

A Research on Cloud Computing and Grid Computing: Some Comparison

Ankit Bansal^{1*} Dr. Vijay Athavale²

¹ Assistant Professor, Gulzar Group of Institutes, Ludhiana

² Professor and Director, ABES Engineering College, Ghaziabad, Uttar Pradesh

Abstract – Cloud computing is the most recent announced technology that has been launched on the network world. Clouds are considered as a new generation of Grid computing. Clouds consist of data centres which are owned by the same institute. The homogeneity within each data centre in the infrastructure is the main feature for the cloud computing compared to grid computing. This paper provides a definition for the cloud, it discusses many aspects of Cloud Computing, and describes architectures for the cloud (by looking at Amazon's application (Grep The Web)) and how its cost definition differs from that of Grid computing. This paper focuses on comparing Cloud Computing to previous generations such as Grid Computing, by reviewing some Security and Policy Issues in Cloud and Grid Computing. At the end, this paper describes the similarities and differences between the Grid and Cloud approaches. Two of most well-known computing paradigms are considered throughout this research. These are: cluster and cloud computing paradigms. Each of the two paradigms is defined, architecture is considered, areas of applications of each paradigm are explored. Cloud Computing has become another buzzword after Web 2.0. However, there are dozens of different definitions for Cloud Computing and there seems to be no consensus on what a Cloud is. On the other hand, Cloud Computing is not a completely new concept; it has intricate connection to the relatively new but thirteen-year established Grid Computing paradigm, and other relevant technologies such as utility computing, cluster computing, and distributed systems in general.

-----X-----

INTRODUCTION

Cloud computing provides secure anytime-anywhere access, high-level security and data privacy. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software and information are provided to computers and other devices as a utility. Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Cloud computing providers deliver applications via the internet, which are accessed from a Web browser, while the business software and data are stored on servers at a remote location. Cloud Computing defined according to Foster, Zhao, Raicu and Lu as: "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet." A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and

presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers.

Cloud computing tries to enhance the concept of Grid computing and avoid the single point of failure when one unit fails. Cloud computing is designed to act as one whole and its compound units to be automatically interchangeable. That is why cloud hosting offers the highest level of data security in the hosting market. Cloud refers to a collection of nodes and the Cloud computing model focuses on the Cloud to provide the services to the customers. Computing clouds & grids are changing the whole IT, service industry, and global economy. Clearly, cloud computing demands ubiquity, efficiency, security, and trustworthiness. Cloud & Grid computing has become a common practice in business, government, education, and entertainment leveraging 50 millions of servers globally installed at thousands of datacenters today.

Living in a world crowded of knowledge and information, connected through the Internet, and

open for access to all knowledge seekers, with a great emphasis on the pace of information exchange and the innovations and new trends in the IT industry, we find ourselves stuck, unable to move a step forward due to misperception of a small definition of a new technology or a trend that leads to innovative jump from one place to another. This leads us to conducting this research as a means of putting the things straight and unveiling the uncertainty and the foggy image of three paradigms important paradigms, these are: cloud computing and grid computing. No doubt, cloud computing has become a ubiquitous technology that nobody can ignore, and many businesses, enterprises, and computer professionals are moving towards it and its applications. Grid computing is a newer technology and it gained a lot of momentum. Cloud computing is one of the latest technologies in the IT industry. It's based on the ideas of service provisioning and tenancy. In cloud computing, service-level agreements (SLAs) are set to control the tenancy of services between the clouding company and the tenant (beneficiary).

Cloud Computing is hinting at a future in which we won't compute on local computers, but on centralized facilities operated by third-party compute and storage utilities. We sure won't miss the shrink-wrapped software to unwrap and install. Needless to say, this is not a new idea. In fact, back in 1961, computing pioneer John McCarthy predicted that "computation may someday be organized as a public utility"—and went on to speculate how this might occur.

