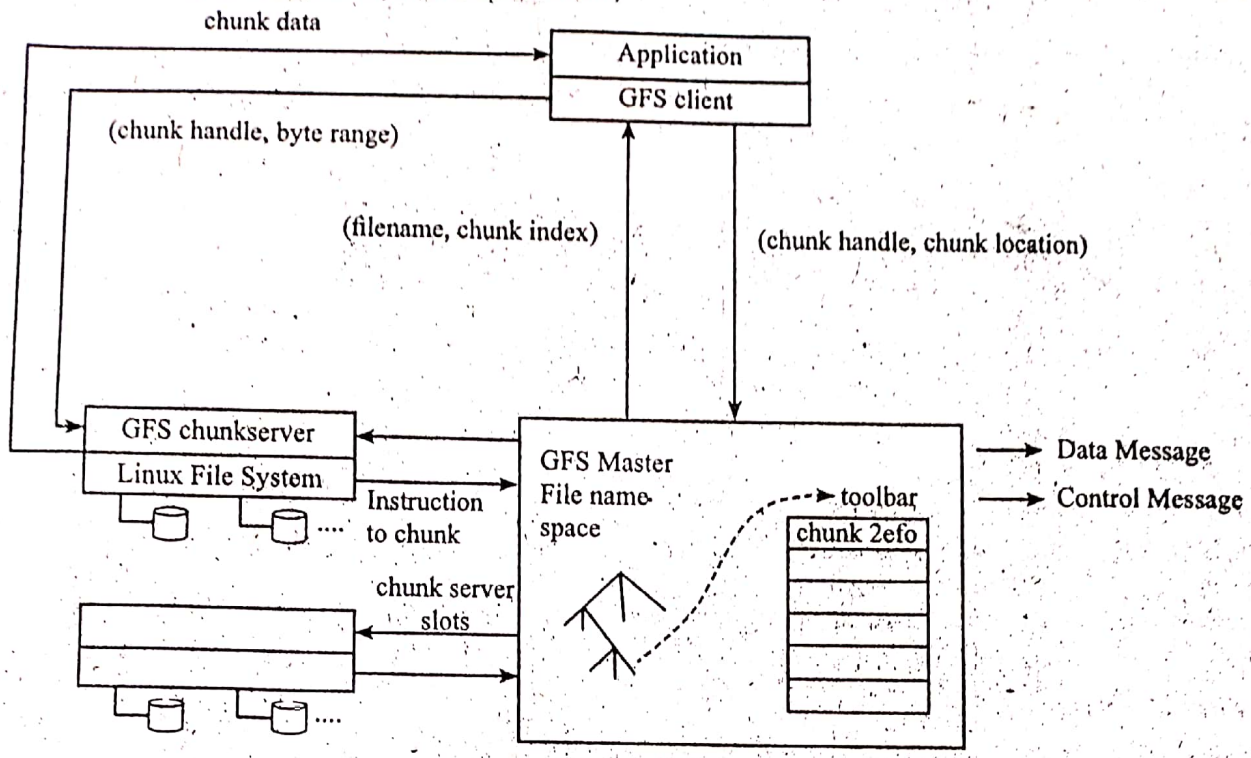


Figure: Google File System Architecture

The above architecture includes only single Master for storing Meta data in cluster. The different nodes act as a chunk servers. Each chunk server is responsible for storing data. A Master is also responsible for managing file system namespace and locking facilities. It also interact with chunk server in order to obtain the management information from them and also to instruct the chunk server to perform task like load balancing/fail recovery.

A single master is capable of managing the whole cluster. It (use of master) inhabit the use of complicated distributed algorithms in GFS architecture design. Despite of this, the inclusion of only single master may effect the performance of the system.

To overcome the performance bottle neck, shadow master are employed by Google. This master allow replication of data on master so as to ensure that data operation performed between client and chunk server are directly done without any further interruptions. Besides this, attached copy of control messages transferred between client and chunk server is maintained for further use. Hence, this facilities allow single master to manage a cluster containing 1,000 nodes. The figure illustrate the data mutations operations like write and append in GFS.

