International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 4 (April 2014), PP.58-66

A Comparative Study of Grid Computing and Cloud Computing

Anushila Dey[1], Aditi Dutt[2], Sumeet Kumar Jain[3], Vaishali Pandurangan[4]
[1,2,3,4]Department of Computer Engineering, NMIMS Mukesh Patel School of Technology Management &Engineering, Mumbai, India

Abstract: - The present competitive world is characterized by individuals and businesses constantly trying to adapt themselves to progressive technological innovations for competitive advantage to stay ahead of the race. Cloud Computing is one such innovation which has made considerable impact in the market. It is the practice of delivering services over the Internet. These services can be broadly categorized into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). For instance, more and more organizations and individuals are migrating data to the servers on the cloud, thus reaping the benefits of IaaS which eliminates the need of own data centres and lowers maintenance costs. Cloud Computing is the amalgamation of existing technologies like Grid Computing, Utility Computing and several other models. Grid Computing is a form of distributed and parallel computing. In simple terms, it is a set of distributed resources working together to achieve a mutual goal thereby enhancing computational power by sharing of resources. In this paper, we try to compare and contrast the models of Grid and Cloud Computing. We also discuss their essential characteristics by reviewing a handful of research papers based on the two technologies.

Keywords: - Cloud Computing; Grid Computing; Grids vs. Clouds

I. OVERVIEW: GRID COMPUTING

Grid computing is an enhanced form of distributed computing. It is different from distributed computing in a way that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed. It focuses on large scare resource sharing and applies these resources to work on a single problem which can require access to huge amounts of data or an inordinate amount of CPU power at the same time in a network.

Grid computing is similar to electric power grid. Electric power grid is a concept in which a user could obtain electric power from any power station present in in the power grid regardless of its geographical location. Whenever a user needed extra or additional power, he or she could simply plug into an electric power grid and receive power on demand. Similarly in Computational Power Grid, the users need to connect to a Computational Grid in order to get access to additional computing power on demand.

In the early 1990s, the term 'grid computing' was originated and it was to make computational power easily accessible to people on demand.

The Grid Computing paradigm became recognized when Ian Foster and Carl Kesselman published their book, "The Grid: Blueprint for a new computing infrastructure" in 1999.

When Ian Foster built the idea of the 'Grid', he gave a three point checklist to help define what grid is: [1]

1. Coordinates resources that are not subject to centralized control,
2. Uses standard, open, general-purpose protocols and interfaces, and
3. Delivers non-trivial qualities of service.

Ian Foster, Carl Kesselman and Steve Tuckie are widely regarded as the "fathers of the grid" as they gave rise to the idea of grid from various existing technologies like distributed computing, object-oriented programming, etc.

They created the Globus Toolkit which included computation management, storage management, security provisioning, data movement, monitoring, etc. Many other such tools have been built which help to build the services needed to create an enterprise or global grid.

According to Foster, grid computing is hardware and software infrastructure which offer a cheap, distributable, coordinated and reliable access to powerful computational capabilities [1].

II. OVERVIEW: CLOUD COMPUTING

During the last several decades, the advancements in technologies have helped the human race to create, operate and allocate growing amount of information in many new ways. The new computing applications, in turn, lead to new demands for even more powerful computing infrastructure. Cloud Computing is thus a computing paradigm, where the data from different locations and datacentres is stored and it provides dynamically scalable infrastructure for application, data and file storage.

Thus, Cloud Computing is said to be a model that delivers information technology services, where the resources are recovered from the web via web-based tools and applications. Cloud Computing information and services can be accessed only when an electronic device has access to the web. Thus, this allows the employees to work from remote places.

The principle of Cloud Computing is “reusability of IT resources”. Cloud Computing is generally compared to traditional concepts of “Grid Computing”, “distributed computing”, “Utility Computing”, or “autonomic computing” is to broaden horizons across organizational boundaries.

Cloud Computing is known to be a specialized distributed paradigm. In contrast to the traditional ones, it is scalable; can be enclosed as an abstract entity that helps in delivering different levels of services to customers outside the Cloud, driven by economies of scale. The services thus provided can be delivered on demand and can be dynamically configured.

