# CLOUD COMPUTING

**A SEMINAR REPORT**

***Submitted by***

**MAHESWARAN.M**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING**

**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY,**

**COCHIN – 682022**

**NOV 2008**

# DIVISION OF COMPUTER ENGINEERING,

# SCHOOL OF ENGINEERING,

# COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY, COCHIN – 682022

## Bonafide Certificate

Certified that this seminar report titled "**Cloud Computing**" is the bonafide work done by **Maheswaran.M** who carried out the work under my supervision.


**Preetha S**
SEMINAR GUIDE
Lecturer,
Division of Computer Science
SOE, CUSAT

**Dr. David Peter S**
Head of the Department
Division of Computer Science
SOE, CUSAT

# Acknowledgement

I am thankful to my seminar guide Mrs. Preetha S, CUSAT for her proper guidance and valuable suggestions. I am also greatly thankful to Mr. David Peter, the head of the Division of Computer Science and Engineering and other faculty members for giving me an opportunity to learn and do this seminar. If not for the above mentioned people, my seminar would never have been completed in such a successfully manner. I once again extend my sincere thanks to all of them.

**Maheswaran.M**

# Table of Contents

# List of figures

# Abstract

Computers have become an indispensable part of life. We need computers everywhere, be it for work, research or in any such field. As the use of computers in our day-to-day life increases, the computing resources that we need also go up. For companies like Google and Microsoft, harnessing the resources as and when they need it is not a problem. But when it comes to smaller enterprises, affordability becomes a huge factor. With the huge infrastructure come problems like machines failure, hard drive crashes, software bugs, etc. This might be a big headache for such a community. Cloud Computing offers a solution to this situation.

Cloud computing is a paradigm shift in which computing is moved away from personal computers and even the individual enterprise application server to a ‘cloud’ of computers. A cloud is a virtualized server pool which can provide the different computing resources of their clients. Users of this system need only be concerned with the computing service being asked for. The underlying details of how it is achieved are hidden from the user. The data and the services provided reside in massively scalable data centers and can be ubiquitously accessed from any connected device all over the world.

Cloud computing is the style of computing where massively scaled IT related capabilities are provided as a service across the internet to multiple external customers and are billed by consumption. Many cloud computing providers have popped up and there is a considerable growth in the usage of this service. Google, Microsoft, Yahoo, IBM and Amazon have started providing cloud computing services. Amazon is the pioneer in this field. Smaller companies like SmugMug, which is an online photo hosting site, has used cloud services for the storing all the data and doing some of its services.

Cloud Computing is finding use in various areas like web hosting, parallel batch processing, graphics rendering, financial modeling, web crawling, genomics analysis, etc.

# 1. Introduction

The Greek myths tell of creatures plucked from the surface of the Earth and enshrined as constellations in the night sky. Something similar is happening today in the world of computing. Data and programs are being swept up from desktop PCs and corporate server rooms and installed in "the compute cloud". In general, there is a shift in the geography of computation.

What is cloud computing exactly? As a beginning here is a definition

> "An emerging computer paradigm where data and services reside in massively scalable data centers in the cloud and can be accessed from any connected devices over the internet"

Like other definitions of topics like these, an understanding of the term cloud computing requires an understanding of various other terms which are closely related to this. While there is a lack of precise scientific definitions for many of these terms, general definitions can be given.

Cloud computing is an emerging paradigm in the computer industry where the computing is moved to a cloud of computers. It has become one of the buzz words of the industry. The core concept of cloud computing is, quite simply, that the vast computing resources that we need will reside somewhere out there in the cloud of computers and we'll connect to them and use them as and when needed.

Computing can be described as any activity of using and/or developing computer hardware and software. It includes everything that sits in the bottom layer, i.e. everything from raw compute power to storage capabilities. Cloud computing ties together all these entities and delivers them as a single integrated entity under its own sophisticated management.

Cloud is a term used as a metaphor for the wide area networks (like internet) or any such large networked environment. It came partly from the cloud-like symbol used to represent the complexities of the networks in the schematic diagrams. It represents all the complexities of the network which may include everything from cables, routers, servers, data centers and all such other devices.

Computing started off with the mainframe era. There were big mainframes and everyone connected to them via “dumb” terminals. This old model of business computing was frustrating for the people sitting at the dumb terminals because they could do only what they were “authorized” to do. They were dependent on the computer administrators to give them permission or to fix their problems. They had no way of staying up to the latest innovations.