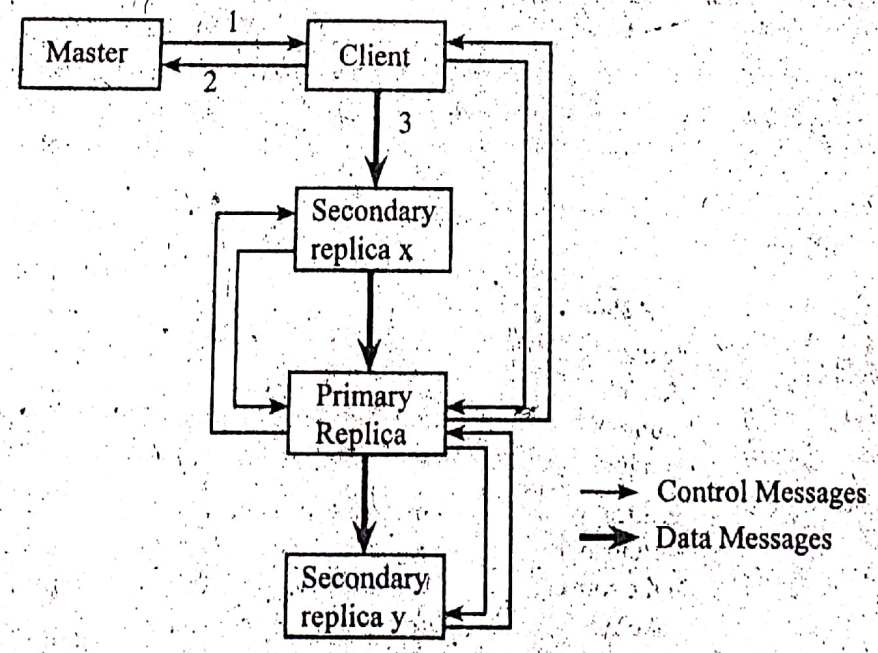


Figure: Data Mutation in Google File System

Data mulation must be performed by creating a data blocks for each replicated copy. The goal of Data mutation is to reduce the involvement of master in the cluster. The steps for performing mutation are as follows,

1. Initially the client query the master about the chunk server which contain the current chunk lease and also he location of the replicas.
2. If the chunk servers does not contain the chunk lease, then the master grants a single lease to the choosen replica and reply the client with a primary identity and the location of replica.
3. The client maintains the cached copy of data for future mutation.
4. The client then forward the data to other replicas. A chunk servers accepts the replicated data and maintains them in the internal LRU buffer cache. To improve the performance of system data flow is decoupled from control flow and also by scheduling the expensive data flow depending on the network technology.
5. After receiving the acknowledgment from the replicas. The client forwards the write request to the primary replica. The primary replica allot consecutive serial numbers to all the mutations received from multiple client and performs serilization on them. It then apply mutations to all the local states in serial order.
6. The write request are now forwarded to secondary replica by primary replica. The secondary replica now apply mutations is the order similar to primary.
7. The secondary replicas now reply the primary about the completion of operation.
8. The primary replica inturn replies to the client with all the errors that are encountered during mutation process. The code generated by client helps in recovering from errors and also in retrying failed mutation.

A GFS allow users to perform append operation. This operation allow users to append data block at the end of file.

GFS offers fast recover capability to recover from various system error. Besides this, it also ensures,

1. High availability
2. High performance
3. High fault tolerance and
4. High scalability.

Q23. Write in brief about the hierarchy of tablet location.

Ans: The hierarchy of the tablet location is depicted in the below figure,

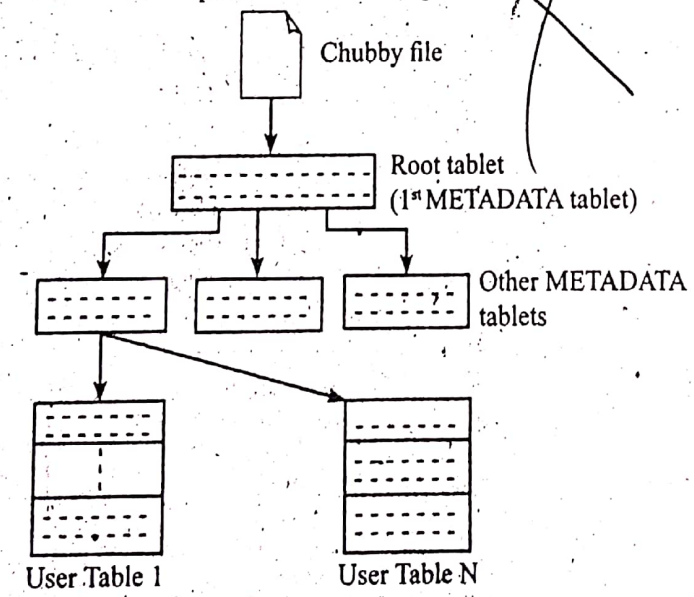


Figure: Tablet Location Hierarchy

In the above figure locating of Big Table data beginning from the files in chubby is shown. The file level contains the files of chubby containing the location of root tablet. This root tablet will contain the tables location in METADATA table. Every METADATA tablet holds a set of user tablets addresses. The root tablet is the first tablet in METADATA table and is special. It does not divide to ensure that the location hierarchy of tablet contains more than three levels.