In the mid-1990s, the term Grid was coined to describe technologies that would allow consumers to obtain computing power on demand. Ian Foster and others posited that by standardizing the protocols used to request computing power, we could spur the creation of a Computing Grid, analogous in form and utility to the electric power grid. Researchers subsequently developed these ideas in many exciting ways, producing for example large-scale federated systems (Tera Grid, Open Science Grid, caBIG, EGEE, Earth System Grid) that provide not just computing power, but also data and software, on demand. Standards organizations (e.g., OGF, OASIS) defined relevant standards. More prosaically, the term was also co-opted by industry as a marketing term for clusters. But no viable commercial Grid Computing providers emerged, at least not until recently.

So is "Cloud Computing" just a new name for Grid? In information technology, where technology scales by an order of magnitude, and in the process reinvents itself, every five years, there is no straightforward answer to such questions. **Yes:** the vision is the same—to reduce the cost of computing, increase reliability, and increase flexibility by transforming computers from something that we buy and operate ourselves to something that is operated by a third party.

But no: things are different now than they were 10 years ago. We have a new need to analyze massive data, thus motivating greatly increased demand for computing. Having realized the benefits of moving from mainframes to commodity clusters, we find that those clusters are quite expensive to operate. We have low-cost virtualization. And, above all, we have multiple billions of dollars being spent by the likes of Amazon, Google, and Microsoft to create real commercial large-scale systems containing hundreds of thousands of computers. The prospect of needing only a credit card to get on-demand access to 100,000+ computers in tens of data centers distributed throughout the world—resources that be applied to problems with massive, potentially distributed data, is exciting! So we are operating at a different scale, and operating at these new, more massive scales can demand fundamentally different approaches to tackling problems. It also enables—indeed is often only applicable to—entirely new problems.

Nevertheless, yes: the problems are mostly the same in Clouds and Grids. There is a common need to be able to manage large facilities; to define methods by which consumers discover, request, and use resources provided by the central facilities; and to implement the often highly parallel computations that execute on those resources. Details differ, but the two communities are struggling with many of the same issues.

The appearance of cloud computing has been observed very recently as a new promising paradigm that delivers IT services as computing utilities for companies, academic computing and enterprises. It has caused an influence in IT industries. According to IBM, a cloud is a pool of virtualized computer resources that hosts a variety of different workloads and allows them to be deployed and scaled-out through the rapid provisioning of virtual or physical machines; supports redundant, self-recovering, highly scalable programming models and resource usage monitoring in real time to enable rebalancing of allocations when needed.

Several enterprises are deciding to use virtual data centres to facilitate infrastructure managing and trying to decrease the need of hardware maintenance. This kind of cyber infrastructure shrinks the difficulties that are brought in deployment of services. For example, many users may like to deploy applications in a distributed environment, regardless the cost of this deployment and losing flexibility. The cause of designing the Clouds to provide services to users and providers needs to be made up for sharing their capability and resources. Clouds have almost the same vision of grid computing, nevertheless still there are highly considerable differences.

A grid is a system that has the ability to manage and organize resources and services that are

distributed across several control domains, utilize protocols and interfaces and supply high quality of service. Grimshaw et.al. define grid computing as “coordinated resource sharing and problem solving in dynamic, multi-institution virtual organizations”. Cloud Computing has been provided by a number of large organisations such as Amazon, Google, Sun and Yahoo. Individual users through other organisations are also adopted it. Amazon has been a key organisation in the development of cloud computing. Amazon modernised their own internal data centres, which resulted in significant increases in internal efficiency. In 2005 Amazon’s cloud computing system called Amazon web services was born. Amazon was the one of the first organisations to provide cloud-computing facility.

CLOUDS, GRIDS, AND DISTRIBUTED SYSTEMS

Many discerning readers will immediately notice that our definition of Cloud Computing overlaps with many existing technologies, such as Grid Computing, Utility Computing, Services Computing, and distributed computing in general. We argue that Cloud Computing not only overlaps with Grid Computing, it is indeed evolved out of Grid Computing and relies on Grid Computing as its backbone and infrastructure support. The evolution has been a result of a 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). As for Utility Computing, it is not a new paradigm of computing infrastructure; rather, 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. Utility computing is typically implemented using other computing infrastructure (e.g. Grids) with additional accounting and monitoring services.

A Cloud infrastructure can be utilized internally by a company or exposed to the public as utility computing. Web 2.0 covers almost the whole spectrum of service-oriented applications, where Cloud Computing lies at the large-scale side. Supercomputing and Cluster Computing have been more focused on traditional non-service applications. Grid Computing overlaps with all these fields where it is generally considered of lesser scale than supercomputers and Clouds.