The personal computer was a rebellion against the tyranny of centralized computing operations. There was a kind of freedom in the use of personal computers. But this was later replaced by server architectures with enterprise servers and others showing up in the industry. This made sure that the computing was done and it did not eat up any of the resources that one had with him. All the computing was performed at servers. Internet grew in the lap of these servers. With cloud computing we have come a full circle. We come back to the centralized computing infrastructure. But this time it is something which can easily be accessed via the internet and something over which we have all the control.

# 2. Cloud Computing

A definition for cloud computing can be given as an emerging computer paradigm where data and services reside in massively scalable data centers in the cloud and can be accessed from any connected devices over the internet.

Cloud computing is a way of providing various services on virtual machines allocated on top of a large physical machine pool which resides in the cloud. Cloud computing comes into focus only when we think about what IT has always wanted - a way to increase capacity or add different capabilities to the current setting on the fly without investing in new infrastructure, training new personnel or licensing new software. Here 'on the fly' and 'without investing or training' becomes the keywords in the current situation. But cloud computing offers a better solution.

We have lots of compute power and storage capabilities residing in the distributed environment of the cloud. What cloud computing does is to harness the capabilities of these resources and make available these resources as a single entity which can be changed to meet the current needs of the user. The basis of cloud computing is to create a set of virtual servers on the available vast resource pool and give it to the clients. Any web enabled device can be used to access the resources through the virtual servers. Based on the computing needs of the client, the infrastructure allotted to the client can be scaled up or down.

From a business point of view, cloud computing is a method to address the scalability and availability concerns for large scale applications which involves lesser overhead. Since the resource allocated to the client can be varied based on the needs of the client and can be done without any fuss, the overhead is very low.

One of the key concepts of cloud computing is that processing of 1000 times the data need not be 1000 times harder. As and when the amount of data increases, the cloud computing services can be used to manage the load effectively and make the processing tasks easier. In the era of enterprise servers and personal computers, hardware was the commodity as the main criteria for the processing capabilities depended on the hardware configuration of the server. But with the advent of cloud computing, the commodity has changed to cycles and bytes - i.e. in cloud computing services, the users are charged based on the number of cycles of execution performed

or the number of bytes transferred. The hardware or the machines on which the applications run are hidden from the user. The amount of hardware needed for computing is taken care of by the management and the client is charged based on how the application uses these resources.

## 2.1. Characteristics of Cloud Computing

1. Self Healing

   Any application or any service running in a cloud computing environment has the property of self healing. In case of failure of the application, there is always a hot backup of the application ready to take over without disruption. There are multiple copies of the same application - each copy updating itself regularly so that at times of failure there is at least one copy of the application which can take over without even the slightest change in its running state.

2. Multi-tenancy

   With cloud computing, any application supports multi-tenancy - that is multiple tenants at the same instant of time. The system allows several customers to share the infrastructure allotted to them without any of them being aware of the sharing. This is done by virtualizing the servers on the available machine pool and then allotting the servers to multiple users. This is done in such a way that the privacy of the users or the security of their data is not compromised.

3. Linearly Scalable

   Cloud computing services are linearly scalable. The system is able to break down the workloads into pieces and service it across the infrastructure. An exact idea of linear scalability can be obtained from the fact that if one server is able to process say 1000 transactions per second, then two servers can process 2000 transactions per second.

4. Service-oriented

   Cloud computing systems are all service oriented - i.e. the systems are such that they are created out of other discrete services. Many such

discrete services which are independent of each other are combined together to form this service. This allows re-use of the different services that are available and that are being created. Using the services that were just created, other such services can be created.

5. SLA Driven

   Usually businesses have agreements on the amount of services. Scalability and availability issues cause clients to break these agreements. But cloud computing services are SLA driven such that when the system experiences peaks of load, it will automatically adjust itself so as to comply with the service-level agreements.

   The services will create additional instances of the applications on more servers so that the load can be easily managed.

6. Virtualized

   The applications in cloud computing are fully decoupled from the underlying hardware. The cloud computing environment is a fully virtualized environment.

7. Flexible

   Another feature of the cloud computing services is that they are flexible. They can be used to serve a large variety of workload types - varying from small loads of a small consumer application to very heavy loads of a commercial application.

# 3. Need for Cloud Computing

What could we do with 1000 times more data and CPU power? One simple question. That's all it took the interviewers to bewilder the confident job applicants at Google. This is a question of relevance because the amount of data that an application handles is increasing day by day and so is the CPU power that one can harness.