The METADATA table stores the tablet location in a row key that is the encoded form of tablets table identifier along with its end row. The BigTable incorporates various optimizations as well as fault tolerant features. The chubby is responsible to ensure that file availability in order to locate the root tablet. The BigTable master is capable of scanning the tablet servers for determining the nodes status. These servers make use of compaction for storing the data in an efficient way. The logs that are shared are used for operation logging of various tablets to decrease the lop space and to maintain system consistency.

4.4 PROGRAMMING ON AMAZON AWS AND MICROSOFT AZURE

Q24. Explain how programming is done on Amazon. Model Paper-IV, Q5(b)

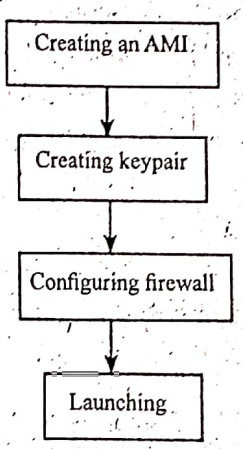
Ans: Amazon is the company that started generating VM's in application hosting. The customers rather than using the physical machines to run the applications will opt to rent VM's With VM's the customers are allowed to load any desired software. In addition to this they can also create, launch and even terminate the server instance by paying for them. Various types of VM's are provided by the amazon. The instance are called as Amazon Machine Images (AMI's) preconfigured with linux or windows in addition to the softwares. There are three types of AMI's defined as follows,

1. **Private AMI:** The images that are created by the users are called private AMI's. They are private by default and can be allowed to be launched by others.

2. **Public AMI:** The images that are created by users and released for AWS community by allowing others to launch and use the instances are called public AMI.

3. **Paid QAMI:** The images created by the users through certain functions and which are allowed to be launched by others by paying for them are called paid QAMI.

The below figure shows the execution environment where AMI's act as the templates for instance running VM's. The workflow for creating VM is as follows,



All the public, private and paid AMI's support the above procedure.

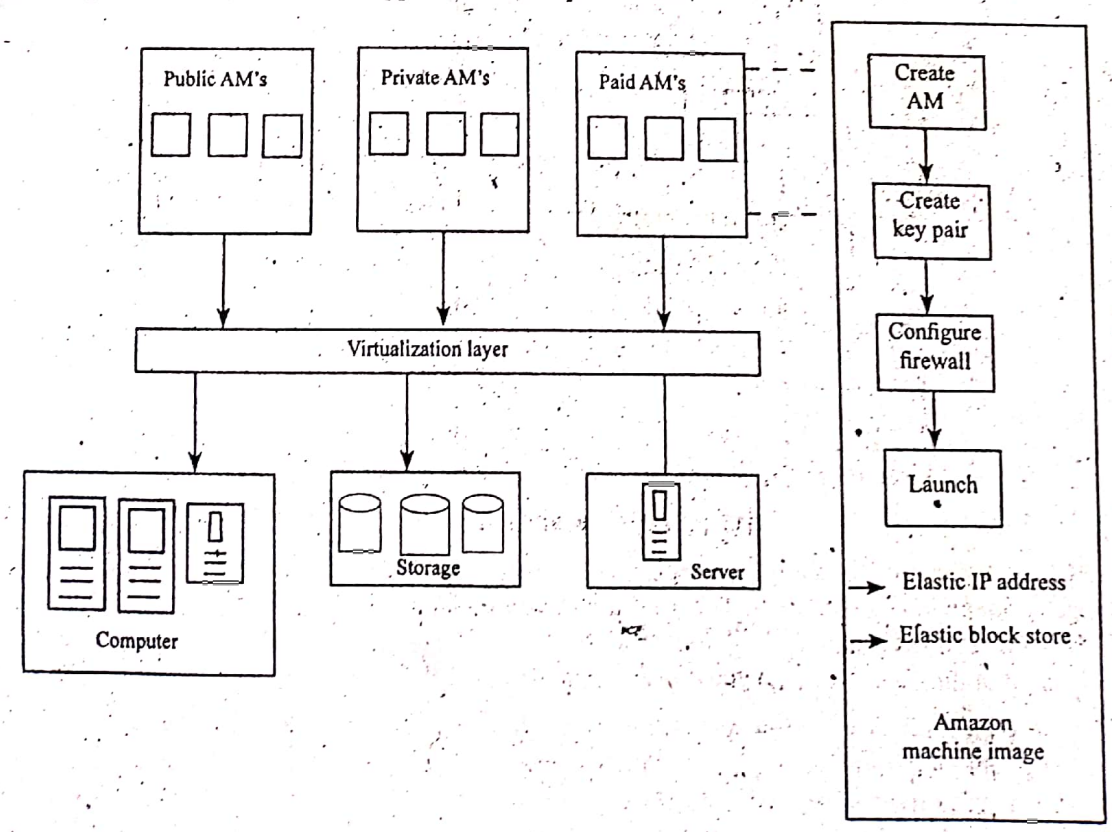


Figure: Execution Environment of Amazon EC2

Q25. Write a short notes on the following,

(i) Amazon Simple Storage Service (S3)

(ii) Amazon Elastic Block Store (EBS) and simpleDB.