The factors which can contribute to the flow of Cloud Computing:

1. Decrease in hardware cost, increase in computing power and storage capacity.
2. The rapid growing data size in internet publishing and archiving.
3. Adoption of Services Computing and Web 2.0 applications.

Many factors of Cloud Computing overlap with the other existing technologies like Grid Computing, Utility Computing, distributed computing in general. In fact, Grid Computing is known to be the backbone of Cloud Computing and thus provide infrastructure support. So, Cloud Computing is evolved out of Grid Computing. The rise of this evolution was the shift in focus from an infrastructure that delivers storage and compute resources (such is the case in Grids) to one that is economy based aiming to deliver more abstract resources and services (such is the case in Clouds). In Utility Computing, it is a business model in which computing resources, such as computation and storage, are packaged as metered services similar to a physical public utility, such as electricity and public switched telephone network.

III. GRID CHARACTERISTICS

The collaborative nature of Grids led to the emergence of multiple organizations that function as one unit through the use of their shared competencies and resources for the purpose of one or more identified goals. Thus, administration of resources in Grids is solved with the concept of virtual organization representing a dynamic set of individuals and/or institutions aligned with a set of resource-sharing rules and conditions to solve a specific (research) goal. Thus, organizations and individuals belonging to a specific virtual organization may share resources for a specific time frame to achieve certain research goal.

Some characteristics of grid are:

- **Large scale:** a Grid must be able to deal with a number of resources ranging from just a few to millions. This raises problem of performance degradation as the Grid size increases.
- **Heterogeneity:** [3] a Grid hosts both software and hardware resources that can be vary ranging from data, files, software components or programs to sensors, scientific instruments, display devices, personal digital organizers, computers, super-computers and networks.
- **Geographical distribution:** [3] Grid’s resources may be located at distant places.
- **Resource sharing:** resources in a Grid belong to many different organizations that allow other organizations (i.e. users) to access them. Non-local resources can thus be used by applications, promoting efficiency and reducing costs.
- **Resource coordination:** resources in a Grid must be coordinated in order to provide combined computing capabilities.
- **Multiple administrations:** [3] each organization may establish different security and administrative policies under which their owned resources can be accessed and used.
- **Consistent access:** a Grid must be built with standard services, protocols and inter-faces thus hiding the heterogeneity of the resources while allowing its scalability. Without such standards, application development and pervasive use would not be possible.
- **Pervasive access:** the Grid must grant access to available resources by adapting to a dynamic environment in which resource failure is commonplace.[3]
- **Transparent access:** a Grid should be seen as a single virtual computer.
- **Dependable access:** a grid must assure the delivery of services under established Quality of Service (QoS) requirements.

IV. CLOUD CHARACTERISTICS

Cloud Computing does not represent a new technology. Rather, it is considered to be a novel form of how existing technologies are used to achieve efficient resource pooling and resource management. Thus, Cloud Computing relies on existing technologies like Grid Computing, service oriented computing, virtualization, Web 2.0, and similar by augmenting those to achieve Cloud related goals like elasticity and energy efficiency. Virtualization lies at the heart of Cloud Computing. It separates computation and technology from the hardware layer and facilitates on demand provisioning of computational resources for arbitrary users.

The 5 essential characteristics of Cloud Computing are as follows:

- **On demand self-services**: computer services such as email, applications, network or server service can be provided without requiring human interaction with each service provider. E.g., Amazon Web Services (AWS), Microsoft, Google, and IBM.
- **Broad network access:** Cloud capabilities are available over the network and can be accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as mobile phones, laptops and PDAs.
- **Resource pooling:** The service providers’ computing resources are pooled together to cater to multiple customers with the help of multi-tenancy. Resources are dynamically assigned to consumers on demand.
- **Rapid elasticity:** Cloud services can be rapidly and elastically provisioned to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured service:** Cloud computing resource usage can be measured, controlled, and reported providing transparency for both the provider and consumer.

V. GRIDS VS. CLOUDS

A. Business Model

The Business model for software is used as a one-time payment for unlimited usage of the software.

The main characteristic of a Cloud based model is “Measured Service” i.e. pay per use. In a cloud-based business model, a customer will pay the provider on a consumption basis, very much like the utility companies charge for basic utilities.