There are many answers to this question. With this much CPU power, we could scale our businesses to 1000 times more users. Right now we are gathering statistics about every user using an application. With such CPU power at hand, we could monitor every single user click and every user interaction such that we can gather all the statistics about the user. We could improve the recommendation systems of users. We could model better price plan choices. With this CPU power we could simulate the case where we have say 1,00,000 users in the system without any glitches.

There are lots of other things we could do with so much CPU power and data capabilities. But what is keeping us back. One of the reasons is the large scale architecture which comes with these are difficult to manage. There may be many different problems with the architecture we have to support. The machines may start failing, the hard drives may crash, the network may go down and many other such hardware problems. The hardware has to be designed such that the architecture is reliable and scalable. This large scale architecture has a very expensive upfront and has high maintenance costs. It requires different resources like machines, power, cooling, etc. The system also cannot scale as and when needed and so is not easily reconfigurable.

The resources are also constrained by the resources. As the applications become large, they become I/O bound. The hard drive access speed becomes a limiting factor. Though the raw CPU power available may not be a factor, the amount of RAM available clearly becomes a factor. This is also limited in this context. If at all the hardware problems are managed very well, there arises the software problems. There may be bugs in the software using this much of data. The workload also demands two important tasks for two completely different people. The software has to

be such that it is bug free and has good data processing algorithms to manage all the data.

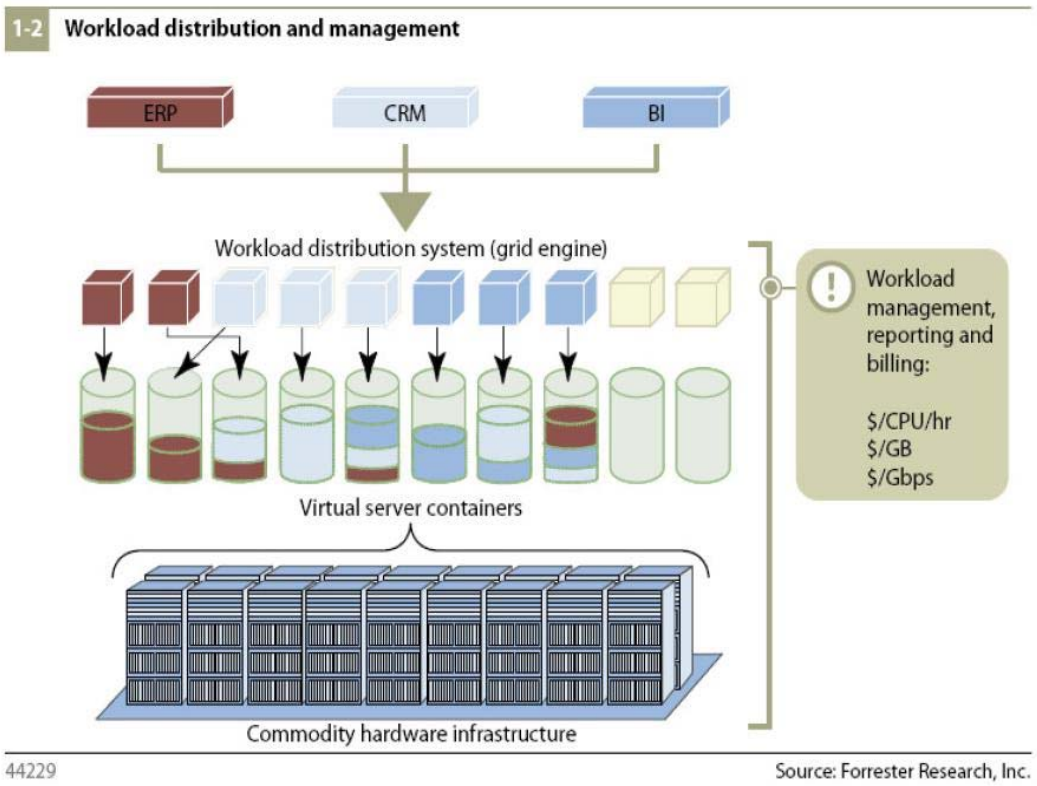
The cloud computing works on the cloud - so there are large groups of often low-cost servers with specialized connections to spread the data-processing chores among them. Since there are a lot of low-cost servers connected together, there are large pools of resources available. So these offer almost unlimited computing resources. This makes the availability of resources a lesser issue.

The data of the application can also be stored in the cloud. Storage of data in the cloud has many distinct advantages over other storages. One thing is that data is spread evenly through the cloud in such a way that there are multiple copies of the data and there are ways by which failure can be detected and the data can be rebalanced on the fly. The I/O operations become simpler in the cloud such that browsing and searching for something in 25GB or more of data becomes simpler in the cloud, which is nearly impossible to do on a desktop.