Ans: Model Paper-III, Q5(a)

(i) Amazon Simple Storage Service (S3): Amazon S3 (Simple Storage Services) aims to provide an interface with simple web services for storing and retrieving the data on web at any time/any where. The service is in the form of object oriented storage. The objects are accessible to the users through SOAP (Simple Object Access Protocol) along with supporting browsers or client programs. A reliable message service among any two processes is provided by SQS. The below figure shows amazon S3 execution environment.

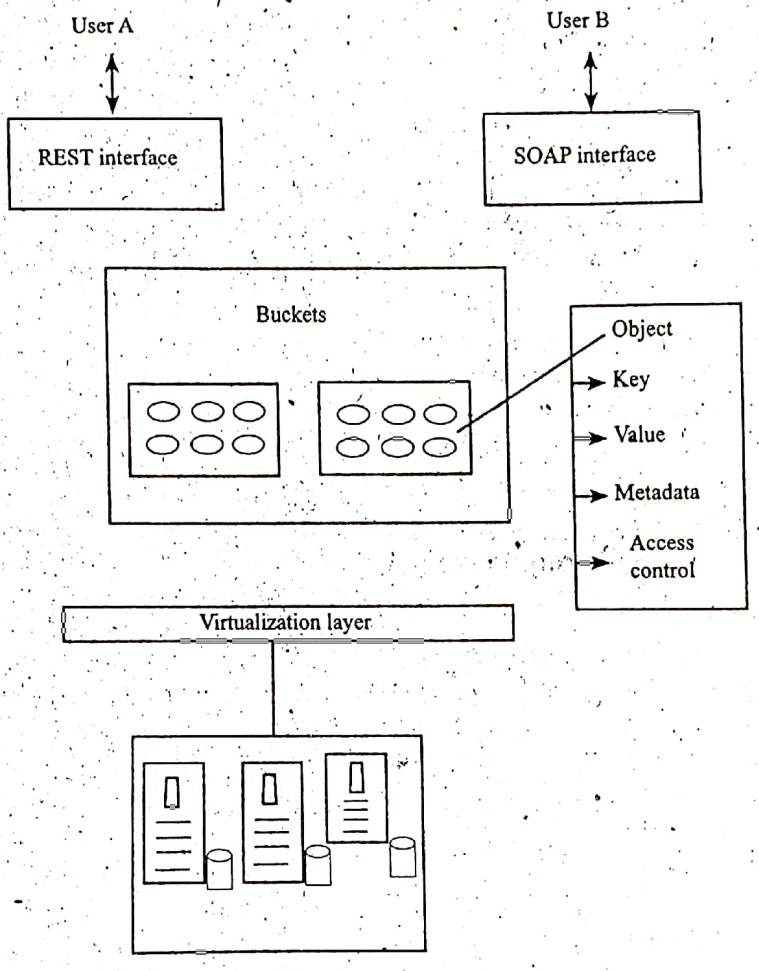


Figure: Execution Environment of Amazon S3

An object is the fundamental unit of the S3. A bucket holds various objects by accessing them through keys. There are even other attributes such as values, access control information and metadata. The users perform read, write and delete on objects through the key-value programming interface. Users can access the data from the amazon clouds through the interfaces REST and SOAP. Features of S3 are illustrated as follows,

- The authentication mechanisms are used to secure the data from unauthorized users. For this users are granted rights on objects by making them private or public.
- It contains URL's and ACL's for every object
- The cost is from $0.55 to 0.15 for every GB per month.
- Transfer cost out of S3 region is from $.08 to $0.15 per GB.
- Redundancy is maintained through geographic dispersion.
- An interface called BitTorrent protocol is used to decrease the cost of high scale distribution.
- It provides 99.999999999% durability and 99.99% availability of objects for a year with less RRSC Reduced Redundancy Storage).
- Data transfer between Amazon EC2 and S3 is not charged.

(ii) **Amazon EBS:** The amazon elastic block store offers a volume block interface to save as well as restore the EC2 instance's virtual images. Once the usage of traditional EC2 instances completes, they will be deleted. The status of them is saved in EBS system upon machine shutdown. The running data and EC2 instances are saved using EBS. The users are allowed to create the storage ranging from 1GB to 1TB. These volumes can be mounted as EC2 instances. Various volumes can be mounted as if they belong one instance. The user is allowed to create file system above amazon EBS volumes or to use in any desired way. Data saving is done through snapshots' to improve the performance. Amazon charges according to the usage.