Amazon essentially provides a centralized Cloud consisting of Compute Cloud EC2 and Data Cloud S3. In EC2, the charges are based on consumption for each instance type and Data Cloud S3 is charged by per GB-Month of storage used and data transfer is charged by TB / month data transfer, depending on the source and target of such transfer. To use this service the user only needs a credit card to get on-demand access to processors in data centres which is distributed throughout the world [2].

The business model for Grid is project-oriented in which the users or community have certain number of service units (i.e. CPU hours) they can spend [2].

For example, the TeraGrid uses this technique and requires increasingly complex proposals be written for increasing number of computational power [4]. The TeraGrid has a large number Grid sites which are hosted at various institutions around the country.

This model has worked well for many Grids around the globe, giving institutions (e.g. TeraGrid) incentives to join various Grids for access to additional resources for all the users from the corresponding institution.

B. Architecture

1) Architecture of Grid Computing

Grids use a network of resource-sharing commodity machines helping in delivering the computation power which is affordable only by the supercomputers and large dedicated clusters used at the time of mid-90s. The integration of the existing resources with their operating systems, hardware, and security infrastructure is focused by the grids. A set of standard protocols, middleware, toolkits and services built on the top of the protocols are defined and provided by the Grids.

The architecture of Grid Computing is described in terms of “layers”, and each layer has a different specific function. The upper layers are generally user-centric, whereas lower layers are more hardware-centric, which mainly focuses on the computer and networks.

The Grid includes five different layers:

1. **Fabric Layer:** The standardized access to the local resource-specific operations is provided by the Grid. Grids help in discovering computers (OS version, hardware configuration, and usage load), storage systems, network resource etc. They usually rely on existing fabric components, for instance, local resource managers. General-purpose components such as GARA (general architecture for advanced reservation) [8].

2. Connectivity Layer: This layer ensures secure connectivity to resources. It defines core communication and authentication protocols for easy and secure network transactions. This layer uses the Public Key Infrastructure (PKI). Every user is recognized by a Certificate Authority (CA) within the grid. In this, the method of single sign-on allows users to authenticate only once. It creates proxy credentials to allow services/agents to act on a user's behalf. The GSI (Grid Security Infrastructure) [9] protocol underlies every Grid transaction.

3. Resource Layer: This layers provides access to the resources and protocols needed for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The GRAM (Grid Resource Access and Management) [10] acts as a job manager and reporter. It helps in allocating the various computational resources and thus monitoring the control of computation of those resources. GridFTP [11] is another protocol for accessing the data faster and holds integrated security.

4. Collective Layer: This layer coordinates the sharing of resources like directory services. It monitors and diagnoses the services. Directory services such as MDS (Monitoring and Discovery Service) [12] allows for the monitoring and discovery of resources. Condor-G [13] and Nimrod-G [14] are examples of co-allocating, scheduling and brokering services, and MPICH [15] for Grid enabled programming systems, and CAS (community authorization service) [16] for global resource policies.

5. Application Layer: The application layer is known to be the highest layer of the Grid structure. This is the layer that Grid users "see" and interact with. This layer comprises of the user applications built on the protocols and APIs and these operate in Virtual Organization (VO) environment. The application layer performs the general management functions like tracking the users using the grid services and the service providers. Grid workflow systems and Grid portals are good examples of the user applications in the application layer.

2) Architecture of Cloud Computing

Clouds are known to be a large pool of computing or storage resources, which can be easily accessed through standard protocols or through an abstract interface. Clouds can also be implemented over the existing technologies based on grid, which in turn leverages the efforts in security, resource management, and virtualization support.

Cloud Computing can be known to have a four layer architecture in comparison to Grid Computing. They are as follows:

1. Fabric Layer: This layer is the lowest layer in the structure of Cloud Computing. It comprises of the raw hardware level resources such as compute resources, storage resources and network resources.

2. Unified Resource Layer: This layer comprises of the encapsulated/abstracted resources which can be exposed to the upper layer and the end users in the form of integrated resources, for example, a virtual computer/cluster, database system etc.

3. Platform Layer: This layer acts as an add-on, on top of the unified resources in the form of a collection of specialized tools, middleware and services and provides a development platform. For instance, a Web hosting environment, etc.

