CLOUD COMPUTING

Introduction

Cloud computing is a type of computing that relies on shared computing resources rather than having local servers or personal devices to handle applications.

Definition by NIST Cloud Computing

The National Institute of Stands and Technology(NIST) has a more comprehensive definition of cloud computing. It describes cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

- Ability / space where you store your data ,process it and can access anywhere from the world
- As a Metaphor for the internet.

✓ Cloud computing is :
  - Storing data /Applications on remote servers
  - Processing Data / Applications from servers
  - Accessing Data / Applications via internet

What is a cloud service??

✓ Cloud computing is taking services and moving them outside an organization's firewall. Applications, storage and other services are accessed via the Web. The services are delivered and used over the Internet and are paid for by the cloud customer on an as-needed or pay-per-use business model.

Service: This term in cloud computing is the concept of being able to use reusable, fine-grained components across a vendor's network.

- Iaas,Paas,Saas,Daas,Naas,Caas are some of the services Provided by different providers

1.1 Characteristics (OR) Features of Cloud Environments:

According to the NIST, all true cloud environments have five key characteristics:

1. ***On-demand self-service:*** This means that cloud customers can sign up for, pay for and start using cloud resources very quickly on their own without help from a sales agent.
2. ***Broad network access:*** Customers access cloud services via the Internet.
3. ***Resource pooling:*** Many different customers (individuals, organizations or different departments within an organization) all use the same servers, storage or other computing resources.
4. ***Rapid elasticity or expansion:*** Cloud customers can easily scale their use of resources up or down as their needs change.
5. ***Measured service:*** Customers pay for the amount of resources they use in a given period of time rather than paying for hardware or software upfront. (Note that in a private cloud, this measured service usually involves some form of charge backs where IT keeps track of how many resources different departments within an organization are using.)

1.2 Applications:

i) Storage:cloud keeps many copies of storage. Using these copies of resources, it extracts another resource if anyone of the resources fails.

ii. Database: are repositories for information with links within the information that help making the data searchable.

Advantages:

i. **Improved availability:**If there is a fault in one database system, it will only affect one fragment of the information, not the entire database.
ii. **Improved performance:** Data is located near the site with the greatest demand and the database systems are parallelized, which allows the load to be balanced among the servers.
iii. **Price** It is less expensive to create a network of smaller computers with the power of one large one.
iv. **Flexibility :** Systems can be changed and modified without harm to the entire database.

Disadvantages:

i. **Complexity** Database administrators have extra work to do to maintain the system.
ii. **Labor costs** With that added complexity comes the need for more workers on the payroll.
iii. **Security** Database fragments must be secured and so must the sites housing the fragments.
iv. **Integrity** It may be difficult to maintain the integrity of the database if it is too complex or changes too quickly.

v. **Standards** There are currently no standards to convert a centralized database into a cloud solution.

iii. Synchronization→allows content to be refreshed across multiple devices.

Ex:

Google docs

Data base services (DaaS): it avoids the complexity and cost of running your own database.

Benefits:

i.Ease of use :don't have to worry about buying, installing, and maintaining hardware for the database as there is no servers to provision and no redundant systems to worry..

ii. **Power** The database isn't housed locally, but that doesn't mean that it is not functional and effective. Depending on your vendor, you can get custom data validation to ensure accurate information. You can create and manage the database with ease.

iii. **Integration** The database can be integrated with your other services to provide more value and power. For instance, you can tie it in with calendars, email, and people to make your work more powerful.

iv. **Management** because large databases benefit from constant pruning and optimization, typically there are expensive resources dedicated to this task. With some DaaS offerings, this management can be provided as part of the service for much less expense. The provider will often use offshore labor pools to take Advantage of lower labor costs there. So it's possible that you are using the service in Chicago, the physical servers are in Washington state, and the database administrator is in the Philippines.

- MS SQL and Oracle are two biggest players of DaaS providers.

MS SQL:

- Microsoft SQL server data services (SSDS),SSDS based on SQL server, announced cloud extension of SQL server tool, in 2008 which is similar to Amazon's simple database (schema –free data storage, SOAP or REST APIs and a pay-as-you-go payment system.
- Variation is first, one of the main selling points of SSDS is that it integrates with Microsoft's sync Framework which is a .NET library for synchronizing dissimilar data sources.
- Microsoft wants SSDS to work as a data hub, synchronizing data on multiple devices so they can be accessed offline.

Core concepts in SSDS:

i. Authority → both a billing unit and a collection of containers
ii. Container → collection of entities and is what you search within.

iii. Entity → property bag of name and value pairs

Oracle:

It introduces three services to provide database services to cloud users. Customers can license

a. Oracle Database 11g
b. Oracle fusion Middleware
c. Oracle enterprise Manager

- **AWS EC2**-Amazon web services Elastic Compute cloud

Oracle delivered a set of free Amazon Machine Images (AMIs) to its customers so they could quickly and efficiently deploy Oracle's database solutions.
Developers can take advantage of the provisioning and automated software deployment to rapidly build applications using Oracle's popular development tools such as Oracle Application Express, Oracle Developer, Oracle Enterprise Pack for Eclipse, and Oracle Workshop for Web Logic. Additionally, Oracle Unbreakable Linux Support and AWS Premium Support is available for Oracle Enterprise Linux on EC2, providing seamless customer support.
"Providing choice is the foundation of Oracle's strategy to enable customers to become more productive and lower their IT costs—whether it's choice of hardware, operating system, or on demand computing—extending this to the Cloud environment is a natural evolution," said Robert Shimp, vice president of Oracle Global Technology Business Unit.
"We are pleased to partner with Amazon Web Services to provide our customers enterpriseclass. Cloud solutions, using familiar Oracle software on which their businesses depend."
Additionally, Oracle also introduced a secure cloud-based backup solution. Oracle Secure Backup Cloud Module, based on Oracle's premier tape backup management software, Oracle Secure Backup, enables customers to use the Amazon Simple Storage Service (Amazon S3) as their database backup destination. Cloud-based backups offer reliability and virtually unlimited capacity, available on demand and with no up-front capital expenditure.
The Oracle Secure Backup Cloud Module also enables encrypted data backups to help ensure complete privacy in the cloud environment. It's fully integrated with Oracle Recovery Manager and Oracle Enterprise Manager, providing users with familiar interfaces for cloud-based backups.
For customers with an ongoing need to quickly move very large volumes of data into or out of the AWS cloud, Amazon allows the creation of network peering connections.

1.3 Cloud Components:

Three components of a cloud computing are :

- **Clients**
- **Data center**
- **Distributed servers**

i. Clients:

- Clients are the devices that the end users interact with to manage their information on the cloud.
- Clients are of three categories :

a. Mobile: mobile devices including PDAs/smart phones like a blackberry, windows, iphone.

b. Thin: are comps that don't have internal hard drives then display the info but rather let server do all the work.

c. Thick: is a regular comp, using web browser like Firefox/Internet Explorer to connect to the cloud.

Thin Vs Thick

i. Price and effect environment
ii. Lower hardware costs
iii. Lower IT costs
iv. Security
v. Data Security
vi. Less Power consumption
vii. Ease of repair or replacement
viii. Less noise

ii. Data Center :

- It is a collection of servers where the application you subscribe and housed.

iii. Distributed Servers:

- Servers are in geographically disparate locations but act as if they're humming away right next to each other.
- This gives the service provider more flexibility in options and security.

EX :

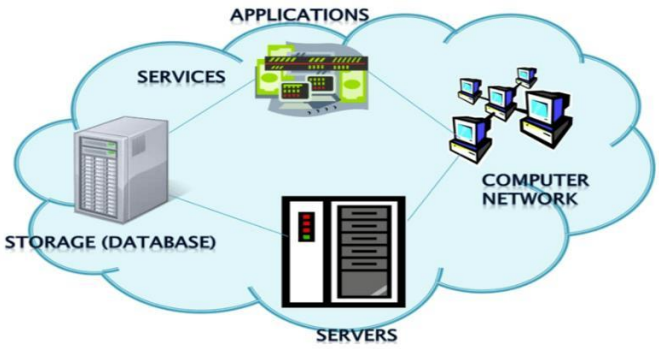
Amazon has their cloud solution all over the world ,if one failed at one site the service would still be accessed through another site

- If cloud needs more h/w they need not throw more servers in the safe room –they can add them at another site and make it part of the cloud.

1.4 Benefits and Limitations of Cloud Computing

The **advantage of cloud computing** is twofold. It is a file backup shape. It also allows working on the same document for several jobs (one person or a nomad traveling) of various types (or PC, tab or smart phone).

Cloud computing simplifies usage by allowing overcoming the constraints of traditional computer tools (installation and updating of software, storage, data portability...). Cloud computing also provides more elasticity and agility because it allows faster access to IT resources (server, storage or bandwidth) via a simple web portal and thus without investing in additional hardware.



Consumers and organizations have many different reasons for choosing to use cloud computing services. They might include the following:

- Convenience
- Scalability
- Low costs
- Security
- Anytime, anywhere access
- High availability

Limitations /Disadvantages:

a) Down time: Since cloud computing systems are internet-based, service outages are always an unfortunate possibility and can occur for any reason.

Best Practices for minimizing planned downtime in a cloud environment:

ii. Design services with high availability and disaster recovery in mind. Leverage the multi-availability zones provided by cloud vendors in your infrastructure.

iii. If your services have a low tolerance for failure, consider multi-region deployments with automated failover to ensure the best business continuity possible.

iv. Define and implement a disaster recovery plan in line with your business objectives that provide the lowest possible recovery time (RTO) and recovery point objectives (RPO).

v. Consider implementing dedicated connectivity such as AWS Direct Connect, Azure Express Route, or Google Cloud's Dedicated Interconnect or Partner Interconnect. These services provide a dedicated network connection between you and the cloud service point of presence. This can reduce exposure to the risk of business interruption from the public internet.

b) Security and Privacy: Code Space and the hacking of their AWS EC2 console, which led to data deletion and the eventual shutdown of the company. Their dependence on remote cloud-based infrastructure meant taking on the risks of outsourcing everything.

Best practices for minimizing security and privacy risks:

- Understand the shared responsibility model of your cloud provider.
- Implement security at every level of your deployment.
- Know who is supposed to have access to each resource and service and limit access to least privilege.
- Make sure your team's skills are up to the task: Solid security skills for your cloud teams are one of the best ways to mitigate security and privacy concerns in the cloud.
- Take a risk-based approach to securing assets used in the cloud Extend security to the device.
- Implement multi-factor authentication for all accounts accessing sensitive data or systems.

c) Vulnerability to Attack: Even the best teams suffer severe attacks and security breaches from time to time.

Best practices to help you reduce cloud attacks:

- Make security a core aspect of all IT operations.
- Keep ALL your teams up to date with cloud security best practices.
- Ensure security policies and procedures are regularly checked and reviewed.
- Proactively classify information and apply access control.
- Use cloud services such as AWS Inspector, AWS CloudWatch, AWS CloudTrail, and AWS Config to automate compliance controls.
- Prevent data ex-filtration.

- Integrate prevention and response strategies into security operations.
- Discover rogue projects with audits.
- Remove password access from accounts that do not need to log in to services.
- Review and rotate access keys and access credentials.
- Follow security blogs and announcements to be aware of known attacks.
- Apply security best practices for any open source software that you are using.

d) Limited control and flexibility: Since the cloud infrastructure is entirely owned, managed and monitored by the service provider, it transfers minimal control over to the customer.
To varying degrees (depending on the particular service), cloud users may find they have less control over the function and execution of services within a cloud-hosted infrastructure. A cloud provider's end-user license agreement (EULA) and management policies might impose limits on what customers can do with their deployments. Customers retain control of their applications, data, and services, but may not have the same level of control over their backend infrastructure.
Best practices for maintaining control and flexibility:

- Consider using a cloud provider partner to help with implementing, running, and supporting cloud services.
- Understanding your responsibilities and the responsibilities of the cloud vendor in the shared responsibility model will reduce the chance of omission or error.
- Make time to understand your cloud service provider's basic level of support. Will this service level meet your support requirements? Most cloud providers offer additional support tiers over and above the basic support for an additional cost.
- Make sure you understand the service level agreement (SLA) concerning the infrastructure and services that you're going to use and how that will impact your agreements with your customers.

e) Vendor Lock-In: organizations may find it difficult to migrate their services from one vendor to another. Differences between vendor platforms may create difficulties in migrating from one cloud platform to another, which could equate to additional costs and configuration complexities.
Best practices to decrease dependency:

- Design with cloud architecture best practices in mind. All cloud services provide the opportunity to improve availability and performance, decouple layers, and reduce performance bottlenecks. If you have built your services using cloud architecture best practices, you are less likely to have issues porting from one cloud platform to another.
- Properly understanding what your vendors are selling can help avoid lock-in challenges.
- Employing a multi-cloud strategy is another way to avoid vendor lock-in. While this may add both development and operational complexity to your deployments, it doesn't have to be a deal breaker. Training can help prepare teams to architect and select best-fit services and technologies.

- Build in flexibility as a matter of strategy when designing applications to ensure portability now and in the future.

f) Costs Savings: Adopting cloud solutions on a small scale and for short-term projects can be perceived as being expensive.

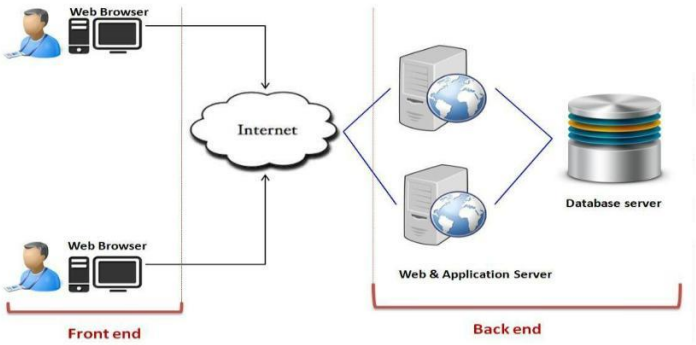
Best practices to reduce costs:

- Try not to over-provision, instead of looking into using auto-scaling services
- Scale DOWN as well as UP
- Pre-pay if you have a known minimum usage
- Stop your instances when they are not being used
- Create alerts to track cloud spending

1.5 Architecture

Let's have a look into Cloud Computing and see what Cloud Computing is made of. Cloud computing comprises of two components front end and back end. **Front end** consist client part of cloud computing system. It comprise of interfaces and applications that are required to access the cloud computing platform.

A central server administers the system, monitoring traffic and client demands to ensure everything runs smoothly. It follows a set of rules called protocols and uses a special kind of software called **MIDDLEWARE.** Middleware allows networked computers to communicate with each other. Most of the time, servers don't run at full capacity. That means there's unused processing power going to waste. It's possible to fool a physical server into thinking it's actually multiple servers, each running with its own independent operating system. The technique is called **server virtualization**. By maximizing the output of individual servers, server virtualization reduces the need for more physical machines.



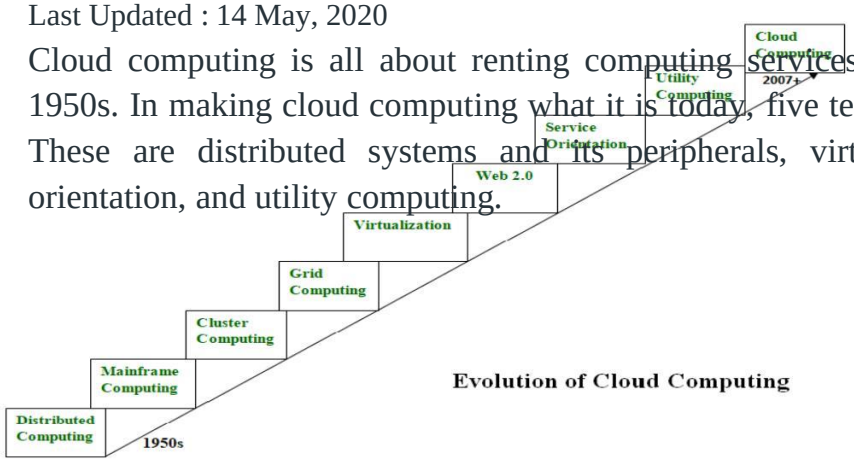
While **back end** refers to the cloud itself, it comprises of the resources that are required for cloud computing services. It consists of virtual machines, servers, data storage, security mechanism etc. It is under provider's control.

Cloud computing distributes the file system that spreads over multiple hard disks and machines. Data is never stored in one place only and in case one unit fails the other will take over automatically. The user disk space is allocated on the distributed file system, while another important component is algorithm for resource allocation. Cloud computing is a strong distributed environment and it heavily depends upon strong algorithm.

Evolution of Cloud Computing

- Difficulty Level : Easy
- Last Updated : 14 May, 2020

Cloud computing is all about renting computing services. This idea first came in the 1950s. In making cloud computing what it is today, five technologies played a vital role. These are distributed systems and its peripherals, virtualization, web 2.0, service orientation, and utility computing.



Evolution of Cloud Computing

- **Distributed Systems:**
  It is a composition of multiple independent systems but all of them are depicted as a single entity to the users. The purpose of distributed systems is to share resources and also use them effectively and efficiently. Distributed systems possess characteristics such as scalability, concurrency, continuous availability, heterogeneity, and independence in failures. But the main problem with this system was that all the

systems were required to be present at the same geographical location. Thus to solve this problem, distributed computing led to three more types of computing and they were-Mainframe computing, cluster computing, and grid computing.

- **Mainframe computing:**
  Mainframes which first came into existence in 1951 are highly powerful and reliable computing machines. These are responsible for handling large data such as massive input-output operations. Even today these are used for bulk processing tasks such as online transactions etc. These systems have almost no downtime with high fault tolerance. After distributed computing, these increased the processing capabilities of the system. But these were very expensive. To reduce this cost, cluster computing came as an alternative to mainframe technology.

  Cluster computing:
  In 1980s, cluster computing came as an alternative to mainframe computing. Each machine in the cluster was connected to each other by a network with high bandwidth. These were way cheaper than those mainframe systems. These were equally capable of high computations. Also, new nodes could easily be added to the cluster if it was required. Thus, the problem of the cost was solved to some extent but the problem related to geographical restrictions still pertained. To solve this, the concept of grid computing was introduced.

- **Grid computing:**
  In 1990s, the concept of grid computing was introduced. It means that different systems were placed at entirely different geographical locations and these all were connected via the internet. These systems belonged to different organizations and thus the grid consisted of heterogeneous nodes. Although it solved some problems but new problems emerged as the distance between the nodes increased. The main problem which was encountered was the low availability of high bandwidth connectivity and with it other network associated issues. Thus. cloud computing is often referred to as "Successor of grid computing".

- **Virtualization:**
  It was introduced nearly 40 years back. It refers to the process of creating a virtual layer over the hardware which allows the user to run multiple instances simultaneously on the hardware. It is a key technology used in cloud computing. It is the base on which major cloud computing services such as Amazon EC2, VMware

vCloud, etc work on. Hardware virtualization is still one of the most common types of virtualization.

- **Web 2.0:**
  It is the interface through which the cloud computing services interact with the clients. It is because of Web 2.0 that we have interactive and dynamic web pages. It also increases flexibility among web pages. Popular examples of web 2.0 include Google Maps, Facebook, Twitter, etc. Needless to say, social media is possible because of this technology only. In gained major popularity in 2004.

- **Service orientation:**
  It acts as a reference model for cloud computing. It supports low-cost, flexible, and evolvable applications. Two important concepts were introduced in this computing model. These were Quality of Service (QoS) which also includes the SLA (Service Level Agreement) and Software as a Service (SaaS).

- **Utility computing:**
  It is a computing model that defines service provisioning techniques for services such as compute services along with other major services such as storage, infrastructure, etc which are provisioned on a pay-per-use basis.

Virtualization and Cloud Computing

The main enabling technology for Cloud Computing is **VIRTUALIZATION**. Virtualization is a partitioning of single physical server into multiple logical servers. Once the physical server is divided, each logical server behaves like a physical server and can run an operating system and applications independently. Many popular companies like VMware and Microsoft provide virtualization services, where instead of using your personal PC for storage and computation, you use their virtual server. They are fast, cost-effective and less time consuming.

For software developers and testers virtualization comes very handy, as it allows developer to write code that runs in many different environments and more importantly to test that code.

Virtualization is mainly used for three main purposes

1) Network Virtualization

2) Server Virtualization

3) Storage Virtualization

Network Virtualization: It is a method of combining the available resources in a network by splitting up the available bandwidth into channels, each of which is independent from the others and each channel is independent of others and can be assigned to a specific server or device in real time.

Storage Virtualization: It is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device that is managed from a central console. Storage virtualization is commonly used in storage area networks (SANs).

Server Virtualization: Server virtualization is the masking of server resources like processors, RAM, operating system etc, from server users. The intention of server virtualization is to increase the resource sharing and reduce the burden and complexity of computation from users.

Virtualization is the key to unlock the Cloud system, what makes virtualization so important for the cloud is that it decouples the software from the hardware. For example, PC's can use virtual memory to borrow extra memory from the hard disk. Usually hard disk has a lot more space than memory. Although virtual disks are slower than real memory, if managed properly the substitution works perfectly. Likewise, there is software which can imitate an entire computer, which means 1 computer can perform the functions equals to 20 computers.

1.6 Classification of Cloud Variants:

i. **Service Model Based**
ii. **Deployment Model Based**

1.6.1 Service Model Based /Models Service / Types of Models

Cloud computing services are divided into three classes, according to the abstraction level of the capability provided and the service model of providers, namely:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS) and
3. Software as a Service. (SaaS)

These abstraction levels can also be viewed as a layered architecture where services of a higher layer can be composed from services of the underlying layer. The reference model explains the role of each layer in an integrated architecture. A core middleware manages physical resources and the VMs deployed on top of them; in addition, it provides the required features (e.g., accounting and billing) to offer multi-tenant pay-as-you-go services.

Cloud development environments are built on top of infrastructure services to offer application development and deployment capabilities; in this level, various programming models, libraries, APIs, and mashup editors enable the creation of a range of business, Web, and scientific applications. Once deployed in the cloud, these applications can be consumed by end users.

INFRASTRUCTURE AS A SERVICE

Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS).

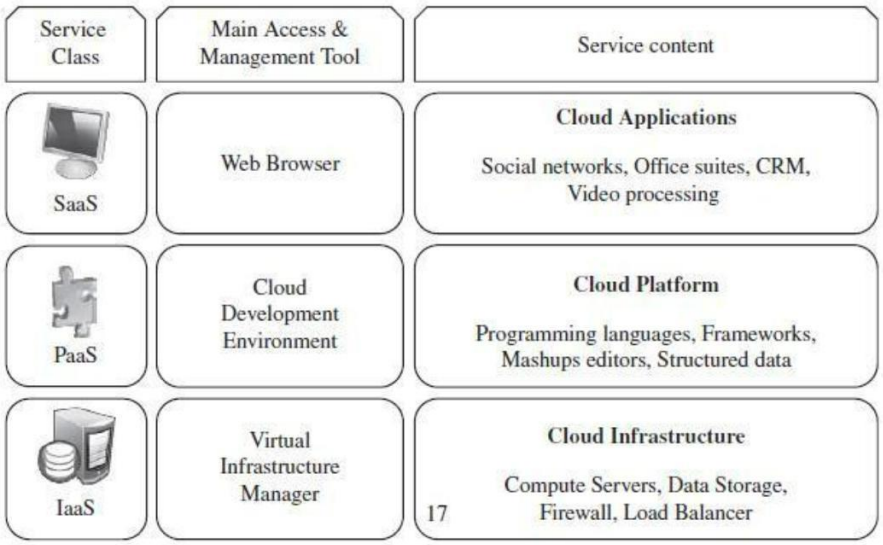


FIGURE 1.3. The cloud computing stack.