Amazon SimpleDB Services: Amazon simpleDB service will provide a simple data model with respect to relational database data model. The user input is sorted into domains which are considered as tables. A table contains items as rows and attribute values as cells of the respective row. A single cell can also be assigned with multiple values.

The developers requires the data to be stored, accessed and queried easily. But this consistency is not considered by the simpleDB. The azure table manage less amounts of data in distributed table so they called s 'Little Table". The BigTable is supposed to store big data. The simpleDB costs $0.140 for each amazon simpleDB machine hours.

Q26. How mircrosoft Azure provides programming support?

Ans: Various features of azure cloud platform are programming components, SQL Azure, client development environment, storage and programming subsystems. They are depicted in the below figure.

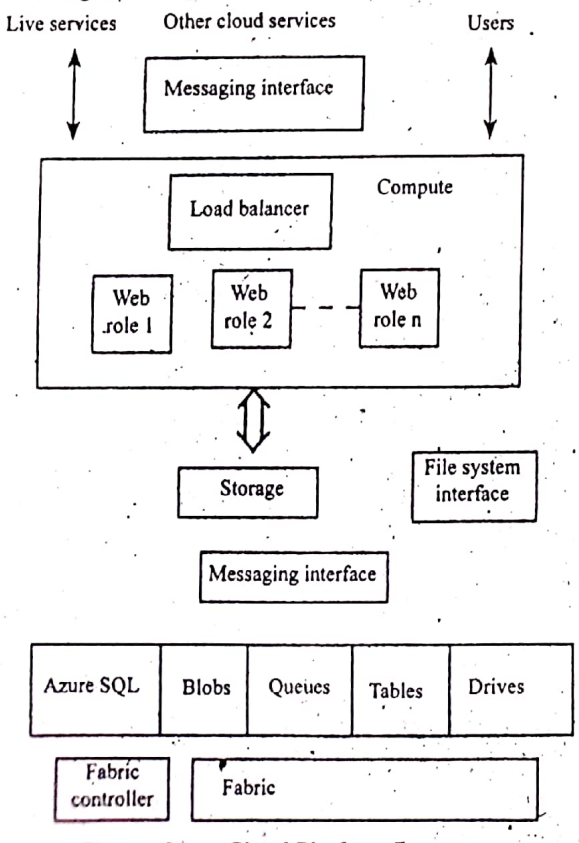


Figure: Azure Cloud Platform Features

The underneath layer in the above figure is fabric that contains virtualized hardware along with sophisticated control environment that dynamically assigns resources and implements fault tolerance. With this even domain name system and monitoring capabilities get implemented. Service models are allowed to be defined by XML templates and various service copies to be initialized.

The services are monitored while the system is running and the users are allowed to access the event logs, trace/debug the data. IIS web server logs, crash dumps, performance counters, crash dump and other files. Azure storage is responsible to hold all this data and debugging is allowed to be performed. The Azure is related to Internet through a customized compute VM known as web role that supports basic microsoft web hosting. These VM's are called appliances. The roles that support HTTPs and TCP provide below methods.

Onstar( ): This method is called fabric on startup that allows user to perform initialization tasks. It will show busy status to load balance until it is false.

Onstop( ): This method is invoked when the role is supported to be shutdown and then it exits.

Run( ): This method contains of main loge.

SQL Azure: The SQL server is provided as a service by SQL Azure. The REST interface is used to access the storage modalities excluding the drivers that are introduce most recently and which are analogous to amazon EBS. It provides a file system interface in the form of durable NTFS volume backed by blob storage. The interface REST are related with URL's by default. The storage gets duplicated three time due to fault tolerance and also guaranteed to be consistent in access.

The basic storage system gets emerged from blobs that are analogous to S3 for amazon. The blobs are classified as,

Account → Containers → Page/Block Blobs

The containers observed to be analogous to directories within traditional file systems where account is the root. The block blob streams the data and arranges them as sequence of blocks 4MB each upto 200 GB. The page blobs are intended for read/write access and contains set of pages with 1TB size.

Azure Tables: The azure table and queue storage modes are intended for less volumes of data. The queue will be offering reliable message delivery and also support work spooling among the web and worker roles. They does not restrict the messages. The azure supports the operations such as PUT, GET, DELETE as well as CREATE and DELETE. Every account is assigned infinite tables containing rows and columns in the form of entities and properties respectively.