The cloud computing applications also provide automatic reconfiguration of the resources based on the service level agreements. When we are using applications out of the cloud, to scale the application with respect to the load is a mundane task because the resources have to be gathered and then provided to the users. If the load on the application is such that it is present only for a small amount of time as compared to the time its working out of the load, but occurs frequently, then scaling of the resources becomes tedious. But when the application is in the cloud, the load can be managed by spreading it to other available nodes by making a copy of the application on to them. This can be reverted once the load goes down. It can be done as and when needed. All these are done automatically such that the resources maintain and manage themselves

# 4. Enabling Technologies

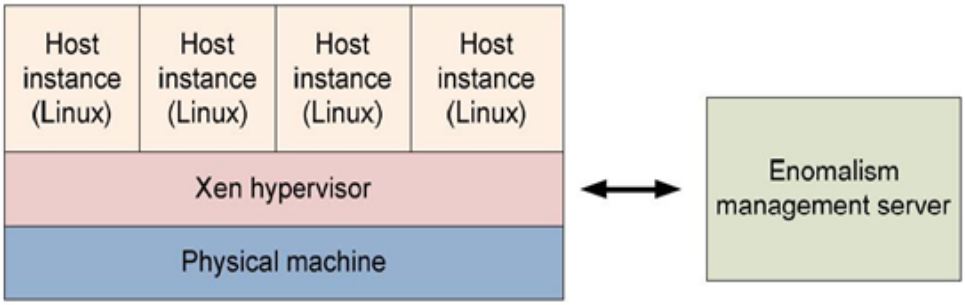
## 4.1. Cloud Computing Application Architecture




This gives the basic architecture of a cloud computing application. We know that cloud computing is the shift of computing to a host of hardware infrastructure that is distributed in the cloud. The commodity hardware infrastructure consists of the various low cost data servers that are connected to the system and provide their storage and processing and other computing resources to the application. Cloud computing involves running applications on virtual servers that are allocated on this distributed hardware infrastructure available in the cloud. These virtual servers are made in such a way that the different service level agreements and reliability issues are met. There may be multiple instances of the same virtual server accessing the different parts of the hardware infrastructure available. This is to make sure that there are multiple copies of the applications which are ready to take over on another one's failure. The virtual server distributes the processing between the infrastructure and the computing is done and the result returned. There will be a workload distribution

management system, also known as the grid engine, for managing the different requests coming to the virtual servers. This engine will take care of the creation of multiple copies and also the preservation of integrity of the data that is stored in the infrastructure. This will also adjust itself such that even on heavier load, the processing is completed as per the requirements. The different workload management systems are hidden from the users. For the user, the processing is done and the result is obtained. There is no question of where it was done and how it was done. The users are billed based on the usage of the system - as said before - the commodity is now cycles and bytes. The billing is usually on the basis of usage per CPU per hour or GB data transfer per hour.

## 4.2.Server Architecture






Cloud computing makes use of a large physical resource pool in the cloud. As said above, cloud computing services and applications make use of virtual server instances built upon this resource pool. There are two applications which help in managing the server instances, the resources and also the management of the resources by these virtual server instances. One of these is the Xen hypervisor which provides an abstraction layer between the hardware and the virtual OS so that the distribution of the resources and the processing is well managed. Another application

that is widely used is the Enomalism server management system which is used for management of the infrastructure platform.

When Xen is used for virtualization of the servers over the infrastructure, a thin software layer known as the Xen hypervisor is inserted between the server's hardware and the operating system. This provides an abstraction layer that allows each physical server to run one or more "virtual servers," effectively decoupling the operating system and its applications from the underlying physical server. The Xen hypervisor is a unique open source technology, developed collaboratively by the Xen community and engineers at over 20 of the most innovative data center solution vendors, including AMD, Cisco, Dell, HP, IBM, Intel, Mellanox, Network Appliance, Novell, Red Hat, SGI, Sun, Unisys, Veritas, Voltaire, and Citrix. Xen is licensed under the GNU General Public License (GPL2) and is available at no charge in both source and object format. The Xen hypervisor is also exceptionally lean-- less than 50,000 lines of code. That translates to extremely low overhead and near-native performance for guests. Xen re-uses existing device drivers (both closed and open source) from Linux, making device management easy. Moreover Xen is robust to device driver failure and protects both guests and the hypervisor from faulty or malicious drivers

The Enomalism virtualized server management system is a complete virtual server infrastructure platform. Enomalism helps in an effective management of the resources. Enomalism can be used to tap into the cloud just as you would into a remote server. It brings together all the features such as deployment planning, load balancing, resource monitoring, etc. Enomalism is an open source application. It has a very simple and easy to use web based user interface. It has a module architecture which allows for the creation of additional system add-ons and plugins. It supports one click deployment of distributed or replicated applications on a global basis. It supports the management of various virtual environments including KVM/Qemu, Amazon EC2 and Xen, OpenVZ, Linux Containers, VirtualBox. It has fine grained user permissions and access privileges.