Grid Computing aims to “enable resource sharing and coordinated problem solving in dynamic, multi-institutional virtual organizations”. There are also a few key features to this definition: First of all, Grids provide a distributed computing paradigm or infrastructure that spans across multiple virtual organizations (VO) where each VO can consist of

either physically distributed institutions or logically related projects/groups. The goal of such a paradigm is to enable federated resource sharing in dynamic, distributed environments. The approach taken by the *de facto* standard implementation – The Globus Toolkit, is to build a uniform computing environment from diverse resources by defining standard network protocols and providing middleware to mediate access to a wide range of heterogeneous resources.

Globus addresses various issues such as security, resource discovery, resource provisioning and management, job scheduling, monitoring, and data management. Half a decade ago, Ian Foster gave a three point checklist to help define what is, and what is not a *Grid*: 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. Although point 3 holds true for Cloud Computing, neither point 1 nor point 2 is clear that it is the case for today’s Clouds. The vision for Clouds and Grids are similar, details and technologies used may differ, but the two communities are struggling with many of the same issues. This paper strives to compare and contrast Cloud Computing with Grid Computing from various angles and give insights into the essential characteristics of both, with the hope to paint a less cloudy picture of what Clouds are, what kind of applications can Clouds expect to support, and what challenges Clouds are likely to face in the coming years as they gain momentum and adoption. We hope this will help both communities gain deeper understanding of the goals, assumptions, status, and directions, and provide a more detailed view of both technologies to the general audience.

COMPARING GRIDS AND CLOUDS SIDE-BY-SIDE

This section aims to compare Grids and Clouds across a wide variety of perspectives, from architecture, security model, business model, programming model, virtualization, data model, compute model, to provenance and applications. We also outline a number of challenges and opportunities that Grid Computing and Cloud Computing bring to researchers and the IT industry, most common to both, but some are specific to one or the other.

Business Model-

Traditional business model for software has been a one-time payment for unlimited use (usually on 1 computer) of the software. 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 such as electricity, gas, and water, and the model relies on

economies of scale in order to drive prices down for users and profits up for providers. Today, Amazon essentially provides a centralized Cloud consisting of Compute Cloud EC2 and Data Cloud S3. The former is charged based on per instance-hour consumed for each instance type and the latter is charged by per GB-Month of storage used. In addition, data transfer is charged by TB / month data transfer, depending on the source and target of such transfer. The prospect of needing only a credit card to get on-demand access to 100,000+ processors in tens of data centers distributed throughout the world—resources that be applied to problems with massive, potentially distributed data, is exciting!

The business model for Grids (at least that found in academia or government labs) is project-oriented in which the users or community represented by that proposal have certain number of service units (i.e. CPU hours) they can spend. For example, the Tera Grid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computational power. The Tera Grid has more than a dozen Grid sites, all hosted at various institutions around the country.

What makes an institution want to join the Tera Grid? When an institution joins the Tera Grid with a set of resources, it knows that others in the community can now use these resources across the country. It also acknowledges the fact that it gains access to a dozen other Grid sites. This same model has worked rather well for many Grids around the globe, giving institutions incentives to join various Grids for access to additional resources for all the users from the corresponding institution.

There are also endeavors to build a Grid economy for a global Grid infrastructure that supports the trading, negotiation, provisioning, and allocation of resources based on the levels of services provided, risk and cost, and users' preferences; so far, resource exchange (e.g. trade storage for compute cycles), auctions, game theory based resource coordination, virtual currencies, resource brokers and intermediaries, and various other economic models have been proposed and applied in practice.

Architecture-

Grids started off in the mid-90s to address large-scale computation problems using a network of resource-sharing commodity machines that deliver the computation power affordable only by supercomputers and large dedicated clusters at that time. The major motivation was that these high performance computing resources were expensive and hard to get access to, so the starting point was to use federated resources that could comprise compute, storage and network resources from multiple geographically distributed institutions, and such resources are generally heterogeneous and dynamic. Grids focused on integrating existing

resources with their hardware, operating systems, local resource management, and security infrastructure.