The table entities are not restricted in number, rather huge number of entities of distributed computers. The general properties such as<name, type, values> are assigned to the entities. The other properties such as PartitionKey and RowKey can be assigned to entities. The purpose of RowKey is to assign unique label to every entry. The purpose of PartitionKey is to get shared.

4.5 EMERGING CLOUD SOFTWARE ENVIRONMENTS

Q27. Discuss in brief about open sourceEucalyptus and Nimbus. Model Paper-III, Q5(b)

Ans: Open Source Eucalyptus: Eucalyptus is a product of eucalyptus system which is a type of is a open software environment. It was developed form a research project at university of California, Santa Barbara. The purpose of it is to bring the cloud computing paradigm to academic supercomputers and clusters. To communicate with cloud service it provides an AWS-compliant EC2-based web service interface. Apart from this, it also provides services like AWS-compliant Walrus and user interface to manage users and images.

It also supports the development of computer cloud and storage cloud. It stores images in Walrus storage system which is similar to Amazon S3 service. It can be uploaded and retrived anytime. This helps users to create special virtual appliances. The below figure depicts the architecture that depends on VM images requirement.

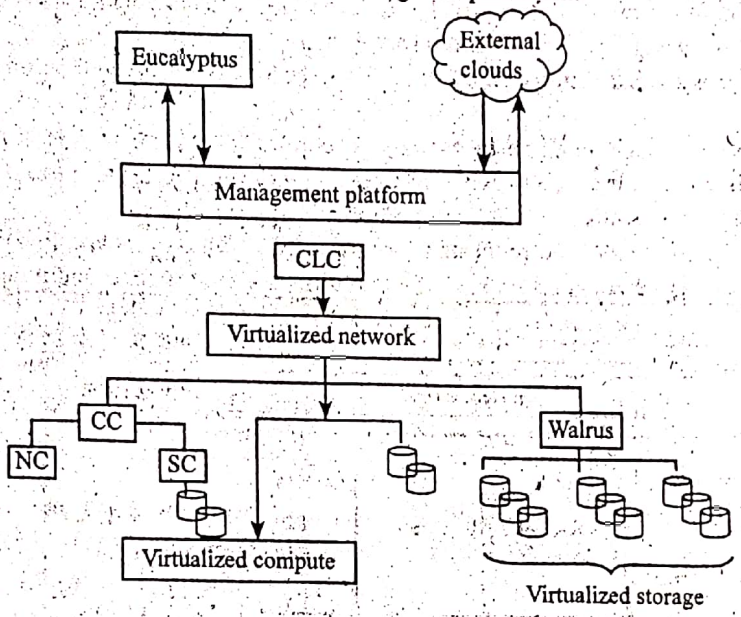
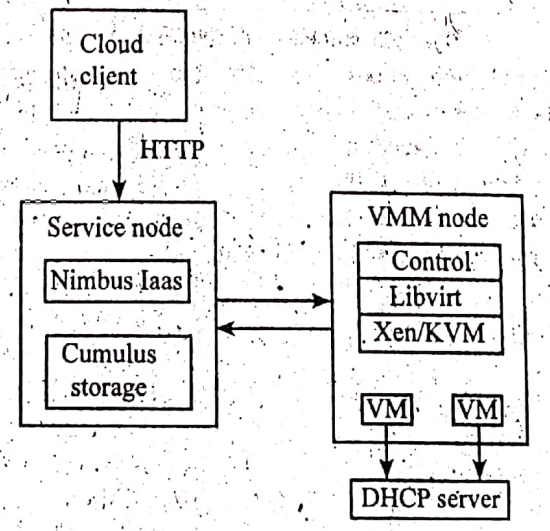


Figure: Eucalyptus Architecture for VM Image Management

Nimbus: Nimbus is a collection of open source tools which aim to offer an IaaS cloud computing solution. It provides a special web interface known as Nimbus Web. Which is placed around python Django Web application installed independent of Nimbus service. A storage cloud implementation known as Cumulus is combined with other central services and it is similar to Amazon S3 REST API.



Nimbus supports the below defined resource management strategies. They are,

1. Resource pool
2. Pilot

1. **Resource Pool:** It is a default resource pool mode which has a direct control on a pool of virtual machine manager nodes.
2. **Pilot:** In this mode the service requests cluster's Local Resource Management System for VM manager inorder to deploy VM's.

Nimbus implements Amazon's EC2 interface inorder to allow the users to make use of clients which are developed with the aim of real EC2 system against Nimbus-based clouds.

Q26. Write about OpenNebula and sector/sphere.

Ans:

Open Nebula: Open Nebula is an open source device that enables users to convert available infrastructure into an IaaS cloud. It is designed to be flexible and modular to merge with various storage and network infrastructure configurations and hypervisor technologies.