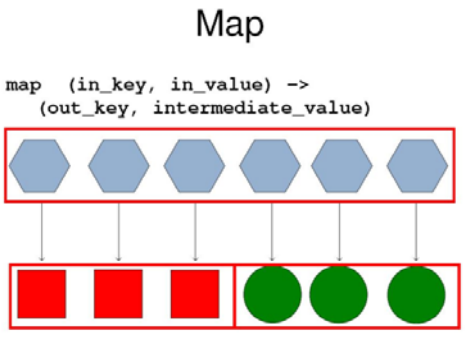
## 4.3.Map Reduce

Map Reduce is a software framework developed at Google in 2003 to support parallel computations over large (multiple petabyte) data sets on clusters of commodity computers. This framework is largely taken from ‘map’ and ‘reduce’ functions commonly used in functional programming, although the actual semantics of the framework are not the same. It is a programming model and an associated implementation for processing and generating large data sets. Many of the real world tasks are expressible in this model. MapReduce implementations have been written in C++, Java and other languages.

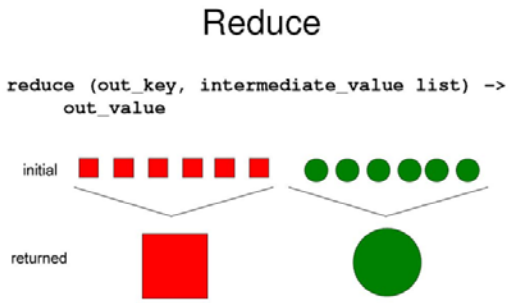
Programs written in this functional style are automatically parallelized and executed on the cloud. The run-time system takes care of the details of partitioning the input data, scheduling the program’s execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a largely distributed system.

The computation takes a set of *input* key/value pairs, and produces a set of *output* key/value pairs. The user of the MapReduce library expresses the computation as two functions: *Map* and *Reduce*.

*Map*, written by the user, takes an input pair and produces a set of *intermediate* key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the *Reduce* function.



The *Reduce* function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per *Reduce* invocation. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.




MapReduce achieves reliability by parceling out a number of operations on the set of data to each node in the network; each node is expected to report back periodically with completed work and status updates. If a node falls silent for longer than that interval, the master node records the node as dead, and sends out the node's assigned work to other nodes. Individual operations use atomic operations for naming file outputs as a double check to ensure that there are not parallel conflicting threads running; when files are renamed, it is possible to also copy them to another name in addition to the name of the task (allowing for side-effects).

## 4.4. Google File System

Google File System (GFS) is a scalable distributed file system developed by Google for data intensive applications. It is designed to provide efficient, reliable access to data using large clusters of commodity hardware. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients.

Files are divided into chunks of 64 megabytes, which are only extremely rarely overwritten, or shrunk; files are usually appended to or read. It is also designed and optimized to run on computing clusters, the nodes of which consist of cheap,

"commodity" computers, which means precautions must be taken against the high failure rate of individual nodes and the subsequent data loss. Other design decisions select for high data throughputs, even when it comes at the cost of latency.

The nodes are divided into two types: one *Master* node and a large number of *Chunkservers*. Chunkservers store the data files, with each individual file broken up into fixed size chunks (hence the name) of about 64 megabytes, similar to clusters or sectors in regular file systems. Each chunk is assigned a unique 64-bit label, and logical mappings of files to constituent chunks are maintained. Each chunk is replicated several times throughout the network, with the minimum being three, but even more for files that have high demand or need more redundancy.

The Master server doesn't usually store the actual chunks, but rather all the metadata associated with the chunks, such as the tables mapping the 64-bit labels to chunk locations and the files they make up, the locations of the copies of the chunks, what processes are reading or writing to a particular chunk, or taking a "snapshot" of the chunk pursuant to replicating it (usually at the instigation of the Master server, when, due to node failures, the number of copies of a chunk has fallen beneath the set number). All this metadata is kept current by the Master server periodically receiving updates from each chunk server ("Heart-beat messages").