4. Application Layer: This layer contains the applications that would run in the cloud for the users.

Cloud in general provides its users with the services according to their needs. So, the users can ask for the amount of services they require. Thus, Clouds provide services at three different levels. These levels are as follows:

1. Infrastructure as a Service (IaaS) [17]: This cloud service provides the cloud users with hardware, software, and equipment (mostly at the unified resource layer) to deliver software application environments with a resource usage-based pricing model. Based on application resource needs, infrastructure can scale up and down dynamically. For instance, Amazon EC2 (Elastic Cloud Computing) [18] Service and S3 (Simple Storage Service) [19], public can access both compute and storage infrastructures with a utility pricing model.

2. Platform as a Service (PaaS) [17]: It provides the Cloud users a high-level integrated environment to build, test and deploy their applications. There are certain restrictions given to the developers on the type of software they can write in exchange for built-in application scalability. For example, Google's App Engine [20] which provides users the capability to build web applications on the same scalable systems that power Google applications.

3. Software as a Service (SaaS) [17]: This delivers the users with special-purpose software that can be remotely accessed by consumers through the Internet with a usage-based pricing model. For example Salesforce.

C. Resource Management

Resource management is found both in Grids and Clouds and includes topics such as the compute model, data model, virtualization and monitoring.

These topics are significant and help in understanding and resolving the main challenges involved in Grid as well as Cloud Computing.

1) Compute Model

Most Grid systems use a batch-scheduled compute model, in which a local resource manager i.e. LRM manages the compute resources for a Grid site, and users submit batch jobs to request and access resources for specific period of time.

Most of the Grids have rules in place that make it compulsory for the batch jobs to identify users and keep track of their credentials, number of resources needed and also the duration of the allocation.

Cloud Computing supports multi-tenancy. In the compute model of Cloud Computing, all the resources are shared by all the users at the same time.

2) Data Model

Data is an extremely vital part in both cloud and grid computing. Data Grids have been designed specially to handle data intensive applications in Grid environments. In this case, virtual data plays a crucial role as it captures relationships between data, programs and computations and proposes various abstractions that a data grid can provide such as location transparency where data can be accessed without specifying data location, materialization transparency: data can be either recomputed on the fly or transferred upon request, depending on the availability of the data and the cost to re-compute, representation transparency where data can be consumed and produced no matter what their actual physical formats and storage are, data are mapped into some abstract structural representation and manipulated in that way [2].

3) Data Locality

Data should be located closer to CPUs to achieve good scalability and data must be distributed over various computers to minimize communication costs. Hence, data processing depends on data storage. When a file needs to be processed, the job scheduler will first consult a storage metadata service to get the host node for each chunk of data and then maps it to the processors which require it.

4) Virtualization

Cloud computing is built on the concept of virtualization. Clouds often run multiple user applications simultaneously. Also, the applications must be able to use all the available resources concurrently without any interruption of service. Virtualization provides the necessary abstraction such that the underlying hardware resources such as storage, network, servers, etc. can be unified together to create a resource pool which can be made available to all the user applications. Some reasons why Clouds adopt virtualization:

1. Efficient utilization of resources as multiple applications run on the same server.
2. Dynamic Configurability for different kinds of applications having different kinds of needs, as the resource requirements for various applications could differ significantly.
3. Quick migration of virtual environments from one system to another without service interruption thus, providing business continuity
4. Automation of resource provisioning, monitoring and maintenance and also caching and reusing resources which in turn improve overall responsiveness.

Grids rely lesser on virtualization compared to Clouds due to certain policies and also due to the fact that each individual organization maintains full control of their resources by not virtualizing them.

5) Monitoring

Monitoring the resources is a challenge that virtualization brings to cloud computing. Many Grids such as TeraGrid also enforce restrictions on what kind of sensors or long-running services a user can launch. Grid monitoring is straight forward because Grids have a different trust model in which users via their identity can access and browse resources. Grid resources are not highly abstracted and virtualized as in Clouds.

In a Cloud, different levels of services can be offered to an end user, the user is only exposed to a predefined API, and the lower level resources are opaque to the user (especially at the PaaS and SaaS level). The user does not have the liberty to deploy him/her own monitoring infrastructure, and the limited information returned to the user may not provide the necessary level of details for him/her to figure out what the resource status is. Monitoring can be argued to be less important in Clouds, as users are interacting with a more abstract layer that is potentially more sophisticated.