A cloud infrastructure enables on-demand provisioning of servers running several choices of operating systems and a customized software stack. Infrastructure services are considered to be the bottom layer of cloud computing systems.

- Amazon Web Services mainly offers IaaS, which in the case of its EC2 service means offering VMs with a software stack that can be customized similar to how an ordinary physical server would be customized.

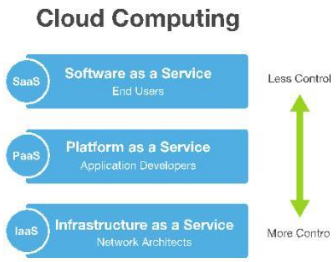
- Users are given privileges to perform numerous activities to the server, such as: starting and stopping it, customizing it by installing software packages, attaching virtual disks to it, and configuring access permissions and firewalls rules.

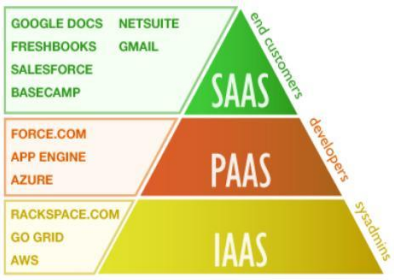
PLATFORM AS A SERVICE

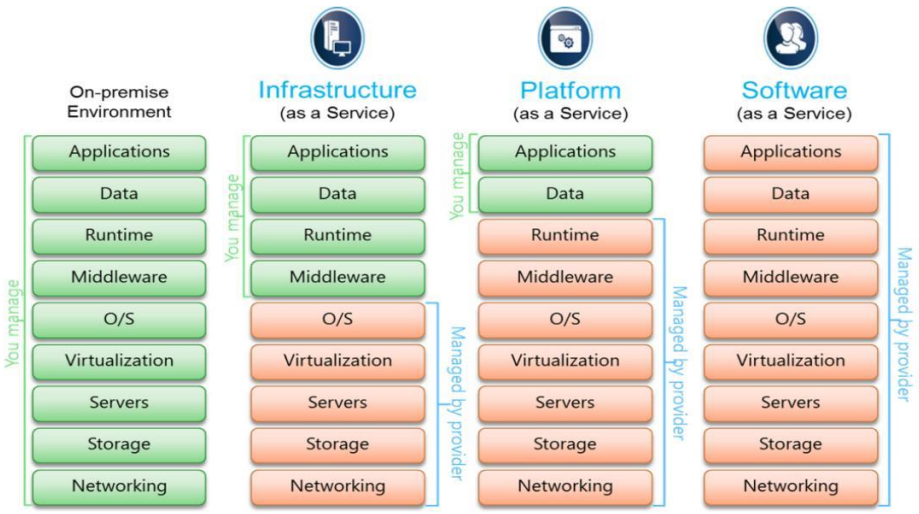
In addition to infrastructure-oriented clouds that provide raw computing and storage services, another approach is to offer a higher level of abstraction to make a cloud easily programmable, known as Platform as a Service (PaaS).

A cloud platform offers an environment on which developers create and deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. In addition, multiple programming models and specialized services (e.g., data access, authentication, and payments) are offered as building blocks to new applications.

Google App Engine, an example of Platform as a Service, offers a scalable environment for developing and hosting Web applications, which should be written in specific programming languages such as Python or Java, and use the services' own proprietary structured object data store. Building blocks include an in-memory object cache (mem cache), mail service, instant messaging service (XMPP), an image manipulation service, and integration with Google Accounts authentication service. Software as a Service Applications reside on the top of the cloud stack. Services provided by this layer can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally. Traditional desktop applications such as word processing and spreadsheet can now be accessed as a service in the Web. This model of delivering applications, known as Software as a Service (F), alleviates the burden of software maintenance for customers and simplifies development and testing for providers. Salesforce.com, which relies on the SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing customers to customize and access applications on demand.







c. Infrastructure as a Service (IaaS) or Hardware as a Service (HaaS)

INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, running several choices of operating systems and a customized software stack.

FEATURES

The most relevant features are:

i. Geographic distribution of data centers;
ii. Variety of user interfaces and APIs to access the system;

a. Specialized components and services that aid particular applications (e.g., load- balancers, firewalls);
b. Choice of virtualization platform and operating systems; and
c. Different billing methods and period (e.g., prepaid vs. postpaid, hourly vs. monthly).

Geographic Presence:

Availability zones are ―distinct locations that are• engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other

availability zones in the same region. Regions, in turn, ―are geographically dispersed and will be in separate geographic areas or countries.

User Interfaces And Access To Servers:

A public IaaS provider must provide multiple access means• to its cloud, thus catering for various users and their preferences. Different types of user interfaces (UI) provide different levels of abstraction, the most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs. GUIs are preferred by end users who need to launch,• customize, and monitor a few virtual servers and do not necessary need to repeat the process several times.

Advance Reservation Of Capacity:

Advance reservations allow users to request for an IaaS• provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time. Amazon Reserved Instances is a form of advance• reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability at anytime during an agreed period and then paying a discounted hourly rate when resources are in use.
Automatic Scaling And Load Balancing:

It allow users to set conditions for when they want their• applications to scale up and down, based on application specific metrics such as transactions per second, number of simultaneous users, request latency, and so forth. When the number of virtual servers is increased by• automatic scaling, incoming traffic must be automatically distributed among the available servers. This activity enables applications to promptly respond to traffic increase while also achieving greater fault tolerance.

Service-Level Agreement:

Service-level agreements (SLAs) are offered by IaaS• providers to express their commitment to delivery of a certain QoS. To customers it serves as a warranty. An SLA usually include availability and performance guarantees. HYPERVISOR AND OPERATING SYSTEM CHOICE: IaaS offerings have been based on heavily customized• open-source Xen deployments. IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings.

PaaS Providers

Public Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low-level details of the platform.

FEATURES

Programming Models, Languages, and Frameworks. Programming models made available by IaaS providers define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform and recover it in case of crashes, as well as to store user data.

***Persistence Options*.** A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data.

1.6.2 Deployment Model Based/Types of CC / Cloud Delivery Models:

Cloud computing can be divided into several sub-categories depending on the physical location of the computing resources and who can access those resources.

a. Public cloud vendors offer their computing services to anyone in the general public. They maintain large data centers full of computing hardware, and their customers share access to that hardware.

b. *Private cloud* is a cloud environment set aside for the exclusive use of one organization. Some large enterprises choose to keep some data and applications in a private cloud for security reasons, and some are required to use private clouds in order to comply with various regulations.

Organizations have two different options for the location of a private cloud: they can set up a private cloud in their own data centers or they can use a hosted private cloud service. With a hosted private cloud, a public cloud vendor agrees to set aside certain computing resources and allow only one customer to use those resources.

c. *Hybrid cloud* is a combination of both a public and private cloud with some level of integration between the two. For example, in a practice called "cloud bursting" a company may run Web servers in its own private cloud most of the time and use a public cloud service for additional capacity during times of peak use.

A multi-cloud environment is similar to a hybrid cloud because the customer is using more than one cloud service. However, a multi-cloud environment does not necessarily have integration among the various cloud services, the way a hybrid cloud does. A multi-cloud environment can

include only public clouds, only private clouds or a combination of both public and private clouds.

d. Community Cloud: Here**,** computing resources are provided for a community and organizations.

1.7 Infrastructure of Cloud Computing

- Cloud infrastructure means the hardware and software components.
- These components are server, storage, and networking and virtualization software.
- These components are required to support the computing requirements of a cloud computing model.

Components of Cloud infrastructure

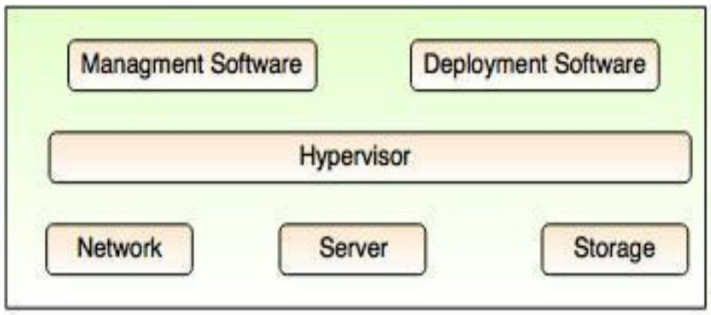


Fig. - Cloud Infrastructural Components

a) Hypervisor

- Hypervisor is a firmware or low-level program. It acts as a Virtual Machine Manager.
- It enables to share a physical instance of cloud resources between several customers.

b) Management Software

- Management software assists to maintain and configure the infrastructure.

c) Deployment Software

- Deployment software assists to deploy and integrate the application on the cloud.

d) Network

- Network is the key component of the cloud infrastructure.
- It enables to connect cloud services over the Internet.
- The customer can customize the network route and protocol i.e possible to deliver network as a utility over the Internet.

e) Server

- The server assists to compute the resource sharing and offers other services like resource allocation and de-allocation, monitoring the resources, provides the security etc.

6) Storage

- Cloud keeps many copies of storage. Using these copies of resources, it extracts another resource if any one of the resources fails.

Intranets and the Cloud: Intranets are customarily used within an organization and are not accessible publicly. That is, a web server is maintained in-house and company information is maintained on it that others within the organization can access. However, now intranets are being maintained on the cloud.

- To access the company's private, in-house information, users have to log on to the intranet by going to a secure public web site.

- There are two main components in client/server computing: servers and thin or light clients.
- The servers house the applications your organization needs to run, and the thin

Clients—who do not have hard drives—display the results.

Hypervisor Applications

- Applications like VMware or Microsoft's Hyper-V allow you to virtualize your servers so

that multiple virtual servers can run on one physical server.

- These sorts of solutions provide the tools to supply a virtualized set of hardware to the guest operating system. They also make it possible to install different operating systems on the same machine. For example, you may need Windows Vista to run one application, while another application requires Linux. It's easy to set up the server to run both operating systems.
- Thin clients use an application program to communicate with an application server.

Most of the processing is done down on the server, and sent back to the client.

There is some debate about where to draw the line when talking about thin clients.

Some thin clients require an application program or a web browser to communicate with the server. However, others require no add-on applications at all. This is sort of a discussion of semantics, because the real issue is whether the work is being done on the server and transmitted back to the thin client.

1.8. Cloud computing techniques

Some traditional computing techniques that have helped enterprises achieve additional computing and storage capabilities, while meeting customer demands using shared physical resources, are:

- **Cluster computing** connects different computers in a single location via LAN to work as a single computer. Improves the combined performance of the organization which owns it

- **Grid computing** enables collaboration between enterprises to carry out distributed computing jobs using interconnected computers spread across multiple locations running independently
- **Utility computing** provides web services such as computing, storage space, and applications to users at a low cost through the virtualization of several backend servers. Utility computing has laid the foundation for today's cloud computing
- **Distributed computing** landscape connects ubiquitous networks and connected devices enabling peer-to-peer computing. Examples of such cloud infrastructure are ATMs, and intranets/ workgroups

Grid Computing Vs Cloud Computing

When we switch on the fan or any electric device, we are less concern about the power supply from where it comes and how it is generated. The power supply or electricity that we receives at our home travels through a chain of network, which includes power stations, transformers, power lines and transmission stations. These components together make a 'Power Grid'. Likewise, 'Grid Computing' is an infrastructure that links computing resources such as PCs, servers,

workstations and storage elements and provides the mechanism required to access them.

Grid Computing is a middle ware to co-ordinate disparate IT resources across a network, allowing them to function as whole. It is more often used in scientific research and in universities for educational purpose. For example, a group of architect students working on a different project requires a specific designing tool and a software for designing purpose but only couple of them got access to this designing tool, the problem is how they can make this tool available to rest of the students. To make available for other students they will put this designing tool on campus network, now the grid will connect all these computers in campus network and allow student to use designing tool required for their project from anywhere. Cloud computing and Grid computing is often confused, though there functions are almost similar there approach for their functionality is different. Let see how they operate-

Cloud Computing	**Grid Computing**
• Cloud computing works more as a service provider for utilizing computer resource	• Grid computing uses the available resource and interconnected computer systems to accomplish a common goal
• Cloud computing is a centralized model	• Grid computing is a decentralized model, where the computation could occur over many administrative model

• Cloud is a collection of computers usually owned by a single party. •	• A grid is a collection of computers which is owned by a multiple parties in multiple locations and connected together so that users can share the combined power of resources
• Cloud offers more services all most all the services like web hosting, DB (Data Base) support and much more	• Grid provides limited services
• Cloud computing is typically provided within a single organization (eg : Amazon)	• Grid computing federates the resources located within different organization.

Utility Computing Vs Cloud Computing

In our previous conversation in "Grid Computing" we have seen how electricity is supplied to our house, also we do know that to keep electricity supply we have to pay the bill. Utility Computing is just like that, we use electricity at home as per our requirement and pay the bill accordingly likewise you will use the services for the computing and pay as per the use this is known as 'Utility computing'. Utility computing is a good source for small scale usage, it can be done in any server environment and requires Cloud Computing.

Utility computing is the process of providing service through an on-demand, pay per use billing method. The customer or client has access to a virtually unlimited supply of computing solutions over a virtual private network or over the internet, which can be sourced and used whenever it's required. Based on the concept of utility computing , grid computing, cloud computing and managed IT services are based.

Through utility computing small businesses with limited budget can easily use software like CRM (Customer Relationship Management) without investing heavily on infrastructure to maintain their clientele base.

Utility Computing	**Cloud Computing**
• Utility computing refers to the ability to charge the offered services, and charge customers for exact usage	• Cloud Computing also works like utility computing, you pay only for what you use but Cloud Computing might be cheaper, as such, Cloud based app can

	be up and running in days or weeks.
• Utility computing users want to be in control of the geographical location of the infrastructure	• In cloud computing, provider is in complete control of cloud computing services and infrastructure
• Utility computing is more favorable when performance and selection infrastructure is critical	• Cloud computing is great and easy to use when the selection infrastructure and performance is not critical
• Utility computing is a good choice for less resource demanding	• Cloud computing is a good choice for high resource demanding
• Utility computing refers to a business model	• Cloud computing refers to the underlying IT architecture

1.9 Security concerns for Cloud Computing

While using cloud computing, the major issue that concerns the users is about its security.
One concern is that cloud providers themselves may have access to customer's. unencrypted data- whether it's on disk, in memory or transmitted over the network. Some countries government may decide to search through data without necessarily notifying the data owner, depending on where the data resides, which is not appreciated and is considered as a privacy breach (Example Prism Program by USA).
To provide security for systems, networks and data cloud computing service providers have joined hands with TCG (Trusted Computing Group) which is non-profit organization which regularly releases a set of specifications to secure hardware, create self-encrypting drives and improve network security. It protects the data from root kits and malware.
As computing has expanded to different devices like hard disk drives and mobile phones, TCG has extended the security measures to include these devices. It provides ability to create a unified data protection policy across all clouds.

Some of the trusted cloud services are Amazon, Box.net, Gmail and many others.

1.10 Privacy Concern & Cloud Computing

Privacy presents a strong barrier for users to adapt into Cloud Computing systems

There are certain measures which can improve privacy in cloud computing.

1. The administrative staff of the cloud computing service could theoretically monitor the data moving in memory before it is stored in disk. To keep the confidentiality of a data, administrative and legal controls should prevent this from happening.
2. The other way for increasing the privacy is to keep the data encrypted at the cloud storage site, preventing unauthorized access through the internet; even cloud vendor can't access the data either.

ii) Full Virtualization

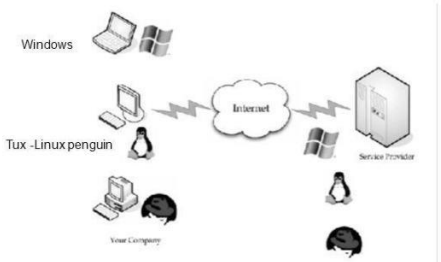
- Full virtualization is a technique in which a complete installation of one machine is run on another. The result is a system in which all software running on the server is within a virtual machine.
- In a fully virtualized deployment, the software running on the server is displayed on the clients.
- Virtualization is relevant to cloud computing because it is one of the ways in which you will access services on the cloud. That is, the remote datacenter may be delivering your services in a fully virtualized format.
- In order for full virtualization to be possible, it was necessary for specific hardware

combinations to be used. It wasn't until 2005 that the introduction of the AMD-Virtualization(AMD-V) and Intel Virtualization Technology (IVT) extensions made it easier to go fully virtualized.

Full virtualization has been successful for several purposes:
i) Sharing a computer system among multiple users
ii) Isolating users from each other and from the control program
iii) Emulating hardware on another machine

iii) Para virtualization

Para virtualization allows multiple operating systems to run on a single hardware device at the same time by more efficiently using system resources, like processors and memory. In full virtualization, the entire system is emulated (BIOS, drive, and so on), but in para virtualization, its management module operates with an operating system that has been adjusted to work in a virtual machine. Para virtualization typically runs better than the full virtualization model, simply because in a fully virtualized deployment, all elements must be emulated.



In a paravirtualized deployment, many different operating systems can run simultaneously.

Virtualization Type	Guest Instances	Virtualization Overhead	System Processing Needs	Total
Full Virtualization	8	10% (80% total)	10% (80% total)	100%
Paravirtualization	8	2% (16% total)	10% (80% total)	96%

Correction: 80%

TABLE 1-1 Processor Power Used in Full Virtualization and Paravirtualization

CHALLENGES AND RISKS

Despite the initial success and popularity of the cloud computing paradigm and the extensive availability of providers and tools, a significant number of challenges and risks are inherent to this new model of computing. Providers, developers, and end users must consider these challenges and risks to take good advantage of cloud computing. Issues to be faced include user privacy, data security, data lock- in, availability of service, disaster recovery, performance, scalability, energy- efficiency, and programmability. Security, Privacy, and Trust: Security and privacy affect the entire cloud computing stack, since there is a massive use of third- party services and infrastructures that are used to host important data or to perform critical operations. In this scenario, the trust toward providers is fundament al to ensure the desired level of privacy for applications hosted in the cloud. | 62 Legal and regulatory issues also need attention. When data are moved into the Cloud, providers may choose to locate them anywhere on the planet. The physical location of data centers determines the set of laws that can be applied to the management of data. For example, specific cryptography techniques could not be used because they are not allowed in some countries. Similarly, country laws can impose that sensitive data, such as patient health records, are to be stored within national borders. Data Lock- In and Standardization: A major concern of cloud computing users is about having their data locked- in by a certain provider. Users may want to move data and applications out from a provider that does not meet their requirements. However, in their current form, cloud computing infrastructures and platforms do not employ standard methods of storing user data and applications. Consequently, they do not interoperate and user data are not portable. The answer

to this concern is standardization. In this direction, there are efforts to create open standard s for cloud computing. The Cloud Computing Interopera bility Forum (CCIF) was formed by organizations such as Intel, Sun, and Cisco in order to "enable a global cloud computing ecosystem whereby organizations are able to seamlessly work together for the purposes for wider industry adoption of cloud computing technology." The development of the Unified Cloud Interface (UCI) by CCIF aims at creating a standard programmatic point of access to an entire cloud infrastructure. In the hardware virtualization sphere, the Open Virtual Format (OVF) aims at facilitating packing and distribution of software to be run on VMs so that virtual appliances can be made portable—that is, seamlessly run on hypervisor of different vendors. Availability, Fault- Tolerance, and Disaster Recovery: It is expected that users will have certain expectations about the service level to be provided once their applications are moved to the cloud. These expectations include availability of the service, its overall performance, and what measures are to be taken when something goes wrong in the system or its component s. In summary, users seek for a warranty before they can comfortably move their business to the cloud. SLAs, which include QoS requirements, must be ideally set up between customers and cloud computing providers to act as warranty. An SLA specifies the details of the service to be provided, including availability and performance guarantees. Additionally, metrics must be agreed upon by all parties, and penalties for violating the expectations must also be approved. Resource Management and Energy- Efficiency: One important challenge faced by providers of cloud computing services is the efficient manage m e n t of virtualized resource pools. Physical resources such as CPU cores, disk space, and network bandwidth must be sliced and shared among virtual machines running potentially heterogeneous workloads. The multidimensional nature of virtual machines complicates the activity of finding a good mapping of VMs onto available physical hosts while maximizing user | 63 utility. Dimensions to be considered include: number of CPUs, amount of memory, size of virtual disks, and network bandwidth. Dynamic VM mapping policies may leverage the ability to suspend, migrate, and resume VMs as an easy way of preempting low- priority allocations in favor of higher- priority ones. Migration of VMs also brings additional challenges such as detecting when to initiate a migration, which VM to migrate, and where to migrate. In addition, policies may take advantage of live migration of virtual machines to relocate data center load without significantly disrupting running services. In this case, an additional concern is the tradeoff between the negative impact of a live migration on the performance and stability of a service and the benefits to be achieved with that migration. Another challenge concerns the outstanding amount of data to be managed in various VM manage m e n t activities. Such data amount is a result of particular abilities of virtual machines, including the ability of traveling through space (i.e., migration) and time (i.e., check pointing and rewinding), operations that may be required in load balancing, backup, and recovery scenarios. In addition, dynamic provisioning of new VMs and replicating existing VMs require efficient mechanisms to make VM block storage devices (e.g., image files) quickly available at selected hosts. Data centers consume r large amounts of electricity. According to a data published by HP[4], 100 server racks can consume 1.3MWof power and another 1.3 MW are required by the

cooling system, thus costing USD 2.6 million per year. Besides the monetary cost, data centers significantly impact the environment in terms of CO2 emissions from the cooling systems

Issues in cloud:

The Eucalyptus : framework was one of the first open- source projects to focus on building IaaS clouds. It has been developed with the intent of providing an open- source implement a tion nearly identical in functionality to Amazon Web Services APIs. Eucalyptus provides the following features: Linux- based controller with administration Web portal; EC2- compatible (SOAP, Query) and S3- compatible (SOAP, REST) CLI and Web portal interfaces; Xen, KVM, and VMWare backends; Amazon EBS- compatible virtual storage devices; interface to the Amazon EC2 public cloud; virtual networks.