It consists of three components. They are as follows,

(a) Core

(b) Capacity manager (or) scheduler

(c) Access drivers.

(a) **Core:** It is a centralized component that controls the complete life cycle of virtual machine. It includes setting networks for set of virtual machines and controls the storage requirements like VM disk image deployment or software environment.

(b) **Capacity Manager:** It controls the working provided by core. It is a requirement or rank matchmaker. It is also used to develop scheduling polices using lease model and reservations.

(c) **Access Drivers:** It provides an abstraction of infrastructure to show the working of monitoring, storage and virtualization services available in cluster.

Apart from this, it provides management interfaces to merge core working with other data-center tools like accounting or monitoring frame works. It implements libvirt API and Command-Line Interface [CLI] for virtual machine management. It also consists two features for changing environment like live migration and VM snaphots.

It also includes EC2 driver, that can send requests to Amazon EC2, Eucalyptus and Elastic Hosts driver. In this image access control is used for images registered and it makes easy to multiuser environments and image sharing.

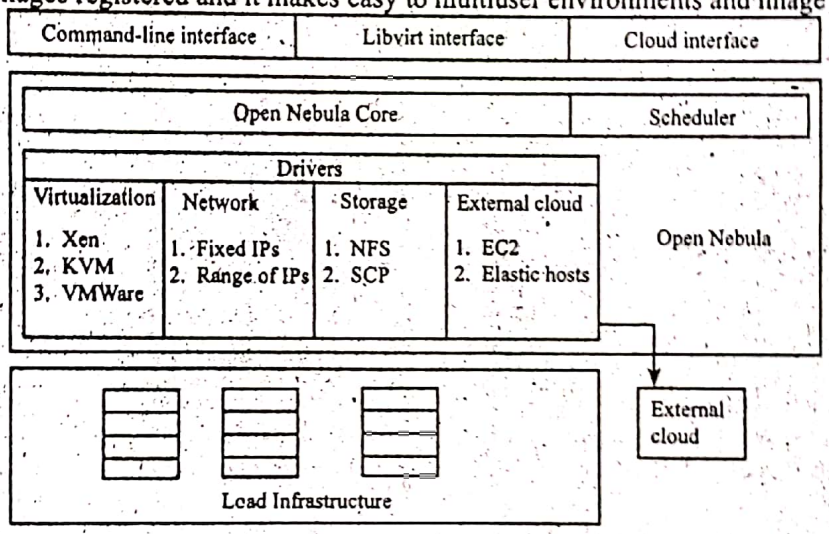


Figure: Architecture of OpenNebula

Sector/Sphere: Sector/sphere is a software which supports huge distributed data storage and data processing on large clusters, in data center or multiple data centers. It consists of sector distributed file system and sphere parallel data processing framework. By using the fast network connection the DFS is placed in large areas and enables user to control large data sets. Fault tolerance is performed by copying and controlling data in file system. It is also familiar with network topology and provides reliability, availability and access. In this communication is achieved using User-data Gram Protocol(UDP) and User-defined Type(UDT) i.e., UDP is used for message passing and UDT is used to transfer data.

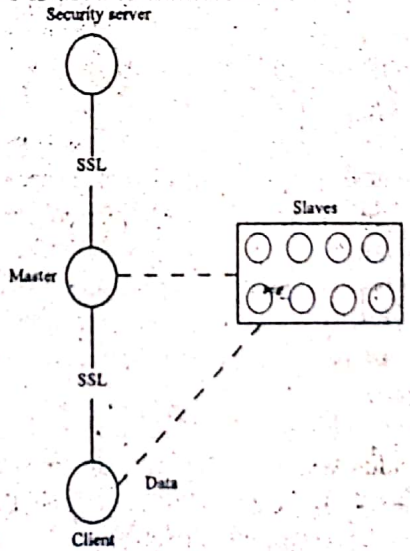


Figure: Architecture of Sector/Sphere

It is a parallel data processing designed to work with data controlled by sector. The data stored in a sector can be processed by the developers using a programming framework provided by sphere. In this application inputs and outputs are known as sector files. To support difficult applications, multiple sphere processing segments are merged.

It consists of four components. They are,

(a) Security server
(b) Slave nodes
(c) Client
(d) Space.

(a) **Security Server:** It is responsible for verifying master servers, slave nodes and users. This master server contains file system metadata, schedule jobs and responds to user's requests.

(b) **Slave Nodes:** It is used to store and execute the data. It can be placed in a single data center or multiple data centers with high-speed network connections.

(c) **Client:** It provides tools and programming APIs for accessing and processing the data.

(d) **Space:** It consists of a framework to support column-based distributed data tables. These tables are stored in the form of columns and are divided into multiple slave nodes. It supports a set of SQL operations.