D. Application Model

1) Grid Computing

Grids have different types of applications, which can differ from high performance computing (HPC) to high throughput computing (HTC). For inter-process communication, these applications mostly use message passing interface. HPC Applications consist of concurrent programs which are designed for multi-threaded as well as multi-process models. The applications also consists of various parallel constructs like threads, local processes, distributed processes, etc. with changing degree of parallelism. HPC application is easily executed tightly coupled parallel jobs within a particular machine with low-latency interconnects and not executed across a wide area network. Grids successfully executed more number of loosely coupled applications that tend to be managed and executed through workflow systems or other sophisticated system and complex applications. The

HTC applications are loosely coupled in nature. These loosely coupled applications are composed of both independent and dependent tasks. Tasks can be small or large, compute-intensive or data-intensive, uniprocessor or multiprocessor. The set of tasks can be static or dynamic, homogeneous or heterogeneous, loosely or tightly coupled.

2) Cloud Computing

Cloud Computing principle provides a similar set of applications. The one thing which is difficult to achieve in Cloud Computing (much success in grids) are HPC applications which require faster and low latency network interconnects for efficient scaling to number of processors.

Scientific gateways is a class of Grid applications which form front-ends to a variety of loosely-coupled and tightly-coupled applications. A Science Gateway is a community-developed set of tools, applications, and data that are integrated via a portal or a suite of applications, usually in a graphical user interface, that is further customized to meet the needs of a specific community. Gateways enable entire communities of users associated with a common discipline to use national resources through a common interface that is configured for optimal use [21]. Web 2.0 technologies are being adopted by the scientific gateways. However, the developments in both Grids and Web 2.0 have been made with very little communication and collaboration between them. Although till now, scientific gateways have only emerged in Grids, in Clouds the gateways have been adopted exclusively for the end-user interaction. The major role will be of the browser and Web 2.0 technologies which will in turn help in how users will interact with Grids and Clouds in the future.

E. Security Model

1) Grid Computing

The security has been designed and built in the fundamental Grid infrastructure. The Grid Security Infrastructure (GSI) is known to be a term for secure, secret, communication between the different software's in grid computing. The key security issues in Grids are: single sign-on, in this the Grid users are allowed to log in only once and get permission for a particular required page or multiple Grid sites, this will in turn make the task easier for accounting and auditing; delegation, only the authorized users can have the access to a particular program, or application to get the required resources and it can further represent to other programs; privacy, integrity and segregation, resources here belongs to one user. It cannot be accessed by any other unauthorized user, and cannot be interfaced during transfer; it can also be communicated for resource allocation, reservation, and sharing, considering both global and local resource usage policies. The public-key based GSI (Grid Security Infrastructure) protocol is used for authentication, communication protection, and authorization. For advanced resource authorization and across communities, CAS (Community Authorization Service) is designed. The Grid computing security is more time consuming, but it adds one extra level of security to prevent unauthorized access and many other.

2) Cloud Computing

Security model in Clouds Computing is known to be simple and less secure than the Grid Computing security model. Cloud infrastructure typically is dependent on Web forms (over SSL) which help in creating and managing the account information for end-users, making their work easy and allow the cloud users to reset or change their passwords, in turn receiving new passwords via Emails in an unsafe and unencrypted communication.

Security is one of the largest concern for the adoption of Cloud Computing. There are seven risks a Cloud user should raise with agent before committing:

1. **Privileged user access:** sensitive data if processed outside the enterprise needs the assurance that they are only accessible and propagated to privileged users;
2. **Regulatory compliance:** a Cloud provider should have external audits and security certifications and the customer needs to verify this, and also if their infrastructure complies with some regulatory security requirements;
3. **Data location:** it is important that the Cloud provider should commit to store and process data in specific jurisdictions and to obey local privacy requirements on behalf of the customer, as a customer will not know where his/her data will be stored;
4. **Data segregation:** one needs to ensure that one customer's data is fully segregated from another customer's data;
5. **Recovery:** it is important that the Cloud provider should have efficient replication and recovery mechanism to restore data if a disaster occurs;
6. **Investigative support:** Cloud services are especially difficult to investigate, if this is important for a customer, then such support needs to be ensured with a predetermined commitment;

7. **Long-term viability:** if the Cloud provider is acquired by another company, the data of the cloud users should be viable. Sensitive data processed outside the enterprise needs the assurance that they are only accessible and propagated to privileged users.