Nimbus3: The Nimbus toolkit is built on top of the Globus framework. Nimbus provides most features in common with other open- source VI managers, such as an EC2- compatible front- end API, support to Xen, and a backend interface to Amazon EC2. However, it distinguishes from others by providing a Globus Web Services Resource Framework (WSRF) interface. It also provides a backend service, named Pilot, which spawns VMs on clusters manage d by a local resource manager (LRM) such as PBS and SGE.

Open Nebula: Open Nebula is one of the most feature- rich open- source VI managers. It was initially conceived to manage local virtual infrastructure, but has also included remote interfaces that make it viable to build public clouds. Altogether r, four programming APIs are available: XML-RPC and libvirt for local interaction; a subset of EC2 (Query) APIs and the Open Nebula Cloud API (OCA) for public access. Open Nebula provides the following features: Linux- based controller; CLI, XML-RPC, EC2- compatible Query and OCA interfaces; Xen, KVM, and VMware backend; interface to public clouds (Amazon EC2, Elastic Hosts); virtual networks; dynamic resource allocation; advance reservation of capacity.

CASE STUDY

The Eucalyptus :

framework was one of the first open- source projects to focus on building IaaS clouds. It has been developed with the intent of providing an open- source implement a tion nearly identical in functionality to Amazon Web Services APIs. Eucalyptus provides the following features: Linux-based controller with administration Web portal; EC2- compatible (SOAP, Query) and S3-compatible (SOAP, REST) CLI and Web portal interfaces; Xen, KVM, and VMWare backends; Amazon EBS- compatible virtual storage devices; interface to the Amazon EC2 public cloud; virtual networks.

Nimbus3 : The Nimbus toolkit is built on top of the Globus framework. Nimbus provides most features in common with other open- source VI manage rs, such as an EC2- compatible front-end API, support to Xen, and a backend interface to Amazon EC2. However, it distinguishes from others by providing a Globus Web Services Resource Framework (WSRF) interface. It also provides a backend service, named Pilot, which spawns VMs on clusters manage d by a local resource manager (LRM) such as PBS and SGE.

Open Nebula:

Open Nebula is one of the most feature- rich open- source VI managers. It was initially conceived to manage local virtual infrastructure, but has also included remote interfaces that make it viable to build public clouds. Altogether, four programming APIs are available: XML-RPC and libvirt for local interaction; a subset of EC2 (Query) APIs and the OpenNebula Cloud API (OCA) for public access. OpenNebula provides the following features: Linux- based controller; CLI, XML-RPC, EC2- compatible Query and OCA interfaces; Xen, KVM, and VMware backend; interface to public clouds (Amazon EC2, ElasticHosts); virtual networks; dynamic resource allocation; advance reservation of capacity.

i) Case-Study of Cloud Computing- Royal Mail

- **Subject of Case-Study:** Using Cloud Computing for effective communication among staff.
- **Reason for using Cloud Computing:** Reducing the cost made after communication for 28,000 employees and to provide advance features and interface of e-mail services to their employees.

Royal mail group, a postal service in U.K, is the only government organization in U.K that serves over 24 million customers through its 12000 post offices and 3000 separate processing sites. Its logistics systems and parcel-force worldwide handles around 404 million parcel a year. And to do this they need an effective communicative medium. They have recognized the advantage of Cloud Computing and implemented it to their system. It has shown an outstanding performance in inter-communication.

Before moving on to Cloud system, the organization was struggling with the out-of-date software, and due to which the operational efficiency was getting compromised. As soon as the organization switched on to Cloud System, 28000 employees were supplied with their new collaboration suite, giving them access to tools such as instant messaging and presence awareness. The employees got more storage place than on local server. The employees became much more productive.

Looking to the success of Cloud Computing in e-mail services and communication .The second strategic move of Royal Mail Group, was to migrating from physical servers to virtual servers, up to 400 servers to create a private cloud based on Microsoft hyper V. This would give a fresh look and additional space to their employees desktop and also provides latest modern exchange environment.

The hyper V project by RMG's (Royal Mail Group) is estimated to save around 1.8 million pound for them in future and will increase the efficiency of the organization's internal IT system.

Case study -2

XYZ is a startup IT organization that develops and sells s/w the org gets a new website development project that needs a web server, application server and a database server. The org has hired 30 employees for this web development project.

Constraints :

Acquiring renting space for new servers
Buying new high end servers
Hiring new IT staff for infrastructure management
Buying licensed OS and other s/w required for development

Solution :Public cloud IaaS

Team leader :

1. Creates an ac
2. Choose an VM image from image repository or create a new image
3. Specify no.of VM's
4. Choose VM type
5. Set necessary configurations for VM
6. After VM launched ,provide IP address of VM to prog team
7. Access VM and start development

Case study -2

Case study -3

XYZ firm gets more revenue ,grows and hence buys some IT infrastructuire.However it continues to use public IaaS cloud for its development work

Now the firm gets a new project that involves sensitive data that restricts the firm to use a public cloud .hence this org is in need of setting up the required infrastructure in its own premise.

Constraints:

Infrastructure cost

Infrastructure optimization

Power consumption

Data center management

Additional expenditure on infrastructure operation with lesser productivity

Solution : **Private IaaS cloud**

Moving to private cloud is :

Moving to private cloud

IT managed → self-service

Physical → virtual

Manual management → automated management

Dedicated → shared

Explanation:

1.Setup cloud infrastructure

2. Setup self-service portal or dashboard

3. Test the cloud environment through self-service

4. Get VM's

5. Use VM's to develop and test applications

6. Manage cloud environment

Cloud Computing Service Provider Companies in 2019

1) Amazon Web Services

AWS is Amazon's cloud web hosting platform which offers fast, flexible, reliable and cost-effective solutions. It offers a service in the form of building block which can be used to create and deploy any kind of application in the cloud. It is the most popular as it was the first to enter the cloud computing space.

Features:

- Easy sign-up process
- Fast Deployments
- Allows easy management of add or remove capacity
- Access to effectively limitless capacity
- Centralized Billing and management
- Offers Hybrid Capabilities and per hour billing

Download link:https://aws.amazon.com/

2) Microsoft Azure

Azure is a cloud computing platform which is launched by Microsoft in February 2010. This open source and flexible cloud platform which helps in development, data storage, service management & hosting solutions.

Features:

- Windows Azure offers the most effective solution for your data needs
- Provides scalability, flexibility, and cost-effectiveness
- Offers consistency across clouds with familiar tools and resources
- Allow you to scale your IT resources up and down according to your business needs

Download link:https://azure.microsoft.com/en-in/

3) Google Cloud Platform

Google Cloud is a set of solution and products which includes GCP & G suite. It helps you to solve all kind of business challenges with ease.

Features:

- Allows you to scale with open, flexible technology
- Solve issues with accessible AI & data analytics
- Eliminate the need for installing costly servers
- Allows you to transform your business with a full suite of cloud-based services

Download link:https://cloud.google.com/

---

4) VMware

VMware is a comprehensive cloud management platform. It helps you to manage a hybrid environment running anything from traditional to container workloads. The tools also allow you to maximize the profits of your organization.

Features:

- Enterprise-ready Hybrid Cloud Management Platform
- Offers Private & Public Clouds
- Comprehensive reporting and analytics which improve the capacity of forecasting & planning
- Offers additional integrations with 3rd parties and custom applications, and tools.
- Provides flexible, Agile services

Download link:https://www.vmware.com/in/cloud-services/infrastructure.html

Oracle Cloud

Oracle Cloud offers innovative and integrated cloud services. It helps you to build, deploy, and manage workloads in the cloud or on premises. Oracle Cloud also helps companies to transform their business and reduce complexity.

Features:

- Oracle offers more options for where and how you make your journey to the cloud
- Oracle helps you realize the importance of modern technologies including Artificial intelligence, chatbots, machine learning, and more
- Offers Next-generation mission-critical data management in the cloud
- Oracle provides better visibility to unsanctioned apps and protects against sophisticated cyber attacks

Download link:https://www.oracle.com/cloud/

5) IBM Cloud

IBM cloud is a full stack cloud platform which spans public, private and hybrid environments. It is built with a robust suite of advanced and AI tools.

Features:

- IBM cloud offers infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS)
- IBM Cloud is used to build pioneering which helps you to gain value for your businesses
- It offers high performing cloud communications and services into your IT environment

Download link:https://www.ibm.com/cloud/

Tips for selecting a Cloud Service Provider

There "best" Cloud Service cannot be defined. You need to a chose a cloud service "best" for your project. Following checklist will help:

- Is your desired region supported?
- Cost for the service and your budget

- For an outsourcing company, Customer/Client Preference of service provider needs to be factored in
- Cost involved in training employees on the Cloud Service Platform
- Customer support
- The provider should have a successful track record of stability/uptime/reliability
- Reviews of the company

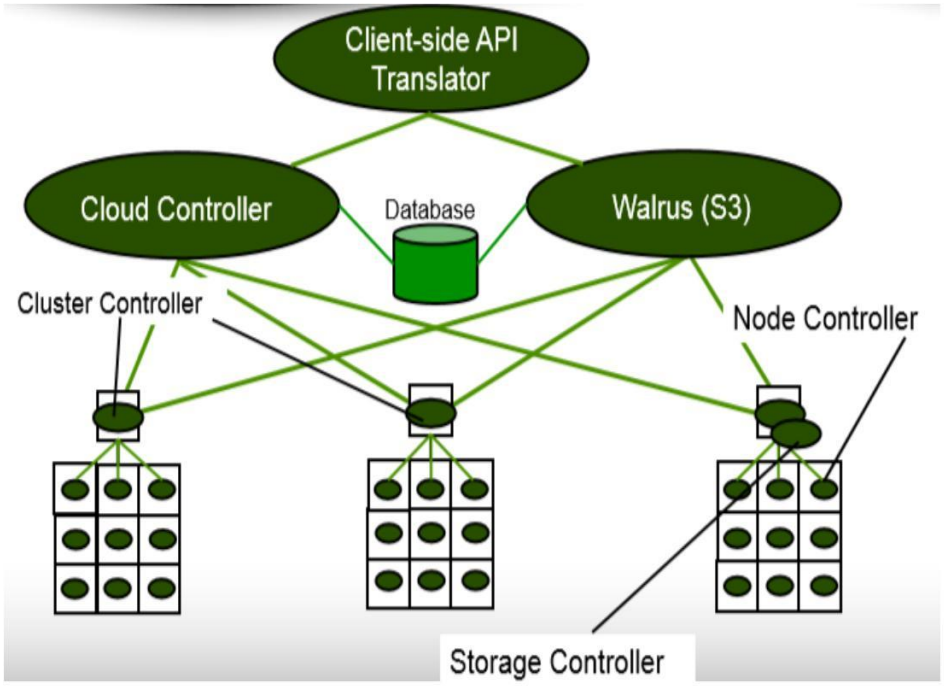
Here is a list of Top 21 Cloud Service Providers for Quick Reference

- Amazon Web Services Alibaba Cloud
- Microsoft Azure Google Cloud Platform
- VMware Rackspace
- Salesforce Oracle Cloud
- Verizon Cloud Navisite
- IBM Cloud OpenNebula
- Pivotal DigtialOceanCloudSigma Dell Cloud
- LiquidWeb LimeStone

MassiveGridQuadranet Kamatera

Eucalyptus

- Eucalyptus is an acronym for **Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems**.
- Eucalyptus is a paid and open-source computer software for building Amazon Web Services (AWS)-compatible private and hybrid cloud computing environments, originally developed by the company Eucalyptus Systems.
- Eucalyptus enables pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change



Eucalyptus has six components:

1.The **Cloud Controller** (CLC) is a Java program that offers EC2-compatible interfaces, as well as a web interface to the outside world.

- In addition to handling incoming requests, the CLC acts as the administrative interface for cloud management and performs high-level resource scheduling and system accounting.
- The CLC accepts user API requests from command-line interfaces like euca2ools or GUI-based tools like the Eucalyptus User Console and manages the underlying compute, storage, and network resources.
- Only one CLC can exist per cloud and it handles authentication, accounting, reporting, and quota management.

2.Walrus, also written in Java, is the Eucalyptus equivalent to AWS Simple Storage Service (S3).

- Walrus offers persistent storage to all of the virtual machines in the Eucalyptus cloud and can be used as a simple HTTP put/get storage as a service solution.
- There are no data type restrictions for Walrus, and it can contain images (i.e., the building blocks used to launch virtual machines), volume snapshots (i.e., point-in-time copies), and application data. Only one Walrus can exist per cloud.

3.The **Cluster Controller** (CC) is written in C and acts as the front end for a cluster within a Eucalyptus cloud and communicates with the Storage Controller and Node Controller.

- It manages instance (i.e., virtual machines) execution and Service Level Agreements (SLAs) per cluster.

4.The **Storage Controller** (SC) is written in Java and is the Eucalyptus equivalent to AWS EBS. It communicates with the Cluster Controller and Node Controller and manages Eucalyptus block volumes and snapshots to the instances within its specific cluster.

- If an instance requires writing persistent data to memory outside of the cluster, it would need to write to Walrus, which is available to any instance in any cluster.

5.The **Node Controller** (NC) is written in C and hosts the virtual machine instances and manages the virtual network endpoints.

- It downloads and caches images from Walrus as well as creates and caches instances.
- While there is no theoretical limit to the number of Node Controllers per cluster, performance limits do exist.

6.The **VMware Broker** is an optional component that provides an AWS-compatible interface for VMware environments and physically runs on the Cluster Controller.

- The VMware Broker overlays existing ESX/ESXi hosts and transforms Eucalyptus Machine Images (EMIs) to VMware virtual disks.
- The VMware Broker mediates interactions between the Cluster Controller and VMware and can connect directly to either ESX/ESXi hosts or to vCenter Server.

Nimbus

- Nimbus is a set of open source tools that together provide an "Infrastructure-as-a-Service" (IaaS) cloud computing solution.
- Mission is to evolve the infrastructure with emphasis on the needs of science, but many non-scientific use cases are supported as well.

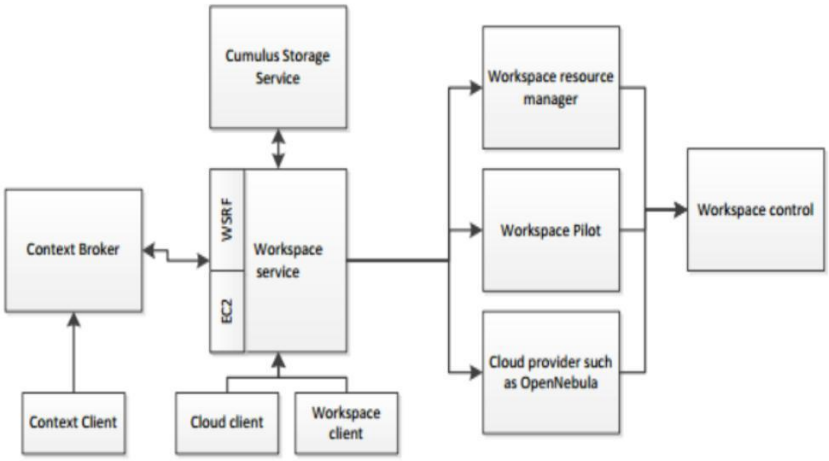
- Nimbus allows a client to lease remote resources by deploying virtual machines (VMs) on those resources and configuring them to represent an environment desired by the user.
- It was formerly known as the "Virtual Workspace Service" (VWS) but the "workspace service" is technically just one the components in the software .
- Nimbus is a toolkit that, once installed on a cluster, provides an infrastructure as a service cloud to its client via WSRF-based or Amazon EC2 WSDL web service APIs.
- Nimbus is free and open-source software, subject to the requirements of the Apache License, version 2.
- Nimbus supports both the hypervisors Xen and KVM and virtual machine schedulers Portable Batch System and Oracle Grid Engine.
- It allows deployment of self-configured virtual clusters via contextualization.
- It is configurable with respect to scheduling, networking leases, and usage accounting.
- Nimbus is comprised of two products:

Nimbus Infrastructure

Nimbus Platform

- **Nimbus Infrastructure** is an open source EC2/S3-compatible Infrastructure-as-a-Service implementation specifically targeting features of interest to the scientific community such as support for proxy credentials, batch schedulers, best-effort allocations and others.
- **Nimbus Platform** is an integrated set of tools, operating in a multi-cloud environment, that deliver the power and versatility of infrastructure clouds to scientific users. Nimbus Platform allows you to reliably deploy, scale, and manage cloud resources.

System Architecture & Design



- The design of nimbus which consists of a number of components based on the web service technology.

1. **Workspace service**

   Allows clients to manage and administer VMs by providing to two interfaces:

- A)One interface is based on the web service is resource framework (WSRF)
- B)The other is based on EC2 WSDL

2. **Workspace resource manager**

   implements VM instance creation on a site management.

3. **Workspace pilot**

- Provides virtualization with significant changes to the site configurations.

4. **workspace control**

- Implements VM instance management such as start, stop and pause VM. It also provides image management and set up networks and provides IP assignment.

5.**context Broker**

- Allows clients coordinate large virtual cluster launches automatically and repeatedly.

6. **Workspace client**

- A complex client that provides full access to the workspace service functionality.

7. **Cloud client**

- A simpler client providing access to selected functionalities in the workspace service.

8. **Storage service**

- cumulus is a web service providing users with storage capabilities to store images and works in conjunction with GridFTP.

Open Nebula

- Open Nebula- is an open source cloud computing platform for managing heterogeneous distributed data centre infrastructures.
- Manages a data centre's virtual infrastructure to build private,public and hybrid implementations of Iaas.
- Two primary uses of open nebula platform are:

data center virtualization

- Many of our users use OpenNebula to manage data center virtualization, consolidate servers, and integrate existing IT assets for computing, storage, and networking.
- In this deployment model, OpenNebula directly integrates with hypervisors (like KVM, Xen or VMware ESX) and has complete control over virtual and physical resources, providing advanced features for capacity management, resource optimization, high availability and business continuity.
- Some of these users also enjoy OpenNebula's cloud management and provisioning features when they additional want to federate data centers, implement cloud bursting, or offer self-service portals for users.

Cloud infrastructure solutions

- We also have users that use OpenNebula to provide a multitenant, cloud-like provisioning layer on top of an existing infrastructure management solution (like VMware vCenter).
- These users are looking for provisioning, elasticity and multi-tenancy cloud features like virtual data centers provisioning, datacenter federation or hybrid cloud computing to

connect in-house infrastructures with public clouds, while the infrastructure is managed by already familiar tools for infrastructure management and operation

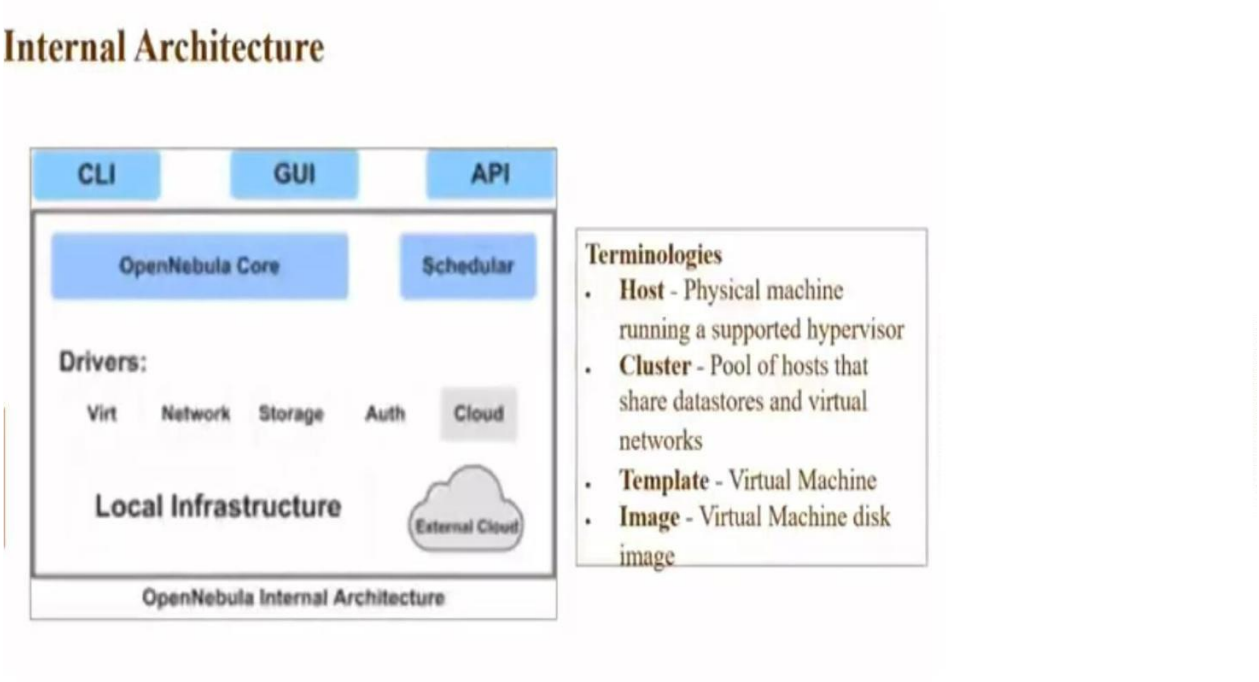
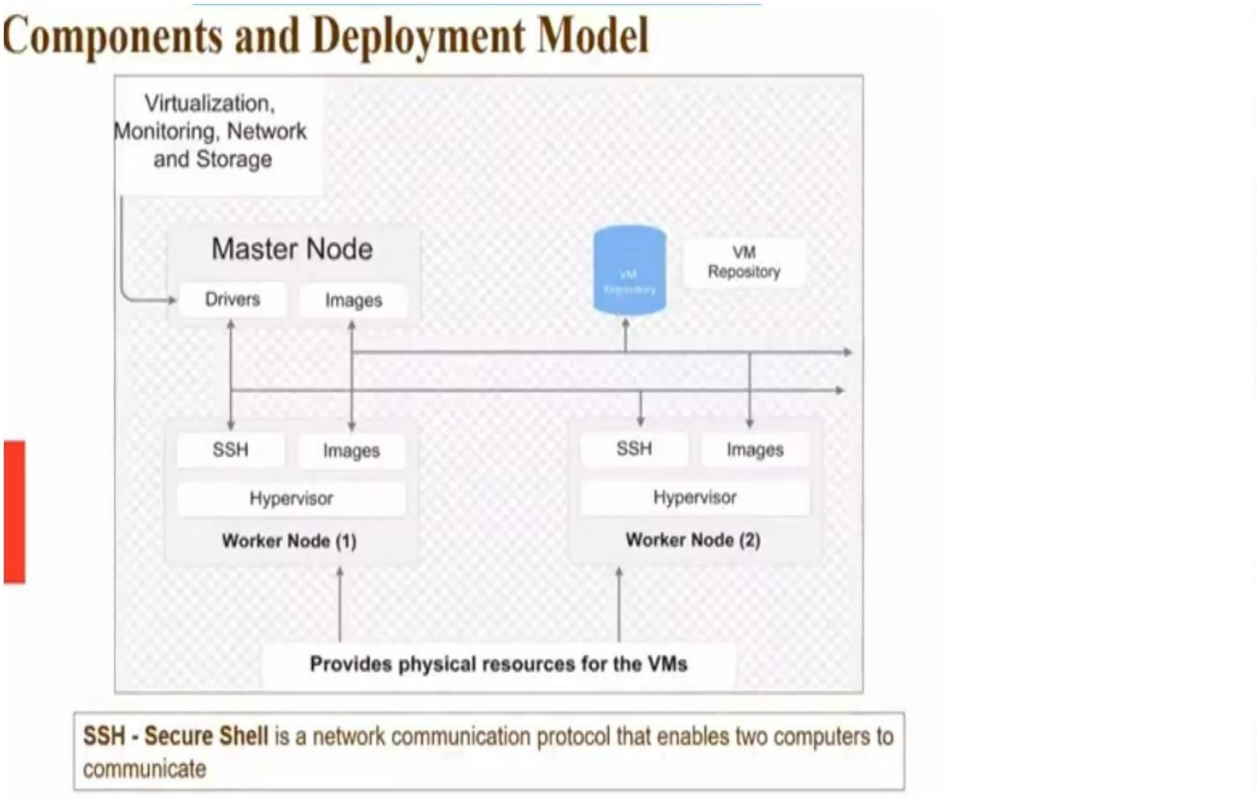


Image Repository: Any storage medium for the VM images (usually a high performing SAN).