Q29. Discuss about open stack.

Ans:

Open Stack: Open stack was introduced by Rack space and NASA in July 2010. It is used to share resources and technologies with a scalable and secure cloud infrastructure.

Features of Open stack are as follows,

(a) Open stack compute
(b) Open stack storage.

(a) **Open Stack Compute:** It is the internal fabric of cloud. It is used to create and control large sets of virtual private servers.

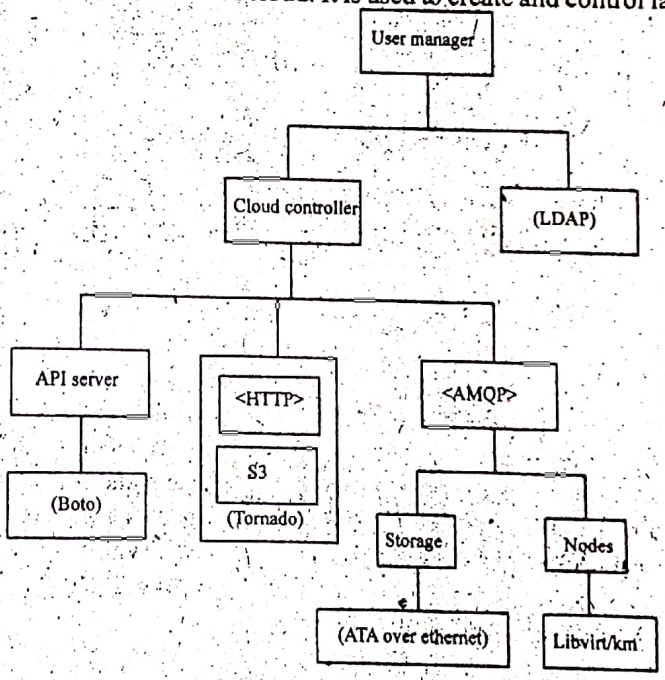


Figure: Architecture of Open Stack Nova System

It (developing) develops a cloud computing internal fabric controller, which is a part of an IaaS system called as Nova. It is built based on the idea of shared-nothing and exchange of message-based information. Communication is done (by through using) message queues. The components get blocked while interacting with each other thereby waiting for the response this can be prevented by using deferred objects. This object containing call backs that occur upon receiving response.

Shared-nothing paradigm can be achieved by changing the system state to distributed data system. In this architecture, the API server receivers HTTP requests from boto and it changes the commands to API format then, it forwards the requests to cloud controller. This cloud controller interacts with user manager with the help of Lightweight Directory Access Protocol (LDAP). Apart from this nova combines networking parts to control private networks, public IP addressing, VPN connectivity and firewall rules. It contains the following types.

1. **Network Controller:** It controls address and virtual LAN allocation.
2. **Routig Node:** It controls the network address translation and transforms public IPs and implements firewall rules.
3. **Addressing Node:** It runs Dynamic Host Configuration Protocol (DHCP) services for private networks.
4. **Tunneling Node:** It provides virtual private network connectivity.

Network state contains the following,

(i) **VAN Assignment:** It is for a project.

(ii) **Private Subnet Assignment:** It is for a security group in VLAN.

(iii) **Private IP Allocations:** It is for running instances.

(iv) **Public IP Allocations:** It is for a project.

(v) **Public IP Associations:** It is for private IP or running instance.

(b) Open Stack Storage: It generates solution based on interacting parts and concepts that consists a proxy server, ring, object server, container server, account server, replication, updates and auditors. This proxy server allows the accounts, containers or objects in this storage rings and route the requests. A ring shows the mapping between names of entities and their physical locations and it contains zones, devices partitions and replicas.

An object server is a simple blob storage server which is used to store, retrieve and delete objects which are on local devices. A container server is used for listing the objects and it is handled by the account server.

Q30. Explain in detail about Manjrasoft Aneka cloud and appliances. Model Paper-IV, Q5(a)

Ans: Manjra Soft Aneka Cloud and Appliances: Aneka is a cloud application platform that is developed by Manjrasoft. It aims to support the development and deployment of parallel and distributed applications on private and public clouds. It produces a collection of API's to utilize distributed resources and business logic applications through programming abstractions. System administrators control tools to observe and control deployed infrastructure. To increase the applications in both Linux and microsoft .NET framework it works as a workload distribution and management platform.

Merits of Aneka

1. It supports multiple programming and application environments.
2. It also supports multiple runtime environments.
3. It uses various virtual and physical machines to accelerate the application production depending upon the quality of service agreement (requirement) of the user.
4. It is layered upon microsoft. NET framework to support LINUX environment.