Permissions for modifications are handled by a system of time-limited, expiring "leases", where the Master server grants permission to a process for a finite period of time during which no other process will be granted permission by the Master server to modify the chunk. The modified chunkserver, which is always the primary chunk holder, then propagates the changes to the chunkservers with the backup copies. The changes are not saved until all chunkservers acknowledge, thus guaranteeing the completion and atomicity of the operation.

Programs access the chunks by first querying the Master server for the locations of the desired chunks; if the chunks are not being operated on (if there are no outstanding leases), the Master replies with the locations, and the program then contacts and receives the data from the chunkserver directly. As opposed to many file systems, it's not implemented in the kernel of an Operating System but accessed through a library to avoid overhead.

## 4.5.Hadoop

Hadoop is a framework for running applications on large cluster built of commodity hardware. The Hadoop framework transparently provides applications both reliability and data motion. Hadoop implements the computation paradigm named MapReduce which was explained above. The application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both MapReduce and the distributed file system are designed so that the node failures are automatically handled by the framework. Hadoop has been implemented making use of Java. In Hadoop, the combination of the entire JAR files and classed needed to run a MapReduce program is called a job. All of these components are themselves collected into a JAR which is usually referred to as the job file. To execute a job, it is submitted to a jobTracker and then executed.

Tasks in each phase are executed in a fault-tolerant manner. If node(s) fail in the middle of a computation the tasks assigned to them are re-distributed among the remaining nodes. Since we are using MapReduce, having many map and reduce tasks enables good load balancing and allows failed tasks to be re-run with smaller runtime overhead.

The Hadoop MapReduce framework has master/slave architecture. It has a single master server or a jobTracker and several slave servers or taskTrackers, one per node in the cluster. The jobTracker is the point of interaction between the users and the framework. Users submit jobs to the jobTracker, which puts them in a queue of pending jobs and executes them on a first-come first-serve basis. The jobTracker manages the assignment of MapReduce jobs to the taskTrackers. The taskTrackers execute tasks upon instruction from the jobTracker and also handle data motion between the ‘map’ and ‘reduce’ phases of the MapReduce job.

Hadoop is a framework which has received a wide industry adoption. Hadoop is used along with other cloud computing technologies like the Amazon services so as to make better use of the resources. There are many instances where Hadoop has been used. Amazon makes use of Hadoop for processing millions of sessions which it uses for analytics. This is made use of in a cluster which has about 1 to 100 nodes. Facebook uses Hadoop to store copies of internal logs and dimension data sources and

use it as a source for reporting/analytics and machine learning. The New York Times made use of Hadoop for large scale image conversions. Yahoo uses Hadoop to support research for advertisement systems and web searching tools. They also use it to do scaling tests to support development of Hadoop.

# 5. Cloud Computing Services

Even though cloud computing is a pretty new technology, there are many companies offering cloud computing services. Different companies like Amazon, Google, Yahoo, IBM and Microsoft are all players in the cloud computing services industry. But Amazon is the pioneer in the cloud computing industry with services like EC2 (Elastic Compute Cloud) and S3 (Simple Storage Service) dominating the industry. Amazon has an expertise in this industry and has a small advantage over the others because of this. Microsoft has good knowledge of the fundamentals of cloud science and is building massive data centers. IBM, the king of business computing and traditional supercomputers, teams up with Google to get a foothold in the clouds. Google is far and away the leader in cloud computing with the company itself built from the ground up on hardware.

## 5.1.Amazon Web Services

The 'Amazon Web Services' is the set of cloud computing services offered by Amazon. It involves four different services. They are Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queue Service (SQS) and Simple Database Service (SDB).

1. Elastic Compute Cloud (EC2)

   Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. It provides on-demand processing power.

   Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build

failure resilient applications and isolate themselves from common failure scenarios.

Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to requisition machines for use, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you desire.

To set up an Amazon EC2 node we have to create an EC2 node configuration which consists of all our applications, libraries, data and associated configuration settings. This configuration is then saved as an AMI (Amazon Machine Image). There are also several stock instances of Amazon AMIs available which can be customized and used. We can then start, terminate and monitor as many instances of the AMI as needed.

Amazon EC2 enables you to increase or decrease capacity within minutes. You can commission one, hundreds or even thousands of server instances simultaneously. Thus the applications can automatically scale itself up and down depending on its needs. You have root access to each one, and you can interact with them as you would any machine. You have the choice of several instance types, allowing you to select a configuration of memory, CPU, and instance storage that is optimal for your application. Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and reliably commissioned. Amazon EC2 provides web service interfaces to configure firewall settings that control network access to and between groups of instances. You will be charged at the end of each month for your EC2 resources actually consumed. So charging will be based on the actual usage of the resources.