Cluster Storage : OpenNebula supports multiple back-ends (e.g. LVM for fast cloning)

VM Directory: The home of the VM in the cluster node

- Stores checkpoints, description files and VM disks
- Actual operations over the VM directory depends on the storage medium
- Should be shared for live-migrations
- You can go on without a shared FS and use the SSH back-eλ



Master node: A single gateway or front-end machine, sometimes also called the master node, is responsible for queuing, scheduling and submitting jobs to the machines in the cluster. It runs several other OpenNebula services mentioned below:

- Provides an interface to the user to submit virtual machines and monitor their status.
- Manages and monitors all virtual machines running on different nodes in the cluster.
- It hosts the virtual machine repository and also runs a transfer service to manage the transfer of virtual machine images to the concerned worker nodes.
- Provides an easy-to-use mechanism to set up virtual networks in the cloud.
- Finally, the front-end allows you to add new machines to your cluster.

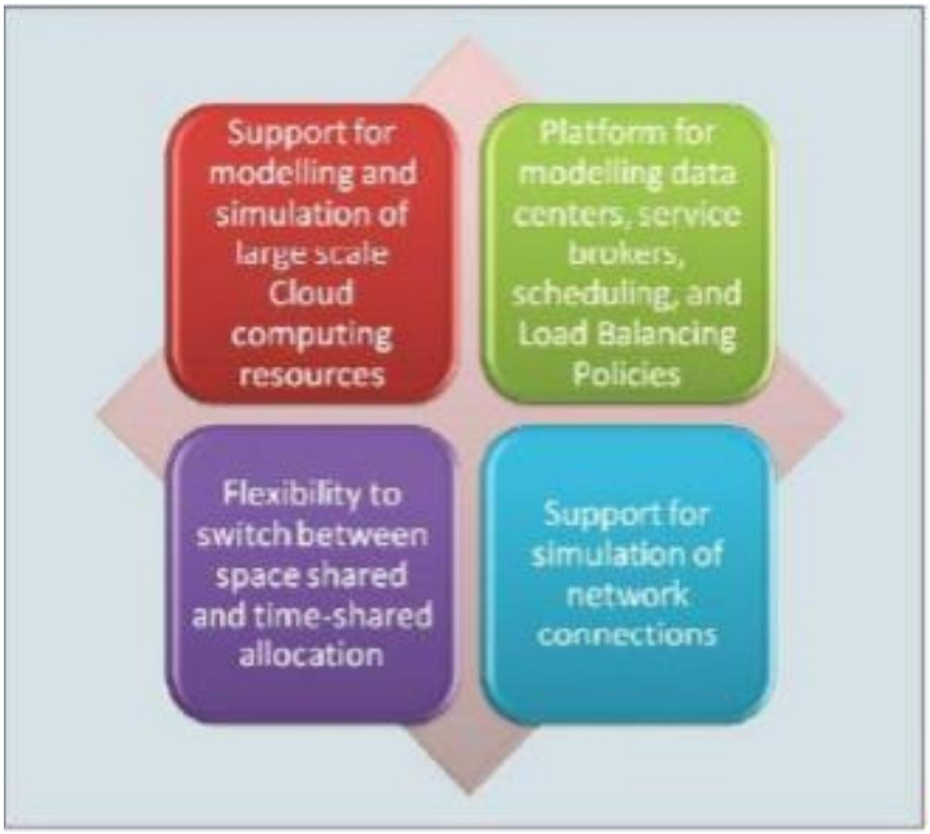
Worker node: The other machines in the cluster, known as 'worker nodes', provide raw computing power for processing the jobs submitted to the cluster. The worker nodes in an

OpenNebula cluster are machines that deploy a virtualisation hypervisor, such as VMware, Xen or KVM.

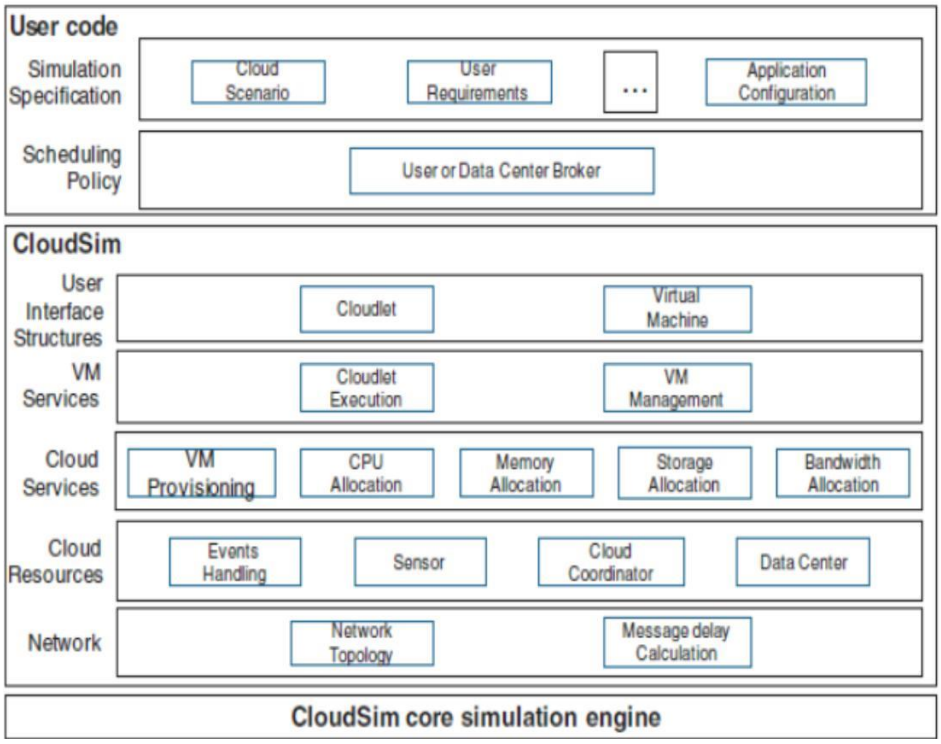
CloudSim

- CloudSim is a framework for modeling and simulation of cloud computing infrastructures and services.
- Originally built primarily at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia, CloudSim has become one of the most popular open source cloud simulators in the research and academia.
- CloudSim is completely written in Java.
- By using CloudSim, developers can focus on specific systems design issues that they want to investigate, without getting concerned about details related to cloud-based infrastructures and services.
- CloudSim is a simulation tool that allows cloud developers to test the performance of their provisioning policies in a repeatable and controllable environment, free of cost.
- It helps tune the bottlenecks before real-world deployment.
- It is a simulator; hence, it doesn't run any actual software.
- It can be defined as 'running a model of an environment in a model of hardware', where technology-specific details are abstracted.
- CloudSim is a library for the simulation of cloud scenarios.
- It provides essential classes for describing data centres, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning.
- It can be used as a building block for a simulated cloud environment and can add new policies for scheduling, load balancing and new scenarios.
- It is flexible enough to be used as a library that allows you to add a desired scenario by writing a Java program.

Features of Cloudsim



Architecture of CloudSim



- The **user code :**layer exposes basic entities such as the number of machines, their specifications, etc, as well as applications, VMs, number of users, application types and scheduling policies.
- The **User Code layer** is a custom layer where the user writes their own code to redefine the characteristics of the stimulating environment as per their new research findings.
- **Network Layer:** This layer of CloudSim has responsibility to make communication possible between different layers. This layer also identifies how resources in cloud environment are places and managed.
- **Cloud Resources:** This layer includes different main resources like datacenters, cloud coordinator (ensures that different resources of the cloud can work in a collaborative way) in the cloud environment
- Cloud Services: This layer includes different service provided to the user of cloud services. The various services of clouds include Information as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)

- User Interface: This layer provides the interaction between user and the simulator.
- The **CloudSim Core simulation engine** provides support for modeling and simulation of virtualized Cloud-based data center environments including queuing and processing of events, creation of cloud system entities (like data center, host, virtual machines, brokers, services, etc.) communication between components and management of the simulation clock.

1.Cloud computing services

1.1 Infrastructure as a service - IaaS

AWS supports everything you need to build and run Windows applications including Active Directory, .NET, System Center, Microsoft SQL Server, Visual Studio, and the first and only fully managed native-Windows file system available in the cloud with FSx for Windows File Server.

The AWS advantage for Windows over the next largest cloud provider

- 2x More Windows Server instances
- 2x more regions with multiple availability zones
- 7x fewer downtime hours in 2018*
- 2x higher performance for SQL Server on Windows
- 5x more services offering encryption

AWS offers the best cloud for Windows, and it is the right cloud platform for running Windows-based applications

Windows on Amazon EC2 enables you to increase or decrease capacity within minutes

i. Broader and Deeper Functionality
ii. Greater Reliability
iii. More Security Capabilities
iv. Faster Performance
v. Lower Costs
vi. More Migration Experience

Popular AWS services for Windows workloads

i. SQL Server on Amazon EC2
ii. Amazon Relational Database Service
iii. Amazon FSx for Window File Server
iv. AWS Directory Service
v. AWS License Manager

Service-level agreement (SLA)

A service-level agreement (SLA) is a contract between a service provider and its internal or external customers that documents what services the provider will furnish and defines the service standards the provider is obligated to meet.

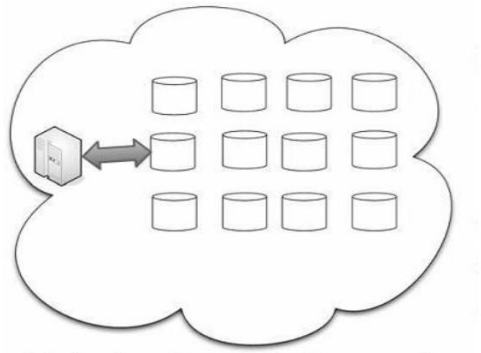
CLOUD STORAGE

3.1.1 Overview

The Basics

Cloud storage is nothing but storing our data with a cloud service provider rather than on a local system, as with other cloud services, we can access the data stored on the cloud via an Internet link. Cloud storage has a number of advantages over traditional data storage. If we store our data on a cloud, we can get at it from any location that has Internet access.

At the most rudimentary level, a cloud storage system just needs one data server connected to the Internet. A subscriber copies files to the server over the Internet, which then records the data. When a client wants to retrieve the data, he or she accesses the data server with a web-based interface, and the server then either sends the files back to the client or allows the client to access and manipulate the data itself.



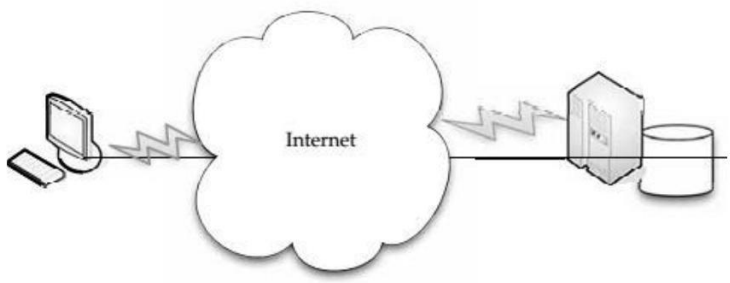
A cloud service provider can simply add more commodity hard drives to increase the organization's capacity.

Cloud storage systems utilize dozens or hundreds of data servers. Because servers require maintenance or repair, it is necessary to store the saved data on multiple machines, providing redundancy. Without that redundancy, cloud storage systems couldn't assure clients that they could access their information at any given time. Most systems store the same data on servers using different power supplies. That way, clients can still access their data even if a power

supply fails.

b.Storage as a Service

The term Storage as a Service (another Software as a Service, or SaaS, acronym) means that a third-party provider rents space on their storage to end users who lack the budget or capital budget to pay for it on their own. It is also ideal when technical personnel are not available or have inadequate knowledge to implement and maintain that storage infrastructure. Storage service providers are nothing new, but given the complexity of current backup, replication, and disaster recovery needs, the service has become popular, especially among small and medium-sized businesses. Storage is rented from the provider using a cost-per-gigabyte-stored or cost-per-data-transferred model. The end user doesn't have to pay for infrastructure; they simply pay for how much they transfer and save on the provider's servers.



Clients rent storage capacity from cloud storage vendors.

A customer uses client software to specify the backup set and then transfers data across a WAN. When data loss occurs, the customer can retrieve the lost data from the serviceprovider.

c.Providers

They are hundreds of cloud storage providers on the Web, and more seem to be added each day. Not only are there general-purpose storage providers, but there are some that are very specialized in what they store.

- ✓ **Google Docs** allows users to upload documents, spreadsheets, and presentations to

Google's data servers. Those files can then be edited using a Google application.

- ✓ **Web email** providers like Gmail, Hotmail, and Yahoo! Mail store email messages on their own servers. Users can access their email from computers and other devices connected to the Internet.
- ✓ **Flickr and Picasa** host millions of digital photographs. Users can create their own online photo albums.
- ✓ **YouTube** hosts millions of user-uploaded video files.
- ✓ **Hostmonster and GoDaddy** store files and data for many client web sites.
- ✓ **Facebook and MySpace** are social networking sites and allow members to post pictures and other content. That content is stored on the company's servers.
- ✓ **MediaMax and Strongspace** offer storage space for any kind of digital data.

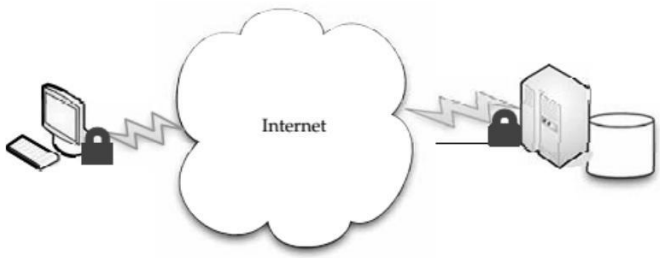
d. Security:

To secure data, most systems use a combination of techniques:

i. **Encryption** A complex algorithm is used to encode information. To decode the encrypted files, a user needs the encryption key. While it's possible to crack encrypted information, it's very difficult and most hackers don't have access to the amount of computer processing power they would need to crack the code.

ii. **Authentication processes** this requires a user to create a name and password.

iii. **Authorization practices** The client lists the people who are authorized to access information stored on the cloud system. Many corporations have multiple levels of authorization. For example, a front-line employee might have limited access to data stored on the cloud and the head of the IT department might have complete and free access to everything.



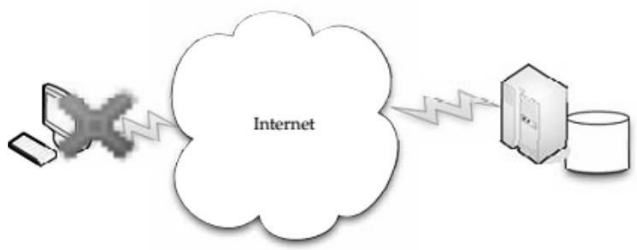
Encryption and authentication are two security measures you can use to keep your data safe on a cloud storage provider.

e. Reliability

Most cloud storage providers try to address the reliability concern through redundancy, but the possibility still exists that the system could crash and leave clients with no way to access their saved data.

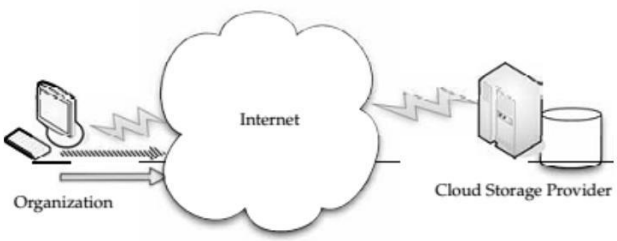
Advantages

- Cloud storage is becoming an increasingly attractive solution for organizations. That's because with cloud storage, data resides on the Web, located across storage systems rather than at a designated corporate hosting site. Cloud storage providers balance server loads and move data among various datacenters, ensuring that information is stored close to where it is used.
- Storing data on the cloud is advantageous, because it allows us to protect our data incase there's a disaster. we may have backup files of our critical information, but if there is a fire or a hurricane wipes out our organization, having the backups stored locally doesn't help.
- Amazon S3 is the best-known storage solution, but other vendors might be better for large enterprises. For instance, those who offer service level agreements and direct access to customer support are critical for a business moving storage to a service provider.



If there is a catastrophe at your organization, having your files backed up at a cloud storage provider means you won't have lost all your data.

- A lot of companies take the "appetizer" approach, testing one or two services to see how well they mesh with their existing IT systems. It's important to make sure the services will provide what we need before we commit too much to the cloud.



Many companies test out a cloud storage vendor with one or two services before committing too much to them. This "appetizer" approach ensures the provider can give you what you want.

3.1.2 Cloud Storage Providers

Amazon and Nirvanix are the current industry top storage providers.

a. Amazon Simple Storage Service (S3)

- The best-known cloud storage service is Amazon's Simple Storage Service (S3), which launched in 2006.
- Amazon S3 is designed to make web-scale computing easier for developers. Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the Web. It gives any developer access to the same highly scalable data storage infrastructure that Amazon uses to run its own

global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Amazon S3 is intentionally built with a minimal feature set that includes the following functionality:

- ✓ Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects that can be stored is unlimited.
- ✓ Each object is stored and retrieved via a unique developer-assigned key.
- ✓ Objects can be made private or public, and rights can be assigned to specific users.
- ✓ Uses standards-based REST and SOAP interfaces designed to work with any Internet- development toolkit.

Design Requirements

Amazon built S3 to fulfill the following design requirements:

- ✓ **Scalable** Amazon S3 can scale in terms of storage, request rate, and users to support an unlimited number of web-scale applications.
- ✓ **Reliable** Store data durably, with 99.99 percent availability. Amazon says it does not allow any downtime.
- ✓ **Fast** Amazon S3 was designed to be fast enough to support high-performance applications. Server-side latency must be insignificant relative to Internet latency. Any performance bottlenecks can be fixed by simply adding nodes to the system.
- ✓ **Inexpensive** Amazon S3 is built from inexpensive commodity hardware components. As a result, frequent node failure is the norm and must not affect the overall system. It must be hardware-agnostic, so that savings can be captured as Amazon continues to drive down infrastructure costs.
- ✓ **Simple** Building highly scalable, reliable, fast, and inexpensive storage is difficult. Doing so in a way that makes it easy to use for any application anywhere is more difficult. Amazon S3 must do both.

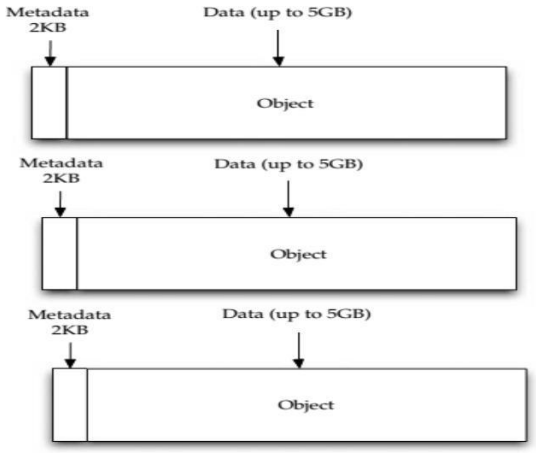
Design Principles

Amazon used the following principles of distributed system design to meet Amazon S3 requirements:

- ✓ **Decentralization** It uses fully decentralized techniques to remove scaling bottlenecks and single points of failure.
- ✓ **Autonomy** The system is designed such that individual components can make decisions based on local information.
- ✓ **Local responsibility** Each individual component is responsible for achieving its consistency; this is never the burden of its peers.
- ✓ **Controlled concurrency** Operations are designed such that no or limited concurrency control is required.
- ✓ **Failure toleration** The system considers the failure of components to be a normal modeof operation and continues operation with no or minimal interruption.
- ✓ **Controlled parallelism** Abstractions used in the system are of such granularity that parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.
- ✓ **Small, well-understood building blocks** Do not try to provide a single service that does everything for everyone, but instead build small components that can be used as building blocks for other services.
- ✓ **Symmetry** Nodes in the system are identical in terms of functionality, and require no or minimal node-specific configuration to function.
- ✓ **Simplicity** The system should be made as simple as possible, but no simpler.

How S3 Works

S3 stores arbitrary objects at up to 5GB in size, and each is accompanied by up to 2KB of metadata. Objects are organized by *buckets*. Each bucket is owned by an AWS account and the buckets are identified by a unique, user-assigned key.



Multiple objects are stored in buckets in Amazon S3.

Buckets and objects are created, listed, and retrieved using either a REST-style or SOAP interface. Objects can also be retrieved using the HTTP GET interface or via BitTorrent.

An access control list restricts who can access the data in each bucket. Bucket names and keys are formulated so that they can be accessed using HTTP. Requests are authorized using an access control list associated with each bucket and object, for instance:

b Nirvanix

Nirvanix uses custom-developed software and file system technologies running on Intel storage servers at six locations on both coasts of the United States. They continue to grow, and expect to add dozens more server locations. SDN Features Nirvanix Storage Delivery Network (SDN) turns a standard 1U server into an infinite capacity network attached storage (NAS) file accessible by popular applications and immediately integrates into an organization's existing archive and backup processes.