In order to support the creation of the so called "Virtual Organizations"—a logical entity within which distributed resources can be discovered and shared as if they were from the same organization, Grids define and provide a set of standard protocols, middleware, toolkits, and services built on top of these protocols. Interoperability and security are the primary concerns for the Grid infrastructure as resources may come from different administrative domains, which have both global and local resource usage policies, different hardware and software configurations and platforms, and vary in availability and capacity.

Grids provide protocols and services at five different layers as identified in the Grid protocol architecture. At the **fabric layer**, Grids provide access to different resource types such as compute, storage and network resource, code repository, etc. Grids usually rely on existing fabric components, for instance, local resource managers (i.e. PBS, Condor, etc). General-purpose components such as GARA (general architecture for advanced reservation), and specialized resource management services such as Falcon (although strictly speaking, Falcon also provides services beyond the fabric layer).

Resource Management-

This section describes the resource management found in Grids and Clouds, covering topics such as the compute model, data model, virtualization, monitoring, and provenance. These topics are extremely important to understand the main challenges that both Grids and Clouds face today, and will have to overcome in the future.

Compute Model: Most Grids use a batch-scheduled compute model, in which a local resource manager (LRM), such as PBS, Condor, SGE manages the compute resources for a Grid site, and users submit batch jobs (via GRAM) to request some resources for some time. Many Grids have policies in place that enforce these batch jobs to identify the user and credentials under which the job will run for accounting and security purposes, the number of processors needed, and the duration of the allocation. For example, a job could say, stage in the input data from a URL to the local storage, run the application for 60 minutes on 100 processors, and stage out the results to some FTP server. The job would wait in the LRM's wait queue until the 100 processors were available for 60 minutes, at which point the 100 processors would be allocated and dedicated to the application for the duration of the job.

Due to the expensive scheduling decisions, data staging in and out, and potentially long queue times, many Grids don't natively support interactive applications; although there are efforts in the Grid community to enable lower latencies to resources via multi-level scheduling, to allow applications with many short-running tasks to execute efficiently on Grids.

Cloud Computing compute model will likely look very different, with resources in the Cloud being shared by all users at the same time (in contrast to dedicated resources governed by a queuing system). This should allow latency sensitive applications to operate natively on Clouds, although ensuring a good enough level of QoS is being delivered to the end users will not be trivial, and will likely be one of the major challenges for Cloud Computing as the Clouds grow in scale, and number of users.

Data Model: While some people boldly predicate that future Internet Computing will be towards Cloud Computing centralized, in which storage, computing, and all kind of other resources will mainly be provisioned by the Cloud, we envision that the next-generation Internet Computing will take the triangle model: Internet Computing will be centralized around Data, Clouding Computing, as well as Client Computing. Cloud Computing and Client Computing will coexist and evolve hand in hand, while data management (mapping, partitioning, querying, movement, caching, replication, etc) will become more and more important for both Cloud Computing and Client Computing with the increase of data-intensive applications.

Programming Model-

Although programming model in Grid environments does not differ fundamentally from traditional parallel and distributed environments, it is obviously complicated by issues such as multiple administrative domains; large variations in resource heterogeneity, stability and performance; exception handling in highly dynamic (in that resources can join and leave pretty much at any time) environments, etc. Grids primarily target large-scale scientific computations, so it must scale to leverage large number/amount of resources, and we would also naturally want to make programs run fast and efficient in Grid environments, and programs also need to finish correctly, so reliability and fault tolerance must be considered.

Application Model-

Grids generally support many different kinds of applications, ranging from high performance computing (HPC) to high throughput computing (HTC). HPC applications are efficient at executing tightly coupled parallel jobs within a particular machine with low-latency interconnects and are generally not executed across a wide area network

Grid; these applications typically use message passing interface (MPI) to achieve the needed inter-process communication. On the other hand, Grids have also seen great success in the execution of more loosely coupled applications that tend to be managed and executed through workflow systems or other sophisticated and complex applications. Related to HTC applications loosely coupled nature, there are other application classes, such as Multiple Program Multiple Data (MPMD), MTC, capacity computing, utility computing, and embarrassingly parallel, each with their own niches. These loosely coupled applications can be composed of many tasks (both independent and dependent tasks) that can be individually scheduled on many different computing resources across multiple administrative boundaries to achieve some larger application goal. Tasks may be small or large, uniprocessor or multiprocessor, compute-intensive or data-intensive. The set of tasks may be static or dynamic, homogeneous or heterogeneous, loosely or tightly coupled. The aggregate number of tasks, quantity of computing, and volumes of data could be small but also extremely large.