2. Simple Storage Service (S3)

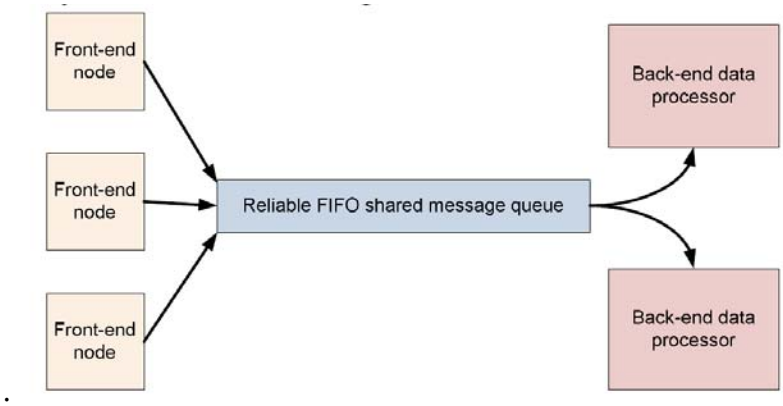
S3 or Simple Storage Service offers cloud computing storage service. It offers services for storage of data in the cloud. It provides a high-availability large-store database. It provides a simple SQL-like language. It has been designed for interactive online use. S3 is storage for the Internet. It is designed

to make web-scale computing easier for developers. 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, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites.

Amazon S3 allows write, read and delete of objects containing from 1 byte to 5 gigabytes of data each. The number of objects that you can store is unlimited. Each object is stored in a bucket and retrieved via a unique developer-assigned key. A bucket can be located anywhere in Europe or the Americas but can be accessed from anywhere. Authentication mechanisms are provided to ensure that the data is kept secure from unauthorized access. Objects can be made private or public, and rights can be granted to specific users for particular objects. Also the S3 service also works with a pay only for what you use method of payment.

3. Simple Queue Service (SQS)

Amazon Simple Queue Service (SQS) offers a reliable, highly scalable, hosted queue for storing messages as they travel between computers. By using SQS, developers can simply move data between distributed components of their applications that perform different tasks, without losing messages or requiring each component to be always available.




With SQS, developers can create an unlimited number of SQS queues, each of which can send and receive an unlimited number of messages.

Messages can be retained in a queue for up to 4 days. It is simple, reliable, secure and scalable.

4. Simple Database Service (SDB)

Amazon SimpleDB is a web service for running queries on structured data in real time. This service works in close conjunction with the Amazon S3 and EC2, collectively providing the ability to store, process and query data sets in the cloud. These services are designed to make web-scale computing easier and more cost-effective to developers. Traditionally, this type of functionality is accomplished with a clustered relational database, which requires a sizable upfront investment and often requires a DBA to maintain and administer them. Amazon SDB provides all these without the operational complexity. It requires no schema, automatically indexes your data and provides a simple API for storage and access. Developers gain access to the different functionalities from within the Amazon's proven computing environment and are able to scale instantly and need to pay only for what they use.

## 5.2. Google App Engine

Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. You can serve your app using a free domain name on the `appspot.com` domain, or use Google Apps to serve it from your own domain. You can share your application with the world, or limit access to members of your organization. App Engine costs nothing to get started. Sign up for a free account, and you can develop and publish your application at no charge and with no obligation. A free account can use up to 500MB of persistent storage and enough CPU and bandwidth for about 5 million page views a month.

Google App Engine makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. The environment includes the following features:

- dynamic web serving, with full support for common web technologies
- persistent storage with queries, sorting and transactions
- automatic scaling and load balancing
- APIs for authenticating users and sending email using Google Accounts
- a fully featured local development environment that simulates Google App Engine on your computer

Google App Engine applications are implemented using the Python programming language. The runtime environment includes the full Python language and most of the Python standard library. Applications run in a secure environment that provides limited access to the underlying operating system. These limitations allow App Engine to distribute web requests for the application across multiple servers, and start and stop servers to meet traffic demands.

App Engine includes a service API for integrating with Google Accounts. Your application can allow a user to sign in with a Google account, and access the email address and displayable name associated with the account. Using Google Accounts lets the user start using your application faster, because the user may not need to create a new account. It also saves you the effort of implementing a user account system just for your application

App Engine provides a variety of services that enable you to perform common operations when managing your application. The following APIs are provided to access these services: Applications can access resources on the Internet, such as web services or other data, using App Engine's URL fetch service. Applications can send email messages using App Engine's mail service. The mail service uses Google infrastructure to send email messages. The Image service lets your application manipulate images. With this API, you can resize, crop, rotate and flip images in JPEG and PNG formats.