VI. GRID COMPUTING VS. CLOUD COMPUTING

Table I: GC VS.CC

Parameter	Grid Computing	Cloud Computing
When?	The concept of grids was proposed in 1995. The Open science grid (OSG) started in 1995 The EDG (European Data Grid) project began in 2001.	In the late 1990`s Oracle and EMC offered early private cloud solutions. However the term cloud computing didn't gain prominence until 2007.
What?	Grids enable access to **shared** computing power and storage capacity from your desktop	Clouds enable access to **leased** computing power and storage capacity from your desktop
Why use them?	• You don`t need to buy or maintain your own large computer centre • You can complete more work more quickly and tackle more difficult problems. • You can share data with your distributed team in a secure way.	• You don`t need to buy or maintain your own personal computer center • You can quickly access extra resources during peak work periods
Workflow Management	Physical node	EC2 instance
Where are the computing resources?	In computing centres distributed across different sites, countries and continents.	The cloud provider's private data centers which are often centralized in a few locations with excellent network connections and cheap electrical power.
Who uses the service?	Research collaborations, called "Virtual Organizations", which bring together researchers around the world working in the same field.	Small to medium commercial businesses or researchers with generic IT needs
Who pays for the service?	Governments - providers and users are usually publicly funded research organisations, for example through National Grid Initiatives.	The cloud provider pays for the computing resources; the user pays to use them
What are they useful for?	Grids were designed to handle large sets of limited duration jobs that produce or use large quantities of data	Clouds best support long term services and longer running jobs (E.g. facebook.com)
Benefits	• **Collaboration:** grid offers a federated platform for distributed and collective work. • **Ownership :** resource providers maintain ownership of the resources they contribute to the grid • **Transparency:** the technologies used are open source, encouraging trust and transparency. • **Resilience:** grids are located at multiple sites, reducing the risk in case of a failure at one site that removes significant resources from the infrastructure.	• **Flexibility:** users can quickly outsource peaks of activity without long term commitment • **Reliability:** provider has financial incentive to guarantee service availability (Amazon, for example, can provide user rebates if availability drops below 99.9%) • **Ease of use:** relatively quick and easy for non-expert users to get started but setting up sophisticated virtual machines to support complex applications is more difficult.

Drawbacks	• **Reliability:** grids rely on distributed services maintained by distributed staff, often resulting in inconsistency in reliability across individual sites, although the service itself is always available. • **Complexity:** grids are complicated to build and use, and currently users require some level of expertise. • **Commercial:** grids are generally only available for not-for-profit work, and for proof of concept in the commercial sphere	• **Generality:** clouds do not offer many of the specific high-level services currently provided by grid technology. • **Security:** users with sensitive data may be reluctant to entrust it to external providers or to providers outside their borders. • **Opacity:** the technologies used to guarantee reliability and safety of cloud operations are not made public. • **Rigidity:** the cloud is generally located at a single site, which increases risk of complete cloud failure. • **Provider lock-in:** there's a risk of being locked in to services provided by a very small group of suppliers.
Principle	Grid needs processing from you	Cloud does the processing for you
Goal	Pervasive, uniform, and reliable access to data, storage capacity and computation power	Use of different services such as servers, storage and applications are delivered to an organization's computers and devices through the Internet.
Functioning	Grid computing separate everything into different parts.	Cloud computing arrange everything into one place.
Transparency	Low	High
Ownership	Multiple	Single
Multitask	Yes	Yes
Types of service	Network, memory, CPU, bandwidth, device, storage, etc.	IaaS, PaaS, SaaS, Everything as a service
Example of real world	SETI, BOINC, Folding@home, GIMPS	Amazon Web Service (AWS), Google apps
Number of users	Few	Unlimited
Operating System	Any OS	A high performance machine (virtual machine) on which multiple OS can runs
Resource management	Distributed (separated)	Centralized/Distributed Both
Scheduling	Decentralized	Both Centralized/ Decentralized
Infrastructure	Low level	High level
Scalability	Normal	High
Abstraction	Low	High
Bandwidth	Low	High
Future	Cloud computing	Next generation of internet

VII. CONCLUSION

Cloud Computing is selling like hot cakes nowadays, especially in the IT industry. There is a motley of Cloud service providers like Google, Amazon, Salesforce, Dropbox, etc. which offer a variety of services each with flexible pricing options and scalability.