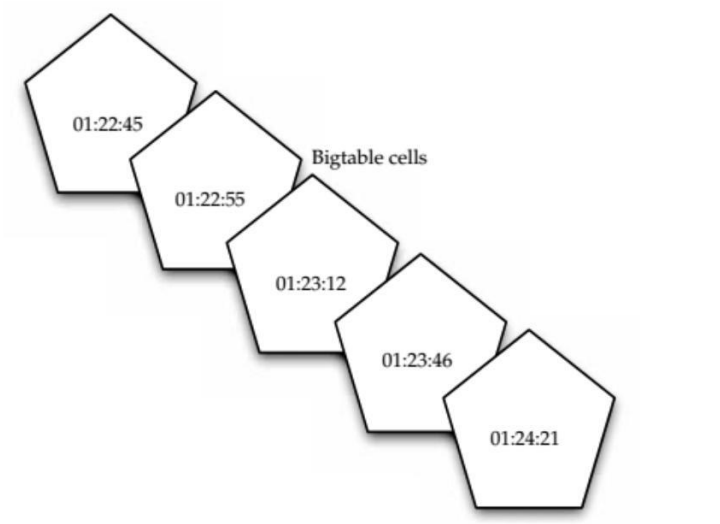
Nirvanix has built a global cluster of storage nodes collectively referred to as the Storage Delivery Network (SDN), powered by the Nirvanix Internet Media File System (IMFS). The SDN intelligently stores, delivers, and processes storage requests in the best network location, providing the best user experience in the marketplace.

Benefits of CloudNAS: The benefits of cloud network attached storage (CloudNAS) include

- ✓ Cost savings of 80–90 percent over managing traditional storage solutions
- ✓ Elimination of large capital expenditures while enabling 100 percent storage utilization
- ✓ Encrypted offsite storage that integrates into existing archive and backup processes
- ✓ Built-in data disaster recovery and automated data replication on up to three geographically dispersed storage nodes for a 100% SLA
- ✓ Immediate availability to data in seconds, versus hours or days on offline tape.

c.Google Bigtable Datastore:

- Datastore In cloud computing, it's important to have a database that is capable of handling numerous users on an on-demand basis. To serve that market, Google introduced its Bigtable. Google started working on it in 2004 and finally went public with it in April 2008. Bigtable was developed with very high speed, flexibility, and extremely high scalability in mind. A Bigtable database can be petabytes in size and span thousands of distributed servers. Bigtable is available to developers as part of the Google App Engine, their cloud computing platform.
- Google describes Bigtable as a fast and extremely scalable DBMS. This allows Bigtable to scale across thousands of commodity servers that can collectively store petabytes of data. Each table in Bigtable is a multidimensional sparse map. That is, the table is made up of rows and columns, and each cell has a timestamp. Multiple versions of a cell can exist, each with a different timestamp. With this stamping, we can select certain versions of a web page, or delete cells that are older than a given date and time.



In Google Bigtable, multiple copies of a cell exist, each with a different timestamp.

d. MobileMe:

- **it** is Apple's solution that delivers push email, push contacts, and push calendars from the MobileMe service in the cloud to native applications on iPhone, iPod touch, Macs, and PCs.
- It provides a suite of ad-free web applications that deliver a desktop like experience through any browser.

e. Live Mesh:

- It is Microsoft's **"software plus services"** platform and experience that enables PCs and other devices to be aware of each other through internet,enabling individuals and organizations to manage ,access and share their files and applications on the web.

components:

- A platform that defines and models a user's digital relationships among devices, data, applications, and people—made available to developers through an open data model and protocols.
- A cloud service providing an implementation of the platform hosted in Microsoft

datacenters.

- Software, a client implementation of the platform that enables local applications to run offline and interact seamlessly with the cloud.
- A platform experience that exposes the key benefits of the platform for bringing together a user's devices, files and applications, and social graph, with news feeds across all of these.

Standards

- Standards make the World Wide Web go around, and by extension, they are important to cloud computing. Standards are what make it possible to connect to the cloud and what make it possible to develop and deliver content.

3.2.1 Applications

A cloud application is the software architecture that the cloud uses to eliminate the need to install and run on the client computer. There are many applications that can run, but there needs to be a standard way to connect between the client and the cloud.

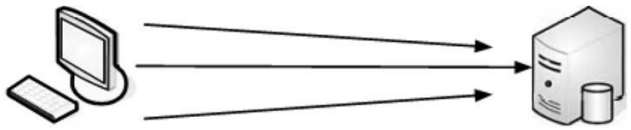
a. Communication:

HTTP

To get a web page from our cloud provider, we will likely be using the Hypertext Transfer Protocol (HTTP) as the computing mechanism to transfer data between the cloud and our organization. HTTP is a stateless protocol. This is beneficial because hosts do not need to retain information about users between requests, but this forces web developers to use alternative methods for maintaining users' states. HTTP is the language that the cloud and our computers use to communicate.

XMPP

The Extensible Messaging and Presence Protocol (XMPP) is being talked about as the next big thing for cloud computing.



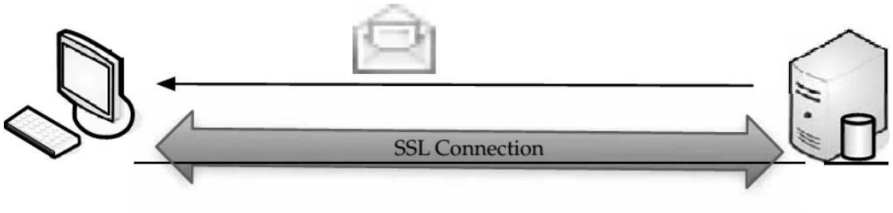
HTTP requires multiple polling events to update status from the web browser.

XMPP maintains a connection between the client and the web server.

The Problem with ***Polling*** When we wanted to sync services between two servers, the most common means was to have the client ping the host at regular intervals. This is known as polling. This is generally how we check our email. Every so often, we ping our email server to see if we got any new messages. It's also how the APIs for most web services work.

b. security

- SSL is the standard security technology for establishing an encrypted link between a web server and browser. This ensures that data passed between the browser and the web server stays private. To create an SSL connection on a web server requires an SSL certificate. When our cloud provider starts an SSL session, they are prompted to complete a number of questions about the identity of their company and web site. The cloud provider's computers then generate two cryptographic keys—a public key and a private key.



1. The browser checks the web site's certificate to ensure that the site you are connecting to is the real site and not someone else intercepting and spoofing the site.
2. The browser and web site decide on what type of encryption to use.
3. The browser and server send each other unique codes to use when encrypting information to be sent.
4. The browser and server use the encryption to start talking.
5. The browser shows the encrypting icon, and web pages are passed as secured.

3.2.2 Client

a.HTML

- HTML is to improve its usability and functionality.W3C is the organization that is charged with designing and maintaining the language. When you click on a link in a web page, you are accessing HTML code in the form of a hyperlink, which then takes you to another page.

How HTML works??

i. HTML is a series of short codes typed into a text file called TAGS which is created by web page design software.

ii. This text is saved as an HTML file and viewed through a browser.

iii. The browser reads the file and translates the text into the form the author wanted you to see.

- Writing HTML can be done using a number of methods, with either a simple text editor or a powerful graphical editor.
- Tags are seen like normal text but in <angle brackets>.tags is what allows things like tables and images to appear in a web page.
- Different tags perform different functions. Here tags cannot be seen through browser but affects how the browser behaves.
-

b.DHTML: There are four parts to DHTML:

DOM: (Document Object Model) allows you to access web page and makes changes with DHTML.DOM specifies every part of a web page and provides consistent naming conventions, allowing accessing your web pages and changing their properties.

Scripts: common scripting language in DHTML. Are java scripts and ActiveX. Scripts are used to control the objects specified in the DOM.

CSS: (Cascading style sheets) are used to control the look and feel of web page, where style sheets list the color and font s of text, the background colors and images, and the placement of objects on the page. Using scripting and DOM you can change the style of various elements.

XHTML: nothing unique about XHTML but it is important because there are more things working from it than just the browser.

DHTML features:

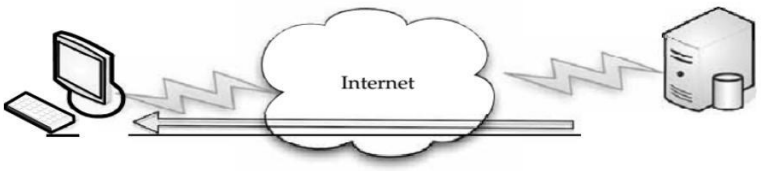
Four main features are:

i. Changing the tags and properties
ii. Real-time positioning
iii. Dynamic fonts
iv. Data binding

3.2.3 Infrastructure

Infrastructure is a way to deliver virtualization to our cloud computing solution.

a.Virtualization: Whenever something new happens in the world of computing, competitors duke it out to have their implementation be the standard. Virtualization is somewhat different, and major players worked together to develop a standard.



In a virtualized environment, applications run on a server and are displayed on the client. The server can be local or on the other side of the cloud.

VMware, AMD, BEA Systems, BMC Software, Broadcom, Cisco, Computer Associates International, Dell, Emulex, HP, IBM, Intel, Mellanox, Novell, QLogic, and Red Hat all worked together to advance open virtualization standards. VMware says that it will provide its partners with access to VMware ESX Server source code and interfaces under a new program called VMware Community Source. This program is designed to help partners influence the direction of VMware ESX Server through a collaborative development model and shared governance process.

These initiatives are intended to benefit end users by :

i. **Expanding virtualization solutions** the availability of open-standard virtualization interfaces and the collaborative nature of VMware Community Source are intended to accelerate the availability of new virtualization solutions.

ii. **Expanded interoperability and supportability** Standard interfaces for hypervisors are expected to enable interoperability for customers with heterogeneous virtualized environments.

iii. **Accelerated availability of new virtualization-aware technologies** Vendors across the technology stack can optimize existing technologies and introduce new technologies for running in virtual environments.

Open Hypervisor Standards

Hypervisors are the foundational component of virtual infrastructure and enable computer system partitioning. An open-standard hypervisor framework can benefit customers by enabling innovation across an ecosystem of interoperable virtualization vendors and solutions.

VMware contributed an existing framework of interfaces, called Virtual Machine Hypervisor Interfaces (VMHI), based on its virtualization products to facilitate the development of these standards in an industry-neutral manner.

Community Source

The Community Source program provides industry partners with an opportunity to access VMware ESX Server source code under a royalty-free license. Partners can contribute shared code or create binary modules to spur and extend interoperable and integrated virtualization solutions. The idea is to combine the best of both the traditional commercial and open-source development models. Community members can participate and influence the governance of VMware ESX Server through an architecture board.

b.OVF

As the result of VMware and its industry partners' efforts, a standard has already been developed called the Open Virtualization Format (OVF). OVF describes how virtual appliances can be packaged in a vendor-neutral format to be run on any hypervisor. It is a platform-independent, extensible, and open specification for the packaging and distribution of virtual appliances composed of one or more virtual machines.

VMware developed a standard with these features:

- **Optimized for distribution**
  - ✓ Enables the portability and distribution of virtual appliances
  - ✓ Supports industry-standard content verification and integrity checking
  - ✓ Provides a basic scheme for the management of software licensing
- **A simple, automated user experience**
  - ✓ Enables a robust and user-friendly approach to streamlining the installation process
  - ✓ Validates the entire package and confidently determines whether each virtual machine should be installed
  - ✓ Verifies compatibility with the local virtual hardware
- Portable virtual machine packaging
  - ✓ Enables platform-specific enhancements to be captured

✓Supports the full range of virtual hard disk formats used for virtual machines today, and is extensible to deal with future formats that are developed

✓ Captures virtual machine properties concisely and accurately

- Vendor and platform independent

✓ Does not rely on the use of a specific host platform, virtualization platform, or guest operating system

- Extensible

✓ Designed to be extended as the industry moves forward with virtual appliance technology

- Localizable

✓ Supports user-visible descriptions in multiple locales

✓ Supports localization of the interactive processes during installation of an appliance

✓ Allows a single packaged appliance to serve multiple market opportunities

3.2.4 Service

- A web service, as defined by the World Wide Web Consortium (W3C), "is a software system designed to support interoperable machine-to-machine interaction over a network" that may be accessed by other cloud computing components. Web services are often web API's that can be accessed over a network, like the Internet, and executed on a remote system that hosts the requested services.

a.Data

Data can be stirred and served up with a number of mechanisms; two of the most popular are JSON and XML.

JSON

JSON is short for JavaScript Object Notation and is a lightweight computer data interchange format. It is used for transmitting structured data over a network connection in a process called serialization. It is often used as an alternative to XML.

JSON Basics JSON is based on a subset of JavaScript and is normally used with that language. However, JSON is considered to be a language-independent format, and code for parsing and generating JSON data is available for several programming languages. This makes it a good replacement for XML when JavaScript is involved with the exchange of data, like AJAX.

XML vs. JSON:

JSON should be used instead of XML when JavaScript is sending or receiving data. The reason for this is that when we use XML in JavaScript, we have to write scripts or use libraries to handle the DOM objects to extract the data our need. However, in JSON, the object is already an object, so no extra work needs to be done.

Example The following is a sample JSON representation of an object describing a person:

```
{
     "firstName": "Johnny",
     "lastName": "Johnson",
     "address": {
          "streetAddress": "123 Main Street",
          "city": "Minneapolis",
          "state": "MN",
          "postalCode": 55102
     },
     "phoneNumbers": [
          "612 555-9871",
          "952 555-1598"
     ]
 }
```

XML

Extensible Markup Language (XML) is a standard, self-describing way of encoding text and data so that content can be accessed with very little human interaction and exchanged across a wide variety of hardware, operating systems, and applications. XML provides a standardized way to represent text and data in a format that can be used across platforms. It can also be used with a wide range of development tools and utilities.

HTML vs XML

✓ Separation of form and content HTML uses tags to define the appearance of text, while XML tags define the structure and the content of the data. Individual applications will be specified by the application or associated style sheet.

✓ XML is extensible Tags can be defined by the developer for specific application, while HTML's tags are defined by W3C.

Benefits of XML include:

i. **Self-describing data** XML does not require relational schemata, file description tables, external data type definitions, and so forth. Also, while HTML only ensures the correct presentation of the data, XML also guarantees that the data is usable.

ii. **Database integration** XML documents can contain any type of data—from text and numbers to multimedia objects to active formats like Java.

iii. **No reprogramming** if modifications are made Documents and web sites can be changed with XSL Style Sheets, without having to reprogram the data.

iv. **One-server view of data** XML is exceptionally ideal for cloud computing, because data spread across multiple servers looks as if it is stored on one server.

v. **Open and extensible** XML's structure allows us to add other elements if we need them. We can easily adapt our system as our business changes.

vi. **Future-proof** The W3C has endorsed XML as an industry standard, and it is supported by all leading software providers. It's already become industry standard in fields like healthcare.

vii. **Contains machine-readable context information** Tags, attributes, and element structure provide the context for interpreting the meaning of content, which opens up possibilities for development.

✓**Content vs. presentation** XML tags describe the meaning of the object, not its presentation. That is, XML describes the look and feel of a document, and the application presents it as described.

b.Web Services

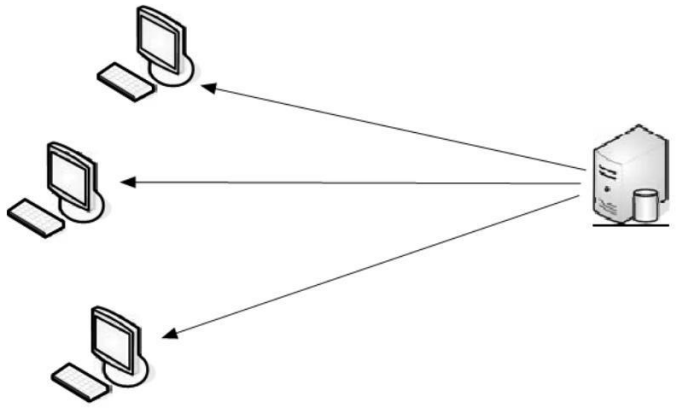
Web services describe how data is transferred from the cloud to the client.

REST

Representational state transfer (REST) is a way of getting information content from a web site by reading a designated web page that contains an XML file that describes and includes the desired content.

For instance, REST could be used by our cloud provider to provide updated subscription information. Every so often, the provider could prepare a web page that includes content and XML statements that are described in the code. Subscribers only need to know the uniform

resource locator (URL) for the page where the XML file is located, read it with a web browser, understand the content using XML information, and display it appropriately.



Clients send a request to the web server for information, using the same URL. The web site has updated its content, and uses REST to send the information back to the clients.

REST is similar in function to the Simple Object Access Protocol (SOAP), but is easier to use. SOAP requires writing or using a data server program and a client program (to request the data). However, SOAP offers more capability. For instance, if we were to provide syndicated content from our cloud to subscribing web sites, those subscribers might need to use SOAP, which allows greater program interaction between the client and the server.

Benefits REST offers the following benefits:

✓It gives better response time and reduced server load due to its support for the caching of representations.

✓ Server scalability is improved by reducing the need to maintain session state.

✓A single browser can access any application and any resource, so less client-side software needs to be written.

✓A separate resource discovery mechanism is not needed, due to the use of hyperlinks in representations.

- ✓ Better long-term compatibility and resolvability characteristics exist than in RPC. This is due to:
  - The ability of documents, like HTML, to evolve with both forward- and backward-compatibility.
  - Resources can add support for new content types as they are defined, without eliminating support for older content types.

SOAP

- Simple Object Access Protocol (SOAP) is a way for a program running in one kind of operating system (such as Windows Vista) to communicate with a program in the same or another kind of an operating system (such as Linux) by using HTTP and XML as the tools to exchange information.
- Procedure Calls Often, remote procedure calls (RPC) are used between objects like DCOM or COBRA, but HTTP was not designed for this use. RPC is a compatibility problem, because firewall and proxy servers will block this type of traffic. Because web protocols already are installed and available for use by the major operating systems, HTTP and XML provide an easy solution to the problem of how programs running under different operating systems in a network can communicate with each other.
- SOAP describes exactly how to encode an HTTP header and an XML file so that a program on one computer can call a program in another computer and pass it information. It also explains how a called program can return a response.

SOAP was developed by Microsoft, DevelopMentor, and Userland Software.

- One of the advantages of SOAP is that program calls are more likely to get through firewalls that normally screen out requests for those applications. Because HTTP requests are normally allowed through firewalls, programs using SOAP can communicate with programsanywhere.

Sample When you look at the following SOAP example, you can see how it is based on HTTP. In fact, the first line in the request is nearly identical to a standard HTTP request. Here is the request fully written out:

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

And like a standard HTTP response, a SOAP response follows the similar format. Here is a sample SOAP response:

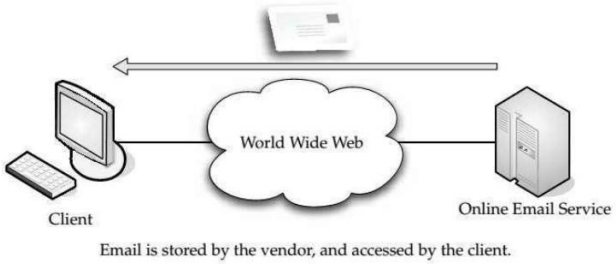
```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```

Standards are extremely important, and something that we take for granted these days. For instance, it's nothing for us to email Microsoft Word documents back and forth and expect them to work on our computers.

Software as a Service

SaaS (Software as a Service) is an application hosted on a remote server and accessed through the Internet.

An easy way to think of SaaS is the web-based email service offered by such companies as Microsoft (Hotmail), Google (Gmail), and Yahoo! (Yahoo Mail). Each mail service meets the



Email is stored by the vendor, and accessed by the client.

basic criteria: the vendor (Microsoft, Yahoo, and so on) hosts all of the programs and data in a central location, providing end users with access to the data and software, which is accessed across the World Wide Web.

SaaS can be divided into two major categories:

• Line of business services these are business solutions offered to companies and enterprises. They are sold via a subscription service. Applications covered under this category include business processes, like supply- chain management applications, customer relations applications, and similar business-orientedtools.

• Customer-oriented services These services are offered to the general public on a subscription basis. More often than not, however, they are offered for free and supported by advertising. Examples in this category include the aforementioned web mail services, online gaming, and consumer banking, among others.

Advantages

• There's a faster time to value and improved productivity, when compared to the long implementation cycles and failure rate of enterprise software.

• There are lower software licensing costs. • SaaS offerings feature the biggest cost savings over installed software by eliminating

the need for enterprises to install and maintain hardware, pay labor costs, and maintain the applications.

• SaaS can be used to avoid the custom development cycles to get applications to theorganization quickly.

• SaaS vendors typically have very meticulous security audits. SaaS vendors allow companies to have the most current version of an application as possible. This allows the organization to spend their development dollars on new innovation in their industry, rather than supporting old versions of applications.

SaaS, on the other hand, has no licensing. Rather than buying the application, you pay for it through the use of a subscription, and you only pay for what you use. If you stop using the application, you stop paying.

You only pay for the applications when you use them.

3.3.2.Vendor Advantages

- SaaS is advantage to Vendors also.And financial benefit is the top one—vendors get a constant stream of income, often what is more than the traditional software licensing setup. Additionally, through SaaS, vendors can fend off piracy concerns and unlicensed use of software.
- Vendors also benefit more as more subscribers come online. They have a huge investment in physical space, hardware, technology staff, and process development.

The more these resources are used to capacity, the more the provider can clear as margin.

Virtualization Benefits

- Virtualization makes it easy to move to a SaaS system. One of the main reasons is that it is easier for independent software vendors (ISVs) to adopt SaaS is the growth of virtualization. The growing popularity of some SaaS vendors using Amazon's EC2 cloud platform and the overall popularity of virtualized platforms help with the development of SaaS.

3.3.3.Companies Offering SaaS Intuit

- QuickBooks has been around for years as a conventional application for tracking business accounting. With the addition of QuickBooks online, accounting has moved to the cloud. QuickBooks Overview QuickBooks Online (www.qboe.com) gives small business owners the ability to access their financial data whether they are at work, home, or on the road. Intuit Inc. says the offering also gives users a high level of security because data is stored on firewall-protected servers and protected via automatic data backups.
- There is also no need to hassle with technology—software upgrades are included at no extra charge.
- For companies that are growing, QuickBooks Online Plus offers advanced features such as automatic billing and time tracking, as well as the ability to share information with employees in multiple locations.
- QuickBooks Online features include :

• The ability to access financial data anytime and from anywhere. QuickBooks Online is accessible to users 24 hours a day, seven days a week.

• Automated online banking. Download bank and credit card transactions automatically every night, so it's easy to keep data up to date.

• Reliable automatic data backup. Financial data is automatically backed up every day and is stored on Intuit's firewall-protected servers, which are monitored to keep critical business information safe and secure. QuickBooks Online also supports 128-bit Secure Sockets Layer

(SSL) encryption.

• No software to buy, install, or maintain and no network required. The software is hosted online, so small business users never have to worry about installing new software or upgrades. QuickBooks Online remembers customer, product, and vendor information, so users don't have to re-enter data.

• Easy accounts receivable and accounts payable. Invoice customers and track customer payments. Create an invoice with the click of a button. Apply specific credits to invoices or apply a single-customer payment to multiple jobs or invoices. Receive bills and enter them into QuickBooks Online with the expected duedate.

• Write and print checks. Enter information in the onscreen check form and print checks.

Google

- Google's SaaS offerings include Google Apps and Google Apps Premier Edition.

Google Apps, launched as a free service in August 2006, is a suite of applications that includes Gmail webmail services, Google Calendar shared calendaring, Google Talk instant messaging and Voice over IP, and the Start Page feature for creating a customizable home page on a specific domain.

- Google also offers Google Docs and Spreadsheets for all levels of Google Apps. Additionally, Google Apps supports Gmail for mobile on BlackBerry handheld devices.

Google Apps Premier Edition has the following unique features:

• Per-user storage of 10GBs Offers about 100 times the storage of the average corporate mailbox.

• APIs for business integration APIs for data migration, user provisioning, single sign-on, and mail gateways enable businesses to further customize the service for unique environments.

• Uptime of 99.9 percent Service level agreements for high availability of Gmail, with Google monitoring and crediting customers if service levels are not met.

• Advertising optional Advertising is turned off by default, but businesses can choose to include Google's relevant target-based ads if desired.

• Low fee Simple annual fee of $50 per user account per year makes it practical to offer these

applications to select users in the organization.

Microsoft

Microsoft Office Live Small Business offers features including Store Manager, an e-commerce tool to help small businesses easily sell products on their own web site and on eBay; and E-mail Marketing beta, to make sending email newsletters and promotions simple and affordable.

The following features are available in Microsoft Office Live Small Business:

• Store Manager is a hosted e-commerce service that enables users to easily sell products on their own web site and on eBay.

• Custom domain name and business email is available to all customers for free for one year. Private domain name registration is included to help customers protect their contact information from spammers. Business email now includes 100 company-branded accounts, each with 5GB of storage.

• Web design capabilities, including the ability to customize the entire page, as well as the header, footer, navigation, page layouts, and more.

• Support for Firefox 2.0 means Office Live Small Business tools and features are now compatible with Macs.

• A simplified sign-up process allows small business owners to get started quickly. Users do not have to choose a domain name at sign-up or enter their credit card information.

• Domain flexibility allows businesses to obtain their domain name through any provider and redirect it to Office Live Small Business. In addition, customers may purchase additional domain names.

• Synchronization with Microsoft Office Outlook provides customers with access to vital business information such as their Office Live Small Business email, contacts, and calendars, both online and offline.

• E-mail Marketing beta enables users to stay connected to current customers and introduce themselves to new ones by sending regular email newsletters, promotions, and updates.

IBM

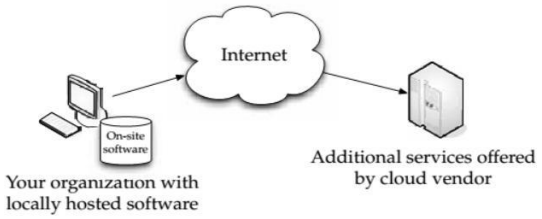
Big Blue—IBM offers its own SaaS solution under the name "Blue Cloud."

Blue Cloud is a series of cloud computing offerings that will allow corporate datacenters to operate more like the Internet by enabling computing across a distributed, globally accessible fabric of resources, rather than on local machines or remote server farms.

Blue Cloud is based on open-standards and open-source software supported by IBM software, systems technology, and services. IBM's Blue Cloud development is supported by more than 200 IBM Internet-scale researchers worldwide and targets clients who want to explore the extreme scale of cloud computing infrastructures.

Software plus Services

Software plus Services takes the notion of Software as a Service (SaaS) to complement packaged software. Here are some of the ways in which it can help the client organization.



With Software plus Services, locally hosted software is supplemented by cloud offerings.

3.4.1 Overview

- **User experience**: Browsers have limitations as to just how rich the user experience can be. Combining client software that provides the features we want with the ability of the Internet to deliver thoseexperiences gives us the best of both worlds.

• **Working offline** Not having to always work online gives us the flexibility to do our work, but without the limitations of the system being unusable. By connecting occasionally and synching data, we get a good solution for road warriors and telecommuters who don't have the same bandwidth or can't always be connected.

• **Privacy worries**: No matter how we use the cloud, privacy is a major concern. With Software plus Services, we can keep the most sensitive data housed on-site, while less

sensitive data can be moved to thecloud.

- **Marketing:** Software plus Services gives vendors a chance to keep their names in front of clients. Since it's so easy to move from vendor to vendor, providing a part software/part-Internet solution makes it easier to sell our product to a client.
- **Power:** More efficiency is realized by running software locally and synching to the cloud as needed.
- **Flexibility:** Vendors can offer software in different sizes and shapes—whether onsite or hosted. This gives customers an opportunity to have the right-sized solution.

Software plus Services offerings that prevalent companies have.

Vendors

i. **Microsoft :** Microsoft offers Dynamics CRM, Microsoft Outlook, Windows Azure, and Azure Services Platform. Windows Azure is a collection of cloud-based services, including Live Framework, .NET Services, SQL Services, CRM Services, SharePoint Services, and Windows Azure Foundation Services for compute, storage, and management.

ii. **Adobe:** Adobe Integrated Runtime (AIR) brings Flash, ActionScript, and MXML/Flex to the PC. Using AIR, vendors can build desktop applications that access the cloud.

iii. **Salesforce.com** : Salesforce.com's AppExchange is a set of APIs that vendors can use to create desktop applications to access salesforce data and run on the desktop of an end user.

iv. **Apple**: Apple offers a number of cloud-enabled features for its iPhone/iPod touch. Not only does it come with an integrated Safari web browser, but they also offer a software developer's kit (SDK) that allows software to be created for the iPhone/ iPod touch. Vendors can build their own applications, and on-the-go users can access cloud offerings with those applications.

v. **Google:** Google's mobile platform is called "Android" and helps vendors build software for mobile phones. Google also offers its Google Apps and the Google Chrome browser, which also installs Google Gears software on the desktop. This allows offline and online solutions.

3.4.2 Mobile Device Integration:

How Mobile Device Integration is done. How Microsoft Online provides this?

A key component of Software plus Services is the ability to work in the cloud from a mobile

device. Google Android:

A broad alliance of leading technology and wireless companies joined forces to develop Android, an open and comprehensive platform for mobile devices. Google Inc., T-Mobile, HTC, Qualcomm, Motorola, and others collaborated on the development of Android through the Open Handset Alliance, a multinational alliance of technology and mobile industry leaders.

Open Handset Alliance:

- Thirty-four companies have formed the Open Handset Alliance, which aims to develop technologies that will significantly lower the cost of developing and distributing mobile devices and services. The Android platform is the first step in this direction—a fully

integrated mobile "software stack" that consists of an operating system, middleware, and user-friendly interface and applications. This alliance include major companied like,

• Google (www.google.com)
• HTC (www.htc.com)
• Intel (www.intel.com)
• LG (www.lge.com)
• Marvell (www.marvell.com)
• Motorola (www.motorola.com)
• NMS Communications (www.nmscommunications.com)
• NTT DoCoMo Inc. (www.nttdocomo.com)
• Qualcomm (www.qualcomm.com)
• Samsung (www.samsung.com) Etc...

3.4.3. Providers:

The following development solutions we may consider for creating our own Software plus Services deployments.

a.Adobe AIR:

Adobe Systems offers its Adobe Integrated Runtime (AIR), formerly code-named Apollo. Adobe AIR is a cross- operating-system application runtime that allows developers to use HTML/CSS, AJAX, Adobe Flash, and Adobe Flex to extend rich Internet applications (RIAs) to the desktop.

- For its popular iPhone and iPod touch devices, Apple offers its iPhone Software

Development Kit (SDK) as well as enterprise features such as support for Microsoft Exchange ActiveSync to provide secure, over-the-air push email, contacts, and calendars as well as remote wipe, and the addition of Cisco IPsec VPN for encrypted access to private corporate networks.

App Store:

The iPhone software contains the App Store, an application that lets users browse, search, purchase, and wirelessly download third-party applications directly onto their iPhone or iPod touch. The App Store enables developers to reach every iPhone and iPod touch user. Developers set the price for their applications (including free) and retain 70 percent of all sales revenues. Users can download free applications at no charge to either the user or developer, or purchase priced applications with just one click. Enterprise customers can create a secure, private page on the App Store accessible only by their employees.

3.4.4 Microsoft Online:

Microsoft provides Software plus Services offerings, integrating some of its most popular and prevalent offerings, like Exchange. Not only does Microsoft's Software plus Services offering allow a functional way to serve our organization, but it also provides a means to function on the cloud in simple way.

Hybrid Model

With Microsoft services like Exchange Online, SharePoint Online, and CRM 4.0, organizations big and small have more choices in how they access and manage enterprise from entirely web-based, to entirely on-premise solutions, and anywhere in between. Having a variety of solutions to choose from gives customers the mobility and flexibility they need to meet constantly evolving business needs. To meet this demand, Microsoft is moving toward a hybrid strategy of Software plus Services, the goal of which is to empower customers and partners with richer applications, more choices, and greater opportunity through a combination of on-premise software, partner-hosted software, and Microsoft-hosted software. As part of this strategy, Microsoft expanded its Microsoft Online Services which includes Exchange Online and SharePoint Online to organizations of all sizes. With services like Microsoft Online Services

and Microsoft Dynamics CRM 4.0, organizations will have the flexibility required to address their business needs.

Exchange Online and SharePoint Online

Exchange Online and SharePoint Online are two examples of how partners can extend their reach, grow their revenues, and increase the number to sales in a Microsoft-hosted scenario. In September 2007, Microsoft initially announced the worldwide availability of Microsoft Online Services—which includes Exchange Online, SharePoint Online, Office Communications Online, and Office Live Meeting—to organizations with more than 5,000 users. The extension of these services to small and mid-sized businesses is appealing to partners in the managed services space because they see it as an opportunity to deliver additional services and customer value on top of Microsoft-hosted Exchange Online or SharePoint Online. Microsoft Online Services opens the door for partners to deliver reliable business services such as desktop and mobile email, calendaring and contacts, instant messaging, audio and video conferencing, and shared workspaces—all of which will help increase their revenue stream and grow their businesses.

Microsoft Dynamics CRM 4.0:

Microsoft Dynamics CRM 4.0, released in December of 2007 which provides a key aspect of Microsoft's Software plus Services strategy. The unique advantages of the new Microsoft Dynamics CRM 4.0, which can be delivered on-premise or on-demand as a hosted solution, make Microsoft Dynamics CRM an option for solution providers who want to rapidly offer a solution that meets customer needs and maximizes their potential to grow their own business through additional services.

UNIT IV

4.1 Developing Applications

In cloud computing we can develop our own applications to cater the needs of our business. A simple example is developing an app using Android to meet our business using Google App Engine and deploy in App Store. Similarly we may use Intuit's QuickBase which allows us to develop financial-based cloud apps.

4.1.1 Google

To develop an app on the cloud, the Google App Engine is the perfect tool to use, to make this dream become reality. In essence, we will write a bit of code in Python, tweak some HTML code, and then we have our app built, and it only takes a few minutes. Using Google App Engine we can develop our applications without worry about buying servers, load balancers, or DNS tables. Salesforce.com struck up a strategic alliance with Google with the availability of Force.com for Google App Engine. Force.com for Google App Engine is a set of tools and services to enable developer success with application development in the cloud. The offering brings together Force.com and Google App Engine, enabling the creation of entirely new web and business applications. Force.com for Google App Engine builds on the relationship between Salesforce.com and Google, spanning philanthropy, business applications, social networks, and cloud computing.

a) Google Gears

Another development tool that Google offers is Google Gears, an open-source technology for creating offline web applications. This browser extension was made available in its early stages so that the development community could test its capabilities and limitations and help Google improve upon it. Google's long-term hope is that Google Gears can help the industry as a whole move toward a single standard for offline capabilities that all developers can use.

Gears provide three key features:

- A local server, to cache and serve application resources (HTML, JavaScript, images, etc.) without needing to contact a server
- A database, to store and access data from within the browser
- A worker thread pool, to make web applications more responsive by performing expensive operations in the background .

4.1.2 Microsoft

Microsoft's Azure Services Platform is a tool provided for developers who want to write applications that are going to run partially or entirely in a remote datacenter. The Azure Services Platform (Azure) is an Internet-scale cloud services platform hosted in Microsoft datacenters, which provides an operating system and a set of developer services that can be used individually or together. Azure can be used to build new applications to run from the cloud or to enhance existing applications with cloud-based capabilities, and it forms the foundation of all Microsoft's cloud offerings. Its open architecture gives developers the choice to build web applications, applications running on connected devices, PCs, servers, or hybrid solutions offering the best of online and on premises.

Microsoft also offers cloud applications ready for consumption by customers such as Windows Live, Microsoft Dynamics, and other Microsoft Online Services for business such as Microsoft Exchange Online and SharePoint Online. The Azure Services Platform lets developers provide their own unique customer offerings by offering the foundational components of compute, storage, and building block services to author and compose applications in the cloud. Azure utilizes several other Microsoft services as part of its platform, known as the Live Mesh platform.

a) Live Services:

Live Services is a set of building blocks within the Azure Services Platform that is used to handle user data and application resources. Live Services provides developers with a

way to build social applications and experiences across a range of digital devices that can connect with one of the largest audiences on the Web.

b) Microsoft SQL Services:

Microsoft SQL Services enhances the capabilities of Microsoft SQL Server into the cloud as a web-based, distributed relational database. It provides web services that enable relational queries, search, and data synchronization with mobile users, remote offices, and business partners.

c) Microsoft .NET Services :

- Microsoft .NET Services is a tool for developing loosely coupled cloud-based applications. .NET Services includes access control to help secure applications, a service bus for communicating across applications and services, and hosted workflow execution. These hosted services allow the creation of applications that span from on-premises environments to the cloud.
- Microsoft SharePoint Services and Dynamics CRM Services are used to allow developers to collaborate and build strong customer relationships. Using tools like Visual Studio, developers can build applications that utilize SharePoint and CRM capabilities.

d) Microsoft Azure Design

- Azure is designed in several layers, with different things going on under the hood, Layer Zero
- Layer Zero is Microsoft's Global Foundational Service. GFS is akin to the hardware abstraction layer (HAL) in Windows. It is the most basic level of the software that interfaces directly with the servers.

Layer One

Layer One is the base Azure operating system. It used to be code-named "Red Dog," and was designed by a team of operating system experts at Microsoft. Red Dog is the technology that networks and manages the Windows Server 2008 machines that form the

Microsoft hosted cloud.
Red Dog is made up of four pillars:

- Storage (a file system)
- The fabric controller, which is a management system for deploying and provisioning
- Virtualized computation/VM
- Development environment, which allows developers to emulate Red dog on their desktops

Layer Two:

Layer Two provides the building blocks that run on Azure. These services are the aforementioned Live Mesh platform. Developers build on top of these lower-level services when building cloud apps.
SharePoint Services and CRM Services are not the same as SharePoint Online and CRM Online. They are just the platform basics that do not include user interface elements.

Layer Three

At Layer Three exist the Azure-hosted applications. Some of the applications developed by Microsoft include SharePoint Online, Exchange Online, Dynamics CRM, and Online. Third parties will create other applications.

4.1.3 Intuit QuickBase:

Intuit Inc.'s QuickBase launched its new QuickBase Business Consultant Program. The program allows members to use their expertise to create unique business applications tailored specifically to the industries they serve—without technical expertise or coding. This helps members expand their reach into industries formerly served only by IT experts. Using QuickBase, program members will be able to easily build new on- demand business applications from scratch or customize one of 200 available templates and resell them to their clients.

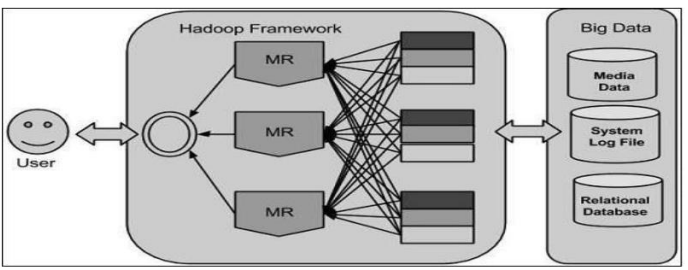
4.2 HADOOP

What??

Using the solution provided by Google, Doug Cutting and his team developed an Open Source Project called HADOOP

Why??

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.



Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed *storage* and *computation* across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

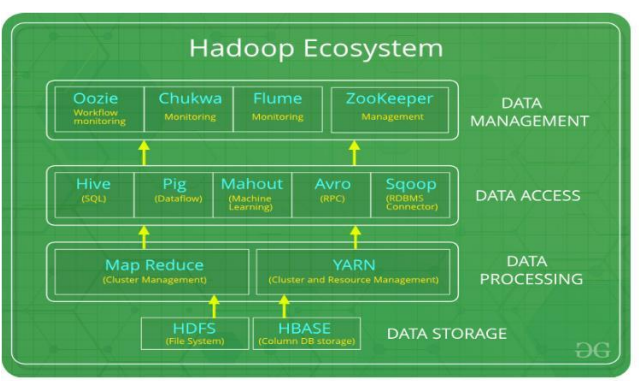
Hadoop Ecosystem

Introduction: *Hadoop Ecosystem* is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There

are *four major elements of Hadoop* i.e. HDFS, MapReduce, YARN, and Hadoop Common. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

Following are the components that collectively form a Hadoop ecosystem:

- HDFS: Hadoop Distributed File System
- YARN: Yet Another Resource Negotiator
- MapReduce: Programming based Data Processing
- Spark: In-Memory data processing
- PIG, HIVE: Query based processing of data services
- HBase: NoSQL Database
- Mahout, Spark MLLib: Machine Learning algorithm libraries
- Solar, Lucene: Searching and Indexing
- Zookeeper: Managing cluster
- Oozie: Job Scheduling



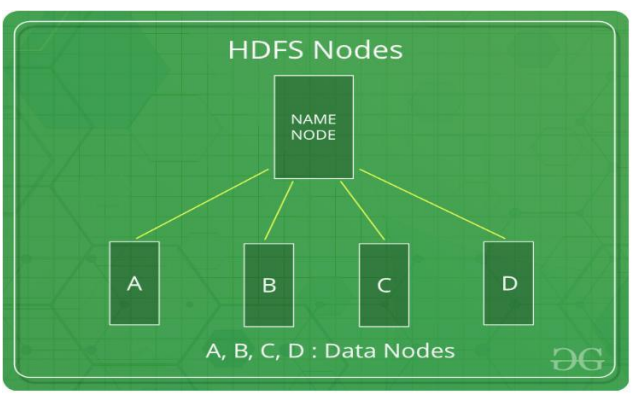
All these toolkits or components revolve around one term i.e. *Data*. That's the beauty of Hadoop that it revolves around data and hence making its synthesis easier.

HDFS:

- HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
- HDFS consists of two core components i.e.

1. Name node
2. Data Node

- Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment. Undoubtedly, making Hadoop cost effective.
- HDFS maintains all the coordination between the clusters and hardware, thus working at the heart of the system.



YARN:

- Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.
- Consists of three major components i.e.

1. Resource Manager
2. Nodes Manager
3. Application Manager

- Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

MapReduce:

- By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing's logic and helps to write applications which transform big data sets into a manageable one.
- MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:

1. *Map()* performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
2. *Reduce()*, as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

Pig:

- Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.

- It is a platform for structuring the data flow, processing and analyzing huge data sets.
- Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.
- Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the JVM.
- Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

Hive:

- With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
- It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
- Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
- JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

Mahout:

- Mahout, allows Machine Learnability to a system or application. Machine Learning, as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.
- It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

Apache Spark:

- It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.

- It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
- Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

Apache HBase:

- It's a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google's BigTable, thus able to work on Big Data sets effectively.
- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data.

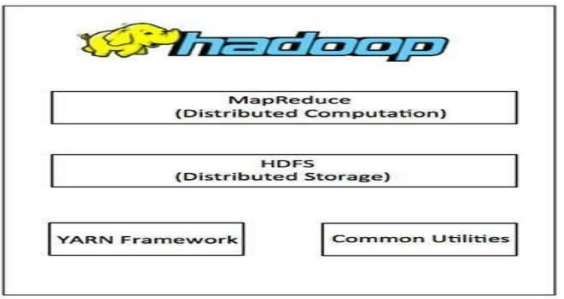
Other Components: Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

- Solr, Lucene: These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.
- Zookeeper: There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often. Zookeeper overcame all the problems by performing synchronization, inter-component based communication, grouping, and maintenance.
- Oozie: Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

Hadoop Architecture:

At its core, Hadoop has two major layers namely –

- Processing/Computation layer (MapReduce), and
- Storage layer (Hadoop Distributed File System).



How Does Hadoop Work?

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, you can tie together many commodity computers with single-CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. Moreover, it is cheaper than one high-end server. So this is the first motivational factor behind using Hadoop that it runs across clustered and low-cost machines.

Hadoop runs code across a cluster of computers. This process includes the following core tasks that Hadoop performs –

- Data is initially divided into directories and files. Files are divided into uniform sized blocks of 128M and 64M (preferably 128M).
- These files are then distributed across various cluster nodes for further processing.

- HDFS, being on top of the local file system, supervises the processing.
- Blocks are replicated for handling hardware failure.
- Checking that the code was executed successfully.
- Performing the sort that takes place between the map and reduce stages.
- Sending the sorted data to a certain computer.
- Writing the debugging logs for each job.

Advantages of Hadoop

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
- Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
- Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.
- Hadoop is supported by GNU/Linux platform and its flavors. Therefore, we have to install a Linux operating system for setting up Hadoop environment. In case you have an OS other than Linux, you can install a Virtualbox software in it and have Linux inside the Virtualbox.

MapReduce

- MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.
- MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.

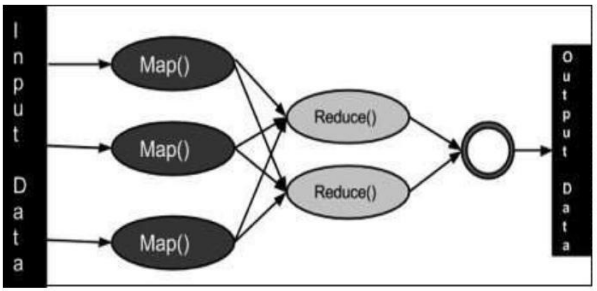
What is MapReduce?

- MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.
- The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a

cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

The Algorithm

- 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.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.



Inputs and Outputs (Java Perspective)

- The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a MapReduce job – (Input) <k1, v1> → map → <k2, v2> → reduce → <k3, v3>(Output).

	Input	Output
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)

Terminology

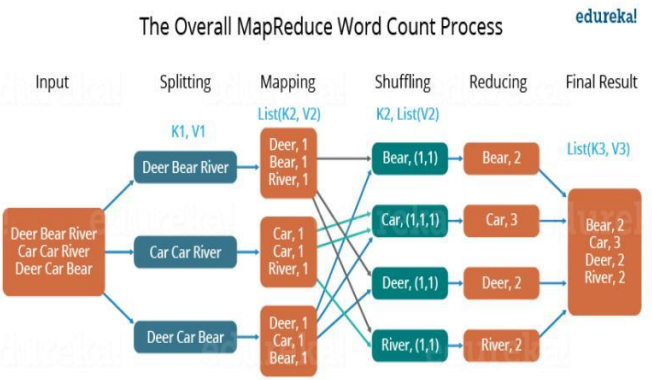
- PayLoad – Applications implement the Map and the Reduce functions, and form the core of the job.
- Mapper – Mapper maps the input key/value pairs to a set of intermediate key/value pair.
- NameNode – Node that manages the Hadoop Distributed File System (HDFS).
- DataNode – Node where data is presented in advance before any processing takes place.
- Master Node – Node where JobTracker runs and which accepts job requests from clients.
- Slave Node – Node where Map and Reduce program runs.
- Job Tracker – Schedules jobs and tracks the assign jobs to Task tracker.
- Task Tracker – Tracks the task and reports status to JobTracker.
- Job – A program of an execution of a Mapper and Reducer across a dataset.
- Task – an execution of a Mapper or a Reducer on a slice of data.
- Task Attempt – A particular instance of an attempt to execute a task on a Slave Node.

MapReduce Tutorial: A Word Count Example of MapReduce

Let us understand how a MapReduce works by taking an example where I have a text file called example.txt whose contents are as follows:

Dear, Bear, River, Car, Car, River, Deer, Car and Bear

Now, suppose, we have to perform a word counts on the sample.txt using MapReduce. So, we will be finding the unique words and the number of occurrences of those unique words.



- First, we divide the input in three splits as shown in the figure. This will distribute the work among all the map nodes.
- Then, we tokenize the words in each of the mapper and give a hardcoded value (1) to each of the tokens or words. The rationale behind giving a hardcoded value equal to 1 is that every word, in itself, will occur once.
- Now, a list of key-value pair will be created where the key is nothing but the individual words and value is one. So, for the first line (Dear Bear River) we have 3 key-value pairs – Dear, 1; Bear, 1; River, 1. The mapping process remains the same on all the nodes.
- After mapper phase, a partition process takes place where sorting and shuffling happens so that all the tuples with the same key are sent to the corresponding reducer.
- So, after the sorting and shuffling phase, each reducer will have a unique key and a list of values corresponding to that very key. For example, Bear, [1,1]; Car, [1,1,1].., etc.
- Now, each Reducer counts the values which are present in that list of values. As shown in the figure, reducer gets a list of values

which is [1,1] for the key Bear. Then, it counts the number of ones in the very list and gives the final output as – Bear, 2.

- Finally, all the output key/value pairs are then collected and written in the output file.

Hadoop Distributed File System

- The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications having large datasets.

Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules –

- Hadoop Common – these are Java libraries and utilities required by other Hadoop modules.
- Hadoop YARN – this is a framework for job scheduling and cluster resource management.

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware.

- HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

i) Fault Tolerance

The fault tolerance in Hadoop HDFS is the working strength of a system in unfavorable conditions. It is highly fault-tolerant. Hadoop framework divides data into blocks. After that creates multiple copies of blocks on different machines in the cluster. So, when any machine in the cluster goes down, then a client can easily access their data from the other machine which contains the same copy of data blocks.

ii) High Availability

Hadoop HDFS is a highly available file system. In HDFS, data gets replicated among the nodes in the Hadoop cluster by creating a replica of the blocks on the other slaves present in HDFS cluster. So, whenever a user wants to access this data, they can access their data from the slaves which contain its blocks. At the time of unfavorable situations like a failure of a node, a user can easily access their data from the other nodes. Because duplicate copies of blocks are present on the other nodes in the HDFS cluster.

iii) High Reliability

HDFS provides reliable data storage. It can store data in the range of 100s of petabytes. HDFS stores data reliably on a cluster. It divides the data into blocks. Hadoop framework stores these blocks on nodes present in HDFS cluster. HDFS stores data reliably by creating a replica of each and every block present in the cluster. Hence provides fault tolerance facility. If the node in the cluster containing data goes down, then a user can easily access that data from the other nodes. HDFS by default creates 3 replicas of each block containing data present in the nodes. So, data is quickly available to the users. Hence user does not face the problem of data loss. Thus, HDFS is highly reliable.

iv) Replication

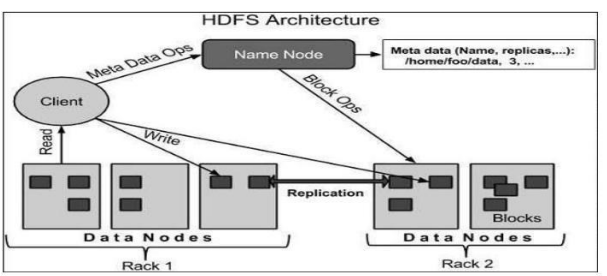
Data Replication is unique features of HDFS. Replication solves the problem of data loss in an unfavorable condition like hardware failure, crashing of nodes etc. HDFS maintain the process of replication at regular interval of time. HDFS also keeps creating replicas of user data on different machine present in the cluster. So, when any node goes down, the user can access the data from other machines. Thus, there is no possibility of losing of user data.

v) Scalability

Hadoop HDFS stores data on multiple nodes in the cluster. So, whenever requirements increase you can scale the cluster. Two scalability mechanisms are available in HDFS: Vertical and Horizontal Scalability.

HDFS Architecture

Given below is the architecture of a Hadoop File System.



- HDFS follows the master-slave architecture and it has the following elements.

a) Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is software that can be run on commodity hardware. The system having

the namenode acts as the master server and it does the following tasks –

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

b) Datanode

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.

- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

c) Block

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS

i) Fault detection and recovery – since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

ii) Huge datasets – HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

iii) Hardware at data – A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

UNIT V

HADOOP I/O

- Hadoop comes with a set of primitives for data I/O and the techniques that are more general than Hadoop, such as *data integrity* and *compression*, but deserve special consideration when dealing with multi-terabyte datasets.
- Others are Hadoop tools or APIs that form the building blocks for developing distributed system, such as *serialization frameworks* and *on-disk data structures*

5.1. Data integrity

The usual way of detecting corrupted data is by computing a *checksum* for the data when it first enters the system, and again whenever it is transmitted across a channel that is unreliable and hence capable of corrupting the data. The data is deemed to be corrupt if the newly generated checksum doesn't exactly match the original. This technique doesn't offer any way to fix the data—merely error detection. A commonly used error-detecting code is CRC-32 (cyclic redundancy check), which computes a 32-bit integer checksum for input of any size.

5.1.1 Data Integrity in HDFS

- HDFS transparently checksums all data written to it and by default verifies checksums when reading data. A separate checksum is created for every io.bytes.per.checksum bytes of data. The default is 512 bytes, and since a CRC-32 checksum is 4 bytes long, the storage overhead is less than 1%.
- Datanodes are responsible for verifying the data they receive before storing the data and its checksum. This applies to data that they receive from clients and from other datanodes during replication. A client writing data sends it to a pipeline of datanodes and the last datanode in the pipeline verifies the checksum. If it detects an error, the client receives a ChecksumException, a subclass of IOException, which it should handle in an application-specific manner, by retrying the operation, for example.
- When clients read data from datanodes, they verify checksums as well, comparing them with the ones stored at the datanode. Each datanode keeps a persistent log of checksum verifications, so it knows the last time each of its blocks was verified. When a client successfully verifies a block, it tells the datanode, which updates its log. Keeping statistics such as these is valuable in detecting bad disks.
- Aside from block verification on client reads, each datanode runs a DataBlockScanner in a background thread that periodically verifies all the blocks stored on the datanode. This is to guard against corruption due to "bit rot" in the physical storage media. See "Datanode block scanner" for details on how to access the scanner reports.
- Since HDFS stores replicas of blocks, it can "heal" corrupted blocks by copying one of the good replicas to produce a new, uncorrupt replica. The way this works is that if a client detects an error when reading a block, it reports the bad block and the datanode it was trying to read from to the namenode before throwing a ChecksumException. The namenode marks the block replica as corrupt, so it doesn't direct clients to it, or try to copy this replica to another datanode. It then schedules a copy of the block to be replicated on another datanode, so its replication factor is back at the expected level. Once this has happened, the corrupt replica is deleted.

- It is possible to disable verification of checksums by passing false to the setVerify Checksum() method on FileSystem, before using the open() method to read a file. The same effect is possible from the shell by using the -ignoreCrc option with the -get or the equivalent -copyToLocal command. This feature is useful if you have a corrupt file that you want to inspect so you can decide what to do with it. For example, you might want to see whether it can be salvaged before you delete it.

5.1.2 LocalFileSystem

The Hadoop LocalFileSystem performs client-side checksumming. This means that when you write a file called filename, the filesystem client transparently creates a hidden file, .filename.crc, in the same directory containing the checksums for each chunk of the file. Like HDFS, the chunk size is controlled by the io.bytes.per.checksum property, which defaults to 512 bytes. The chunk size is stored as metadata in the .crc file, so the file can be read back correctly even if the setting for the chunk size has changed. Checksums are verified when the file is read, and if an error is detected, LocalFileSystem throws a ChecksumException.

Checksums are fairly cheap to compute (in Java, they are implemented in native code), typically adding a few percent overhead to the time to read or write a file. For most pay for data integrity. It is, however, possible to disable checksums: typically when the underlying filesystem supports checksums natively. This is accomplished by using RawLocalFileSystem in place of Local FileSystem. To do this globally in an application, it suffices to remap the implementation for file URIs by setting the property fs.file.impl to the value org.apache.hadoop.fs.RawLocalFileSystem. Alternatively, you can directly create a Raw LocalFileSystem instance, which may be useful if you want to disable checksum verification for only some reads;
For example:

```
Configuration conf= ...
FileSystemfs = new RawLocalFileSystem();
fs.initialize(null, conf);
```

5.1.3 ChecksumFileSystem

LocalFileSystem uses ChecksumFileSystem to do its work, and this class makes it easy to add checksumming to other (nonchecksummed) filesystems, as Checksum FileSystem is just a wrapper around FileSystem. The general idiom is as follows:

```
FileSystemrawFs= ...
FileSystemchecksummedFs = new ChecksumFileSystem(rawFs);
```

The underlying filesystem is called the raw filesystem, and may be retrieved using the getRawFileSystem() method on checksumFileSystem. ChecksumFileSystem has a few more useful methods for working with checksums, such as getChecksumFile() for getting the path of a checksum file for any file. Check the documentation for the others.

If an error is detected by ChecksumFileSystem when reading a file, it will call its reportChecksumFailure () method. The default implementation does nothing, but LocalFileSystem moves the offending file and its checksum to a side directory on the same device called bad_files. Administrators should periodically check for these bad files and take action on them.

5.2 Compression

Table 4-1. A summary of compression formats

Compression format	Tool	Algorithm	Filename extension	Multiple files	Splittable
DEFLATE[3]	N/A	DEFLATE	*.deflate*	No	No
gzip	*gzip*	DEFLATE	*.gz*	No	No
ZIP	*zip*	DEFLATE	*.zip*	Yes	Yes, at file boundaries
bzip2	*bzip2*	bzip2	*.bz2*	No	Yes
LZO	*lzop*	LZO	*.lzo*	No	No

Table 4-2. Hadoop compression codecs

Compression format	Hadoop CompressionCodec
DEFLATE	org.apache.hadoop.io.compress.DefaultCodec
gzip	org.apache.hadoop.io.compress.GzipCodec
bzip2	org.apache.hadoop.io.compress.BZip2Codec
LZO	com.hadoop.compression.lzo.LzopCodec

- All of the tools listed in Table 4-1 give some control over this trade-off at compression time by offering nine different options

  -1 means optimize for speed and

  -9 means optimize for space

e.g :--- gzip-1 file

The different tools have very different compression characteristics.¬Both gzip and ZIPare general-purpose compressors, and sit in the middle of the space/time trade-off.

- Bzip2compresses more effectively than gzipor ZIP, but is slower.
- LZOoptimizes for speed. It is faster than gzipand ZIP, but compresses slightly less effectively

5.2.1 Codecs

- A codec is the implementation of a compression-decompression algorithm

- The LZO libraries are GPL-licensed and may not be included in Apache distributions, so for this reason the Hadoopcodecs must be downloaded separately from http://code.google.com/p/hadoop-gpl-compression/

Compressing and decompressing streams with CompressionCodec

- CompressionCodechas two methods that allow you to easily compress or decompress data.
- To compress data being written to an output stream, use the createOutputStream(OutputStreamout) method to create a CompressionOutputStream to which you write your uncompressed data to have it written in compressed form to the underlying stream.
  - To decompress data begin read from an input stream, call createIntputStream(InputStreamin) to obtain a CompressionInputStream, which allows you to read uncompressed data from the underlying stream.

```
String codecClassname = args[0];
Class<?>codecClass = Class.forName(codecClassname);
Configuration conf = new Configuration();
CompressionCodec codec = (CompressionCodec)
ReflectionUtils.newInstance(codecClass, conf);
```

```
        CompressionOutputStream out = codec.createOutputStream(System.out);
        IOUtils.copyBytes(System.in, out, 4096, false);
        out.finish();
```

Inferring CompressionCodecsusing CompressionCodecFactory

- If you are reading a compressed file, you can normally infer the codec to use by looking at its filename extension. A file ending in .gzcan be read with GzipCodec, and so on.
- CompressionCodecFactoryprovides a way of mapping a filename extension to a compressionCodecusing its getCodec() method, which takes a Path object for the file in question.
- Following example shows an application that uses this feature to decompress files.

```
String uri = args[0];
Configuration conf = new Configuration();
FileSystemfs = FileSystem.get(URI.create(uri), conf);
Path inputPath = new Path(uri);
CompressionCodecFactory factory = new CompressionCodecFactory(conf);
CompressionCodec codec = factory.getCodec(inputPath);
if (codec == null)
{
        System.err.println("No codec found for " + uri);
        System.exit(1);
}
String outputUri =
CompressionCodecFactory.removeSuffix(uri, codec.getDefaultExtension());
InputStream in = null;
OutputStream out = null;
try {
in = codec.createInputStream(fs.open(inputPath));
out = fs.create(new Path(outputUri));
IOUtils.copyBytes(in, out, conf);
}
Finally
{
IOUtils.closeStream(in);
IOUtils.closeStream(out);
}
```

Native libraries

- For performance, it is preferable to use a native library for compression and decompression. For example, in one test, using the native gziplibraries reduced decompression times by up to 50%and compression times by around 10%(compared to the built-in Java implementation).
- Hadoopcomes with prebuilt native compression libraries for 32-and 64-bit Linux, which you can find in the lib/native directory.
- By default Hadooplooks for native libraries for the platform it is running on, and loads them automatically if they are found.

Table 4-4. Compression library implementations

Compression format	Java implementation	Native implementation
DEFLATE	Yes	Yes
gzip	Yes	Yes
bzip2	Yes	No
LZO	No	Yes

Native libraries –CodecPool

- If you are using a native library and you are doing a lot of compression or decompression in your application, consider using CodecPool, which allows you to reuse compressors and decompressors, thereby amortizing the cost of creating these objects.

```
String codecClassname = args[0];
Class<?>codecClass = Class.forName(codecClassname);
Configuration conf = new Configuration();
CompressionCodec codec = (CompressionCodec)
ReflectionUtils.newInstance(codecClass, conf);
Compressor compressor = null;
try
{
        compressor = CodecPool.getCompressor(codec);
        CompressionOutputStream out =
        codec.createOutputStream(System.out, compressor);
        IOUtils.copyBytes(System.in, out, 4096, false);
        out.finish();
        }
finally
{
        CodecPool.returnCompressor(compressor);
}
```

5.2.2 Compression and Input Splits

- When considering how to compress data that will be processed by MapReduce, it is important to understand whether the compression format supports splitting.
- Consider an uncompressed file stored in HDFS whose size is 1GB. With a HDFS block size of 64MB, the file will be stored as 16 blocks, and a Map Reduce job using this file as input will create 16 input splits, each processed independently as input to a separate map task.
- Imagine now the file is a gzip-compressed file whose compressed size is 1GB. As before, HDFS will store the file as 16 blocks. However, creating a split for each block won't work since it is impossible to start reading at an arbitrary point in the gzipstream, and therefore impossible for a map task to read its split independently of the others.
- In this case, Map Reduce will do the right thing, and not try to split the gzippedfile.This will work, but at the expense of locality. A single map will process the 16 HDFS blocks, most of which will not be local to the map. Also, with fewer maps, the job is less granular, and so may take longer to run.

5.2.3 Using Compression in MapReduce

If your input files are compressed, they will be automatically decompressed as they are read by MapReduce, using the filename extension to determine the codec to use.

For Example...

```
conf.setJobName("Max temperature with output compression");
FileInputFormat.addInputPath(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));

conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);

conf.setBoolean("mapred.output.compress", true);
conf.setClass("mapred.output.compression.codec", GzipCodec.class,
    CompressionCodec.class);

conf.setMapperClass(MaxTemperatureMapper.class);
conf.setCombinerClass(MaxTemperatureReducer.class);
```

```
conf.setCompressMapOutput(true);
conf.setMapOutputCompressorClass(GzipCodec.class);
```

Compressing map output

- Even if your Map Reduce application reads and writes uncompressed data, it may benefit from compressing the intermediate output of the map phase.
- Since the map output is written to disk and transferred across the network to the reducer nodes, by using a fast compressor such as LZO, you can get performance gains simply because the volume of data to transfer is reduced.
- Here are the lines to add to enable gzipmap output compression in your job:

5.3 Serialization

- Serialization is the process of turning structured objects into a byte stream for transmission over a network or for writing to persistent storage. Deserialization is the process of turning a byte stream back into a series of structured objects.

- In Hadoop, interprocess communication between nodes in the system is implemented using remote procedure calls(RPCs). The RPC protocol uses serialization to render the message into a binary stream to be sent to the remote node, which then deserializes the binary stream into the original message.

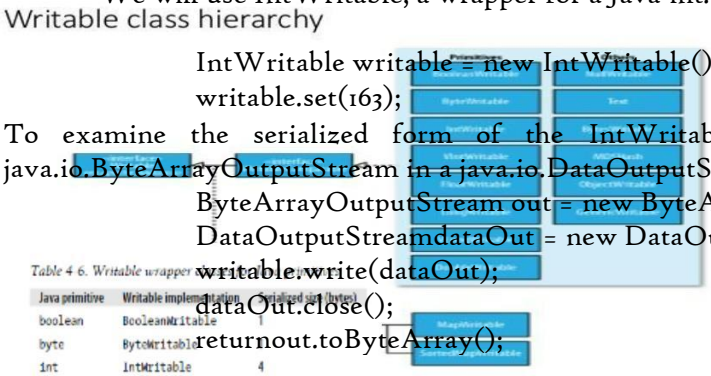
In general, it is desirable that an RPC serialization format is:

- Compact: A compact format makes the best use of network bandwidth
- Fast: Interprocess communication forms the backbone for a distributed system, so it is essential that there is as little performance overhead as possible for the serialization and deserialization process.
- Extensible: Protocols change over time to meet new requirements, so it should be straightforward to evolve the protocol in a controlled manner for clients and servers.
- Interoperable : For some systems, it is desirable to be able to support clients that are written in

different languages to the server.

5.3.1 Writable Interface

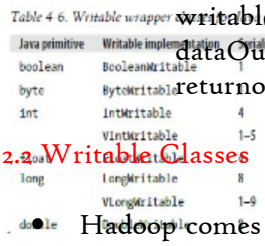
- The Writable interface defines two methods: one for writing its state to a DataOutput binary stream, and one for reading its state from a DataInput binary stream.
- We will use IntWritable, a wrapper for a Java int. We can create one and set its value using the set() method:




```
IntWritable writable = new IntWritable();
writable.set(163);
```

- To examine the serialized form of the IntWritable, we write a small helper method that wraps a java.io.ByteArrayOutputStream in a java.io.DataOutputStream to capture the bytes in the serialized stream

```
ByteArrayOutputStream out = new ByteArrayOutputStream();
DataOutputStreamdataOut = new DataOutputStream(out);
writable.write(dataOut);
dataOut.close();
returnout.toByteArray();
```



Java primitive	Writable implementation	Serialized size (bytes)
boolean	BooleanWritable	1
byte	ByteWritable	1
int	IntWritable	4
	VIntWritable	1–5
long	LongWritable	8
	VLongWritable	1–9

5.2.2 Writable Classes

- Hadoop comes with a large selection of Writable classes in the org.apache.hadoop.io package. They form the class hierarchy shown in Figure 4-1.

Writable Class

- Writable wrappers for Java primitives
- There are Writable wrappers for all the Java primitive types except short and char.
- All have a get() and a set() method for retrieving and storing the wrapped value.

Text

- Text is a Writable for UTF-8 sequences. It can be thought of as the Writable equivalent of java.lang.String.
- The Text class uses an int to store the number of bytes in the string encoding, so the maximum value is 2 GB. Furthermore, Text uses standard UTF -8, which makes it potentially easier to interpoperate with other tools that understand UTF-8.
- The Text class has several features.

```
assertThat(t.charAt(2), is((int) 'z'));
assertThat("Out of bounds", t.charAt(100), is(-1));
```

Table 4-7. Unicode characters

Unicode code point	U+0041	U+00DF	U+6771	U+10400
Name	LATIN CAPITAL LETTER A	LATIN SMALL LETTER SHARP S	N/A (a unified Han ideograph)	DESERET CAPITAL LETTER LONG I
UTF-8 code units	41	c3 9f	e6 9d b1	f0 90 90 80
Java representation	\u0041	\u00DF	\u6771	\uD801\uDC00

- Indexing
- Unicode
- Iteration
- Mutability
- Resorting to String

Indexing

- Indexing for the Text class is in terms of position in the encoded byte sequence, not the Unicode character in the string, or the Java char code unit. For ASCII String, these three concepts of index position coincide.

- Notice that charAt() returns an intrepresenting a Unicode code point, unlike the String variant that returns a char. Text also has a find() method, which is analogous to String's indexOf()

Unicode

- When we start using characters that are encoded with more than a single byte, the differences between Text and String become clear. Consider the Unicode characters shown in Table 4-7 All but the last character in the table, U+10400, can be expressed using a single Java

char.

Iteration

Iterating over the Unicode characters in Text is complicated by the use of byte offsets for indexing, since you can't just increment the index.

- The idiom for iteration is a little obscure: turn the Text object into a java.nio.ByteBuffer. Then repeatedly call the bytesToCodePoint() static method on Text with the buffer. This method extracts the next code point as an intand updates the position in the buffer.

For Example...

```
public class TextIterator
{
        public static void main(String[] args)
        {
                Text t = new Text("\u0041\u00DF\u6771\uD801\uDC00");
                ByteBufferbuf = ByteBuffer.wrap(t.getBytes(), 0, t.getLength());
                Iteration. 102 | Chapter 4: Hadoop I/O intcp;
                while (buf.hasRemaining() && (cp = Text.bytesToCodePoint(buf)) != -1)
                {
                        System.out.println(Integer.toHexString(cp));
                }
        }
}
```

Mutability

Another difference with String is that Text is mutable. You can reuse a Text instance by calling on of the set() methods on it.
For Example...