In theory, Google claims App Engine can scale nicely. But Google currently places a limit of 5 million hits per month on each application. This limit nullifies App Engine's scalability, because any small, dedicated server can have this performance. Google will eventually allow webmasters to go beyond this limit (if they pay).

# 6. Cloud Computing in the Real World

## 6.1. Time Machine

Times machine is a New York Times project in which one can read any issue from Volume 1, Number 1 of The New York Daily Times, on September 18, 1851 through to The New York Times of December 30, 1922. They made it such that one can choose a date in history and flip electronically through the pages, displayed with their original look and feel. Here's what they did. They scanned all their public domain articles from 1851 to 1992 into TIFF files. They converted it into PDF files and put them online. Using 100 Linux computers, the job took about 24 hours. Then a coding error was discovered that required the job be rerun. That's when their software team decided that the job of maintaining this much data was too much to do in-house. So they made use of cloud computing services to do the work.

All the content was put in the cloud, in Amazon. They made use of 100 instances of Amazon EC2 and completed the whole work in less than 24 hours. They uploaded all the TIFF files into the cloud and made a program in Hadoop which does the whole job. Using Amazon.com's EC2 computing platform, the Times ran a PDF conversion app that converted that 4TB of TIFF data into 1.5TB of PDF files. The PDF files were such that they were fully searchable. The image manipulation and the search ability of the software were done using cloud computing services.

## 6.2. IBM Google University Academic Initiative

Google and IBM came up with an initiative to advance large-scale distributed computing by providing hardware, software, and services to universities. Their idea was to prepare students "to harness the potential of modern computing systems," the companies will provide universities with hardware, software, and services to advance training in large-scale distributed computing. The two companies aim to reduce the cost of distributed computing research, thereby enabling academic institutions and their students to more easily contribute to this emerging computing paradigm. Eric Schmidt, CEO of Google, said in a statement. "In order to most effectively serve the long-term interests of our users, it is imperative that students are adequately equipped

to harness the potential of modern computing systems and for researchers to be able to innovate ways to address emerging problems."

The first university to join the initiative is the University of Washington. Carnegie-Mellon University, MIT, Stanford University, the University of California at Berkeley, and the University of Maryland are also participating in the program.

As part of the initiative, Google and IBM are providing a cluster of several hundred computers -- Google's custom servers and IBM BladeCenter and System x servers. Over time, the companies expect the cluster to surpass 1,600 processors. The Linux-based servers will run open source software including Xen's virtualization system and Hadoop, an open source implementation of Google's distributed file system that's managed by the Apache Software Foundation.

Students working with the cluster will have access to a Creative Commons-licensed curriculum for massively parallel computing developed by Google and the University of Washington.

## 6.3.SmugMug

SmugMug is an online photo hosting application which is fully based on cloud computing services. They don’t own any hard drives. All their storage is based in the Amazon S3 instances.

## 6.4.Nasdaq

NASDAQ which had lots of stock and fund data wanted to make extra revenue selling historic data for those stocks and funds. But for this offering, called Market Replay, the company didn't want to worry about optimizing its databases and servers to handle the new load. So it turned to Amazon's S3 service to host the data, and created a lightweight reader app that let users pull in the required data. The traditional approach wouldn't have gotten off the ground economically. NASDAQ took its market data and created flat files for every entity, each holding enough data for a 10-minute replay of the stock's or fund's price changes, on a second-by-second basis. It adds 100,000 files per day to the several million it started with.

# 7. Conclusion

Cloud computing is a powerful new abstraction for large scale data processing systems which is scalable, reliable and available. In cloud computing, there are large self-managed server pools available which reduces the overhead and eliminates management headache. Cloud computing services can also grow and shrink according to need. Cloud computing is particularly valuable to small and medium businesses, where effective and affordable IT tools are critical to helping them become more productive without spending lots of money on in-house resources and technical equipment. Also it is a new emerging architecture needed to expand the Internet to become the computing platform of the future.

# 8. References

1. http://www.infoworld.com/article/08/04/07/15FE-cloud-computing-reality_1.html,
   "What Cloud Computing Really Means"

2. http://www.spinnakerlabs.com/CloudComputing.pdf
   "Welcome to the new era of cloud computing PPT"

3. http://www.johnmwillis.com/
   "Demystifying Clouds" - discusses many players in the cloud space