Even though Cloud provides a plethora of features and flexibilities, there's a security concern in Cloud Computing and this can be regarded as one of its drawbacks. A lot of work is still to be done in the security aspect of Cloud Computing which will make it more accessible and appealing to the customers due to increased reliability and security.

Grid Computing provides cost-effectiveness in utilizing resources, it helps to solve problems with enhanced processing power and it also helps in collaborating resources from many computers and tries to use them in aggregation.

Cloud Computing is strongly related to Grid Computing and as a result Grid and Cloud both have similarities like sharing of resources such as computational power, storage, application, equipment, etc. but also have various differences.

In this paper, we acquainted ourselves with the basic ideas of both Clouds and Grids and the mechanism in which they operate. We also learnt and reviewed the main similarities and differences between them which in turn helped us to clearly understand the dichotomy between the two analogous technologies.

REFERENCES

[1]. Foster. What is the Grid? A Three Point Checklist, July 2002.

[2]. Ian Foster,Yong Zhao, Ioan Raicu, Shiyong Lu "Cloud Computing and Grid Computing 360-Degree Compared"

[3]. Seyyed Mohsen Hashemi,Amid Khatibi Bardsiri "Cloud Computing Vs. Grid Computing", VOL. 2, NO.5, ARPN Journal of Systems and Software ,MAY 2012

[4]. Prof. Milan Kantilal Vachhani, Dr. Kishor H. Atkotiya "Similarities and Contrast between Grid Computing and Cloud Computing", Volume : 3 , Issue : 3, INDIAN JOURNAL OF APPLIED RESEARCH, March 2013

[5]. Bhawna Saxena, "Grid Computing: Virtualization Of Distributed Computing And Data Resources" , Vol. 4, No.1, The Journal of Computer Science and Information Technology, 2006

[6]. "What is cloud computing?

[7]. "http://searchcloudcomputing.techtarget.com/sDefinition/0sid201gci1287881, 00.html.

[8]. Microsoft, "Windows Azure", http://www.microsoft.com/windows azure.

[9]. I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.

[10]. The Globus Security Team. "Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective," Technical Report, Argonne National Laboratory, MCS, 2005.

[11]. I. Foster, C. Kesselman. "Globus: A Metacomputing Infrastructure Toolkit", Intl J. Supercomputer Applications, 11(2):115-128, 1997.

[12]. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke. "Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.

[13]. J. M. Schopf, I. Raicu, L. Pearlman, N. Miller, C. Kesselman, I. Foster, M. D'Arcy. "Monitoring and Discovery in a Web Services Framework: Functionality and Performance of Globus Toolkit MDS4", Technical Report, Argonne National Laboratory, MCS Preprint #ANL/MCS-P1315-0106, 2006.

[14]. J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke. "Condor-G: A Computation Management Agent for MultiInstitutional Grids," Cluster Computing, 5 (3). 237-246. 2002.

[15]. R. Buyya, D. Abramson, J. Giddy. "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid", IEEE Int. Conf. on High Performance Computing in Asia-Pacific Region (HPC ASIA) 2000.

[16]. N. Karonis, B. Toonen, and I. Foster. MPICH-G2: A GridEnabled Implementation of the Message Passing Interface. Journal of Parallel and Distributed Computing, 2003.

[17]. I. Foster, C. Kesselman, L. Pearlman, S. Tuecke, and V. Welch. "The Community Authorization Service: Status and Future," In Proc. of Computing in High Energy Physics (CHEP), 2003.

[18]. "What is Cloud Computing?" Whatis.com. http: //searchsoa.techtarget.com/sDefinition/0,,sid26_gci1287881,00.html, 2008.

[19]. Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2, 2008.

[20]. Amazon Simple Storage Service (Amazon S3), http://aws.amazon.com/s3, 2008.

[21]. Google App Engine, http://code.google.com/appengine/, 2008.

[22]. "XSEDE, Extreme Science and Engineering Discovery Environment", https: //www.xsede.org/gateways-overview.