```
Text t = new Text("hadoop");
t.set("pig");
assertThat(t.getLength(), is(3));
assertThat(t.getBytes().length, is(3));
```

Restoring to String

- Text doesn't have as rich an API for manipulating strings as java.lang.String , so in many cases you need to convert the Text object to a String.

Null Writable

- NullWritable is a special type of Writable, as it has a zero -length serialization. No bytes are written to , or read from , the stream. It is used as a placeholder.

- For example, in MapReduce , a key or a value can be declared as a NullWritable when you don't need to use that position-it effectively stores a constant empty value.

- NullWritable can also be useful as a key in SequenceFile when you want to store a list of values, as opposed to key-value pairs. It is an immutable singleton: the instance can be retrieved by calling NullWritable.get().

5.2.4 Serialization Frameworks

- Although most Map Reduce programs use Writable key and value types, this isn't mandated by the Map Reduce API. In fact, any types can be used, the only requirement is that there be a mechanism that translates to and from a binary representation of each type.
- To support this, Hadoophas an API for pluggable serialization frameworks. A serialization framework is represented by an implementation of Serialization. WritableSerialization,for example, is the implementation of Serialization for Writable types.
- Although making it convenient to be able to use standard Java types in Map Reduce programs, like Integer or String, Java Object Serialization is not as efficient as Writable, so it's not worth making this trade-off.

5.4 File-Based data structure

- For some applications, you need a specialized data structure to hold your data. For MapReduce-based processing, putting each blob of binary data into its own file doesn't scale, so Hadoopdeveloped a number of higher-level containers for these situations.
- Higher-level containers
  - SequenceFile
  - MapFile
  -

5.4.1 SequenceFile

- Imagine a logfile, where each log record is a new line of text. If you want to logbinary types, plain text isn't a suitable format.
- Hadoop'sSequenceFileclass fits the bill in this situation, providing a persistent data structure for binary key-value pairs.To use it as a logfileformat, you would choose a key, such as timestamp represented by a LongWritable, and the value is Writable that represents the quantity being logged.
- SequenceFilealso work well as containers for smaller files. HDFS and Map Reduce are optimized for large files, so packing files into a SequenceFilemakes storing and processing the smaller files more efficient.

Writing a SequenceFile

- To create a SequenceFile, use one of its createWriter() static methods, which returns a SequenceFile.Writerinstance.

- The keys and values stored in a SequenceFiledo not necessarily need to be Writable. Any types that can be serialized and deserializedby a Serialization may be used.
- Once you have a SequenceFile.Writer, you then write key-value pairs, using the append() method. Then when you've finished you call the close() method (SequenceFile.Writerimplements java.io.Closeable)

For example...

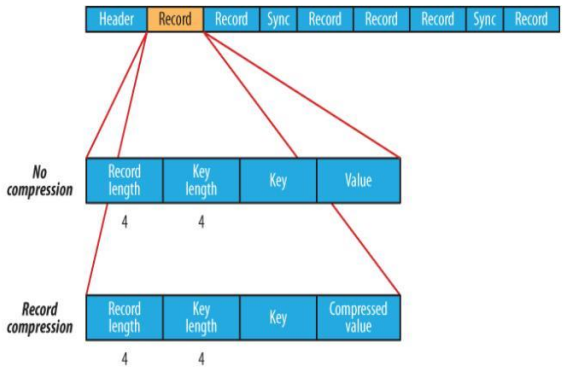
```
IntWritable key = new IntWritable();
Text value = new Text();
SequenceFile.Writer writer = null;
try { writer = SequenceFile.createWriter(fs, conf, path, key.getClass(), value.getClass());
for (int i = 0; i < 100; i++) { key.set(100 - i); value.set(DATA[i % DATA.length]);
System.out.printf("[%s]\t%s\t%s\n", writer.getLength(), key, value);
writer.append(key, value); } } finally { IOUtils.closeStream(writer);
```

Reading a SequenceFile

Reading sequence files from beginning to end is a matter of creating an instance of SequenceFile.Reader, and iterating over records by repeatedlyinvoking one of the next() methods.

If you are using Writable types, you can use the next() method that takes a key and a value argument, and reads the next key and value in the stream into these variables:

```
For example...  public static void main(String[] args) throws IOException
{
        String uri = args[0];
        Configuration conf = new Configuration();
        FileSystemfs = FileSystem.get(URI.create(uri), conf);
        Path path = new Path(uri);

        SequenceFile.Reader reader = null;
        try
        {
                reader = new SequenceFile.Reader(fs, path, conf);
                Writable key = (Writable) ReflectionUtils.newInstance(reader.getKeyClass(), conf);
                Writable value = (Writable) ReflectionUtils.newInstance(reader.getValueClass(), conf);
                long position = reader.getPosition();
                while (reader.next(key, value)) { String syncSeen = reader.syncSeen() ? "*" : "";
        System.out.printf("[%s%s]\t%s\t%s\n", position, syncSeen, key, value);
                position = reader.getPosition(); // beginning of next record
        }
}
Finally
 {
        IOUtils.closeStream(reader);
}
```



5.4.2 MapFile

- A MapFile is a sorted SequenceFile with an index to permit lookups by key. MapFile can be thought of as a persistent form of java.util.Map(although it doesn't implement this interface), which is able to grow beyond the size of a Map that is kept in memory

Writing a MapFile

- Writing a MapFile is similar to writing a Sequence File. You create an instance of MapFile. Writer, then call the append () method to add entries in order. Keys must be instances of WritableComparable, and values must be Writable

For example:

```
String uri = args[0];
Configuration conf = new Configuration();
FileSystemfs = FileSystem.get(URI.create(uri), conf);
IntWritable key = new IntWritable();
Text value = new Text();
MapFile.Writer writer = null;
try
 {
        writer = new MapFile.Writer(conf, fs, uri, key.getClass(), value.getClass());
        for (int i = 0; i < 1024; i++) { key.set(i + 1);
        value.set(DATA[i % DATA.length]);
        writer.append(key, value);
 }
Finally
{
        IOUtils.closeStream(writer);
}
```

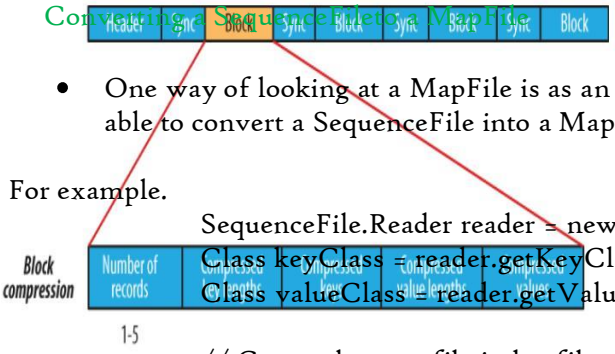
Reading a MapFile

- Iterating through the entries in order in a MapFileis similar to the procedure for a SequenceFile. You create a MapFile. Reader, then call the next() method until it returns false, signifying that no entry was read because the end of the file was reached.

```
publicboolean next(WritableComparable key, Writable val) throws IOException
public Writable get(WritableComparable key, Writable val) throws IOException
```

- The return value is used to determine if an entry was found in the MapFile. If it's null, then no value exist for the given key. If key was found, then the value for that key is read into val, as well as being returned from the method call.
    - For this operation, the MapFile. Readerreads the index file into memory. A very large MapFile's index can take up a lot of memory. Rather than reindex to change the index interval, it is possible to lad only a fraction of the index keys into memory when reading the MapFile by setting the io.amp.index.ksipproperty.

Converting a SequenceFile to a MapFile



- One way of looking at a MapFile is as an indexed and sorted SequenceFile. So it's quite natural to want to be able to convert a SequenceFile into a MapFile.

For example.

```
SequenceFile.Reader reader = new SequenceFile.Reader(fs, mapData, conf);
Class keyClass = reader.getKeyClass();
Class valueClass = reader.getValueClass(); reader.close();

// Create the map file index file

long entries = MapFile.fix(fs, map, keyClass, valueClass, false, conf);
System.out.printf("Created MapFile %s with %d entries\n", map, entries);
```

Reading Data from a Hadoop URL

One of the simplest ways to read a file from a Hadoop filesystem is by using a java.net.URL object to open a stream to read the data from. The general idiom is:

```
InputStream in = null;
try {
   in = new URL("hdfs://host/path").openStream();
   // process in
} finally {
   IOUtils.closeStream(in);
}
```

There's a little bit more work required to make Java recognize Hadoop's hdfs URL scheme. This is achieved by calling the setURLStreamHandlerFactory() method on URL with an instance of FsUrlStreamHandlerFactory. This method can be called only once per JVM, so it is typically executed in a static block. This limitation means that if some other part of your program — perhaps a third-party component outside your control — sets a

URLStreamHandlerFactory, you won't be able to use this approach for reading data from Hadoop. The next section discusses an alternative.

Example 3-1 shows a program for displaying files from Hadoop filesystems on standard output, like the Unix cat command.

Example 3-1. Displaying files from a Hadoop filesystem on standard output using a URLStreamHandler

```
public class URLCat {
   static {
      URL.setURLStreamHandlerFactory(new FsUrlStreamHandlerFactory());
   }
   public static void main(String[] args) throws Exception {
      InputStream in = null;
      try {
         in = new URL(args[0]).openStream();
         IOUtils.copyBytes(in, System.out, 4096, false);
      }
 finally {
         IOUtils.closeStream(in);  }  }  }
```

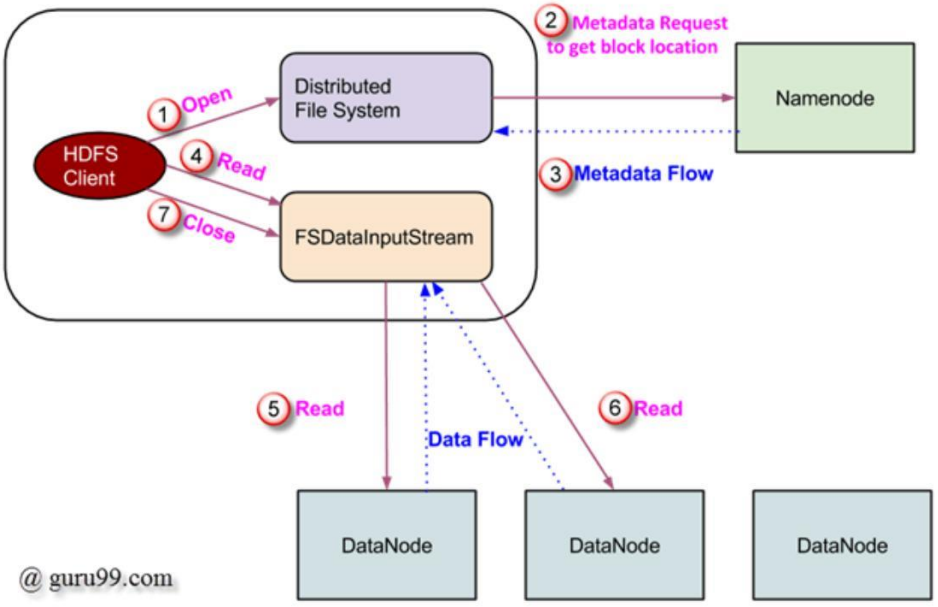
We make use of the handy IOUtils class that comes with Hadoop for closing the stream in the finally clause, and also for copying bytes between the input stream and the output stream (System.out, in this case). The last two arguments to the copyBytes() method are the buffer size used for copying and whether to close the streams when the copy is complete. We close the input stream ourselves, and System.out doesn't need to be closed.

Here's a sample run:[31]

```
% export HADOOP_CLASSPATH=hadoop-examples.jar
% hadoop URLCat hdfs://localhost/user/tom/quangle.txt
On the top of the Crumpetty Tree
The Quangle Wangle sat,
But his face you could not see,
On account of his Beaver Hat.
```

Read Operation In HDFS

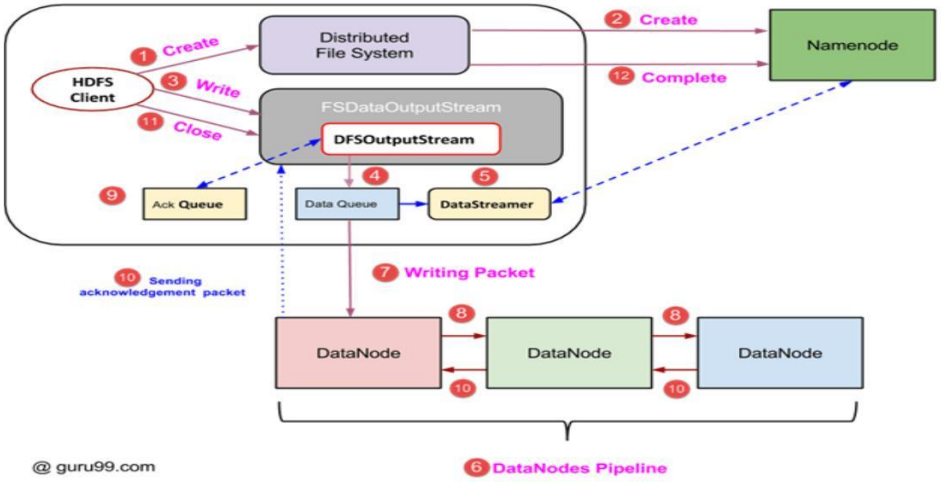
Data read request is served by HDFS, NameNode, and DataNode. Let's call the reader as a 'client'. Below diagram depicts file read operation in Hadoop.



1. A client initiates read request by calling 'open()' method of FileSystem object; it is an object of type DistributedFileSystem.
2. This object connects to namenode using RPC and gets metadata information such as the locations of the blocks of the file. Please note that these addresses are of first few blocks of a file.
3. In response to this metadata request, addresses of the DataNodes having a copy of that block is returned back.
4. Once addresses of DataNodes are received, an object of type FSDataInputStream is returned to the client. FSDataInputStream contains DFSInputStream which takes care of interactions with DataNode and NameNode. In step 4 shown in the above diagram, a client invokes 'read()' method which causes DFSInputStream to establish a connection with the first DataNode with the first block of a file.
5. Data is read in the form of streams wherein client invokes 'read()' method repeatedly. This process of read() operation continues till it reaches the end of block.
6. Once the end of a block is reached, DFSInputStream closes the connection and moves on to locate the next DataNode for the next block
7. Once a client has done with the reading, it calls a close() method.

Write Operation In HDFS

In this,we know how data is written into HDFS through files.



1. A client initiates write operation by calling 'create()' method of DistributedFileSystem object which creates a new file - Step no. 1 in the above diagram.
2. DistributedFileSystem object connects to the NameNode using RPC call and initiates new file creation. However, this file creates operation does not associate any blocks with the file. It is the responsibility of NameNode to verify that the file (which is being created) does not exist already and a client has correct permissions to create a new file. If a file already exists or client does not have sufficient permission to create a new file, then IOException is thrown to the client. Otherwise, the operation succeeds and a new record for the file is created by the NameNode.
3. Once a new record in NameNode is created, an object of type FSDataOutputStream is returned to the client. A client uses it to write data into the HDFS. Data write method is invoked (step 3 in the diagram).
4. FSDataOutputStream contains DFSOutputStream object which looks after communication with DataNodes and NameNode. While the client continues writing data, DFSOutputStream continues creating packets with this data. These packets are enqueued into a queue which is called as DataQueue.
5. There is one more component called DataStreamer which consumes this DataQueue. DataStreamer also asks NameNode for allocation of new blocks thereby picking desirable DataNodes to be used for replication.
6. Now, the process of replication starts by creating a pipeline using DataNodes. In our case, we have chosen a replication level of 3 and hence there are 3 DataNodes in the pipeline.
7. The DataStreamer pours packets into the first DataNode in the pipeline.
8. Every DataNode in a pipeline stores packet received by it and forwards the same to the second DataNode in a pipeline.
9. Another queue, 'Ack Queue' is maintained by DFSOutputStream to store packets which are waiting for acknowledgment from DataNodes.
10. Once acknowledgment for a packet in the queue is received from all DataNodes in the pipeline, it is removed from the 'Ack Queue'. In the event of any DataNode failure, packets from this queue are used to reinitiate the operation.
11. After a client is done with the writing data, it calls a close() method (Step 9 in the diagram) Call to close(), results into flushing remaining data packets to the pipeline followed by waiting for acknowledgment.
12. Once a final acknowledgment is received, NameNode is contacted to tell it that the file write operation is complete.

Access HDFS using JAVA API

In order to interact with Hadoop's filesystem programmatically, Hadoop provides multiple JAVA classes. Package named org.apache.hadoop.fs contains classes useful in manipulation of a file in Hadoop's filesystem. These operations include, open, read, write, and close. Actually, file API for Hadoop is generic and can be extended to interact with other filesystems other than HDFS.

Reading a file from HDFS, programmatically

Object java.net.URL is used for reading contents of a file. To begin with, we need to make Java recognize Hadoop's hdfs URL scheme. This is done by calling setURLStreamHandlerFactory method on URL object and an instance of FsUrlStreamHandlerFactory is passed to it. This method needs to be executed only once per JVM, hence it is enclosed in a static block.

An example code is-

```
public class URLCat {
  static {
    URL.setURLStreamHandlerFactory(new FsUrlStreamHandlerFactory());
  }
  public static void main(String[] args) throws Exception {
    InputStream in = null;
    try {
      in = new URL(args[0]).openStream();
      IOUtils.copyBytes(in, System.out, 4096, false);
    } finally {
      IOUtils.closeStream(in);
    }
  }
}
```

This code opens and reads contents of a file. Path of this file on HDFS is passed to the program as a command line argument.

Access HDFS Using COMMAND-LINE INTERFACE

This is one of the simplest ways to interact with HDFS. Command-line interface has support for filesystem operations like read the file, create directories, moving files, deleting data, and listing directories.

We can run '$HADOOP_HOME/bin/hdfs dfs -help' to get detailed help on every command. Here, 'dfs' is a shell command of HDFS which supports multiple subcommands.

Some of the widely used commands are listed below along with some details of each one.

1. Copy a file from the local filesystem to HDFS

$HADOOP_HOME/bin/hdfs dfs -copyFromLocal temp.txt /

This command copies file temp.txt from the local filesystem to HDFS.

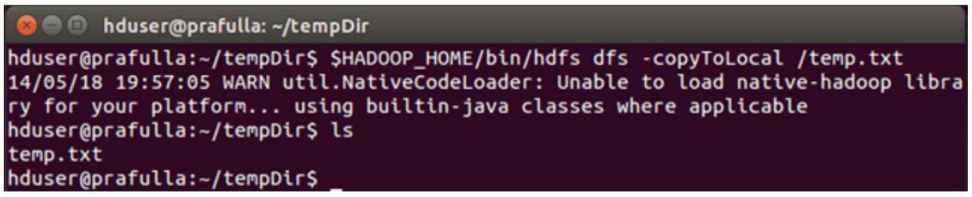
2. We can list files present in a directory using -ls

$HADOOP_HOME/bin/hdfs dfs -ls /

We can see a file 'temp.txt' (copied earlier) being listed under ' / ' directory.

3. Command to copy a file to the local filesystem from HDFS

$HADOOP_HOME/bin/hdfs dfs -copyToLocal /temp.txt



We can see temp.txt copied to a local filesystem.

4. Command to create a new directory

$HADOOP_HOME/bin/hdfs dfs -mkdir /mydirectory

Starting HDFS

Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.

$ hadoop namenode -format

After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.

$ start-dfs.sh

Listing Files in HDFS

After loading the information in the server, we can find the list of files in a directory, status of a file, using 'ls'. Given below is the syntax of ls that you can pass to a directory or a filename as an argument.

$ $HADOOP_HOME/bin/hadoop fs -ls <args>

Inserting Data into HDFS

Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system. Follow the steps given below to insert the required file in the Hadoop file system.

Step 1

You have to create an input directory.

```
$ $HADOOP_HOME/bin/hadoop fs -mkdir /user/input
```

Step 2

Transfer and store a data file from local systems to the Hadoop file system using the put command.

```
$ $HADOOP_HOME/bin/hadoop fs -put /home/file.txt /user/input
```

Step 3

You can verify the file using ls command.

```
$ $HADOOP_HOME/bin/hadoop fs -ls /user/input
```

Retrieving Data from HDFS

Assume we have a file in HDFS called outfile. Given below is a simple demonstration for retrieving the required file from the Hadoop file system.

Step 1

Initially, view the data from HDFS using cat command.

```
$ $HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile
```

Step 2

Get the file from HDFS to the local file system using get command.

```
$ $HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/
```

Shutting Down the HDFS

You can shut down the HDFS by using the following command.

```
$ stop-dfs.sh
```

1. Create a directory in HDFS at given path(s).

Usage:

```
hadoop fs -mkdir <paths>
```

Example:

```
hadoop fs -mkdir /user/saurzcode/dir1 /user/saurzcode/dir2
```

2. List the contents of a directory.

Usage :
hadoop fs -ls <args>
Example:
hadoop fs -ls /user/saurzcode

3. Upload and download a file in HDFS.

Upload:

hadoop fs -put:

Copy single src file, or multiple src files from local file system to the Hadoop data file system

Usage:
hadoop fs -put <localsrc> ... <HDFS_dest_Path>
Example:
hadoop fs -put /home/saurzcode/Samplefile.txt /user/saurzcode/dir3/
Download:
hadoop fs -get:

Copies/Downloads files to the local file system

Usage:
hadoop fs -get <hdfs_src> <localdst>
Example:
hadoop fs -get /user/saurzcode/dir3/Samplefile.txt /home/

4. See contents of a file

Same as unix cat command:

Usage:
hadoop fs -cat <path[filename]>
Example:
hadoop fs -cat /user/saurzcode/dir1/abc.txt

5. Copy a file from source to destination

This command allows multiple sources as well in which case the destination must be a directory.

Usage:
hadoop fs -cp <source> <dest>
Example:
hadoop fs -cp /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

6. Copy a file from/To Local file system to HDFS

copyFromLocal

Usage:
hadoop fs -copyFromLocal <localsrc> URI

Example:
hadoop fs -copyFromLocal /home/saurzcode/abc.txt /user/saurzcode/abc.txt

Similar to put command, except that the source is restricted to a local file reference.

copyToLocal

Usage:
hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>

Similar to get command, except that the destination is restricted to a local file reference.

7. Move file from source to destination.

Note:- Moving files across filesystem is not permitted.

Usage :
hadoop fs -mv <src> <dest>
Example:
hadoop fs -mv /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

8. Remove a file or directory in HDFS.

Remove files specified as argument. Deletes directory only when it is empty

Usage :
hadoop fs -rm <arg>
Example:
hadoop fs -rm /user/saurzcode/dir1/abc.txt
Recursive version of delete.
Usage :
hadoop fs -rmr <arg>
Example:
hadoop fs -rmr /user/saurzcode/

9. Display last few lines of a file.

Similar to tail command in Unix.

Usage :
hadoop fs -tail <path[filename]>
Example:
hadoop fs -tail /user/saurzcode/dir1/abc.txt

10. Display the aggregate length of a file.

Usage :
hadoop fs -du <path>
Example:
hadoop fs -du /user/saurzcode/dir1/abc.txt