Security Model-

Clouds mostly comprise dedicated data centers belonging to the same organization, and within each data center, hardware and software configurations, and supporting platforms are in general more homogeneous as compared with those in Grid environments. Interoperability can become a serious issue for cross-data center, cross-administration domain interactions, imagine running your accounting service in Amazon EC2 while your other business operations on Google infrastructure.

Grids however build on the assumption that resources are heterogeneous and dynamic, and each Grid site may have its own administration domain and operation autonomy. Thus, security has been engineered in the fundamental Grid infrastructure. The key issues considered are: single sign-on, so that users can log on only once and have access to multiple Grid sites, this will also facilitate accounting and auditing; delegation, so that a program can be authorized to access resources on a user's behalf and it can further delegate to other programs; privacy, integrity and segregation, resources belonging to one user cannot be accessed by unauthorized users, and cannot be tampered during transfer; coordinated resource allocation, reservation, and sharing, taking into consideration of both global and local resource usage policies.

The public-key based GSI (Grid Security Infrastructure) protocols are used for authentication, communication protection, and authorization. Furthermore, CAS (Community Authorization Service) is designed for advanced

resource authorization within and across communities. Gruber (A Grid Resource Usage SLA Broker) is an example that has distributed policy enforcement points to enforce both local usage policies and global SLAs (Service Level Agreement), which allows resources at individual sites to be efficiently shared in multi-site, multi-VO environments.

Similarity in Cloud & Grid Computing

As cloud computing represents advancement from grid computing, there are naturally a number of similarities. These are discussed below:

Scalability: Scalability refers to the system's ability to handle increasing amounts of work, or to improve performance. Both grid and cloud computing are scalable, as application instances are load balanced. CPU and network bandwidth is directed on demand. This means that grid and cloud systems have a storage capacity that fluctuates, depending on the number of users, instances and the amount of data transferred at a specific time.

Multitenancy and Multitasking: Multitenancy refers to a situation in which a single instance of software is able to serve multiple clients (tenants). Multitasking refers to when multiple tasks (processes) share processing resources. Multitenancy and multitasking, common to both grid and cloud computing, allow numerous users to perform different tasks and access single or multiple application instances.

DIFFERENCES BETWEEN GRIDS AND CLOUDS

1. Grid computing is better suited for organizations with large amounts of data being requested by a small number of users (or few but large allocation requests), whereas cloud computing is better suited to environments where there are a large number of users requesting small amounts of data (or many but small allocation requests).
2. Grids are well suited for complex scientific work in virtual organizations, explained Wolfgang Gentzsch, who was behind Sun's grid efforts and now sits on the board of directors of the Open Grid Forum and is an advisor to the EU DEISA project. Clouds, on the other hand, are well suited for simple work such as many short-running jobs, he said.
3. Grids require batch job scheduling or sophisticated policies for allocating jobs, while clouds do not.

TOOLS AND SIMULATION ENVIRONMENT FOR CLOUD & GRID COMPUTING

Various tools and products which provide assistance in development applications on cloud & Grid computing are:

1. **Zenoss** a single, integrated product that monitors the entire IT infrastructure. It manages the networks, servers, virtual devices, storage, and cloud deployments.
2. **CloudSim and Cloud Analyst** are important for developers to evaluate the requirements of large-scale cloud applications. It helps developers with insights in how to distribute applications among cloud infrastructures and value added services.
3. **Cloudera**, an open-source Hadoop software framework is increasingly used in cloud computing deployments due to its flexibility with cluster-based, data intensive queries.
4. **Spring Roo** is a next generation rapid application development tool, combined with the power of Google Web Toolkit (GWT) that enables developers to build rich browser apps in enterprise production environments.
5. **Paradyn**, This Grid computing tool supports performance experiment management through techniques for quantitatively comparing several experiments and performance diagnosis based on dynamic instrumentation.
6. **Nimrod-G** uses the Globus middleware services for dynamic resource discovery and dispatching jobs over computational grids. It allows scientists and engineers to model parametric experiments and transparently stage the data and program at remote sites.
7. **Condor-G** represents the work of Globus and Condor projects which enables the utilization of large collections of resources that span across multiple domains as if they all belonged to the user's personal domain.
8. **Globus**, an open source software toolkit that facilitates construction of computational grids and grid based applications, across corporate, institutional and geographic boundaries.

COMPARISON OF RESOURCE ALLOCATION PROCESS IN GRID AND CLOUD TECHNOLOGIES

Nowadays, Cloud and Grid computing are widely used in solving scientific problems; those technologies are based on providing computing services on demand just like conventional power and water National Grids; both technologies were developed with the goal of creating a scalable and powerful virtual computer out of a large collection of homogenous or heterogeneous systems that share various combinations of resources.

Resources allocation, also known as resource management is one of the major fields in both technologies, since it controls the way that resources and services are made available to use by entities like users, applications, or services, and to make sure of the efficient utilization of computing resources and to optimize the performance of the submitted tasks. For instance, a grid manager in one geographical area may have limited access over the system components or there may be a discrepancy of resource availability in the highly distributed cyber physical systems. This discrepancy may lead to severe increase in the energy consumption during the execution of grid applications on one side, and to the realization of tasks submitted by grid users on the other side.

In Cloud Computing, scheduling and resource allocation is a widely considered problem due to its crucial role in the whole of Cloud paradigm. Generally, it controls a number of conflicting objectives; computing resources must be well-managed to prevent overloading and waste of bandwidth, processing unit, memory, etc. This waste relates directly to significant financial loss for large Cloud service providers with regards to energy, operational cost as well as dissatisfaction of the Cloud service user.

Grid and Cloud resources allocation and scheduling deals with a large number of processes (tasks) racing to obtain resources (CPU, memory, storage, network) in order to complete their tasks. This requires an optimal mapping between the competing processes and the underlying computing resources. Moreover, scheduling and resource allocation in Grid Computing is considered as an NP-complete optimization problem. The complexity of this problem depends on the number of the objectives to be minimized such as task inter-relations, makespan, resources utilization and energy consumption, while Cloud resources allocation problem is approved as NP-hard in a strong sense.

Resources Allocation in Grid Technology-

In Grid Resource Allocation, a local resource manager (LRM) manages the computing resources

for a Grid site which contains a pool of resources. When users submit their jobs, these jobs will be considered as batch jobs. Each job needs to be executed by leasing some of the free resources for some time specified by the user. The hosted resources could not be shared with other jobs for that time presents a basic batch scheduler in Grid. Resource Allocation can be classified into three phases:-

Resources discovery represents the first phase, where the Grid scheduler should continuously monitor the status of the participated resources to update its resource availability database to know which resources are free to serve the upcoming requests. This phase is a challenge in its own rights and needs a specific algorithm to cover it. Many algorithms have been proposed to deal with this phase, for example "Scalable Grid Resource Discovery through Distributed Search" has been introduced by Butt et al. and "semantic based scalable decentralized grid resources discovery framework" was proposed by Hassan et al..

The second phase is resources selection, where Grids are usually composed of heterogeneous resources which are completely varied in their number of cores, computation speed and memory size. Therefore, the resource selection phase is focusing on selecting the best resource among the available ones to serve the incoming request. Each request specifies the amount of computing power it needs and for how long.

The third phase is focusing on resources usage; after allocating a suitable resource for the incoming task, the task will then be sent to the selected resource over a wide area network. The resource will be reserved to serve this task for the specified amount of time and it could not be used to host another request during this allocated period.

Resources selection is the critical phase in the whole process. For example, if the incoming requested task needs 3 cores CPU for 3 hours and it was hosted in 4 CPU cores node, one un-needed CPU core would run for 3 hours which affects the total Resource Utilization. This leads to lose much of computation power and energy. , where and represent the number of used processors and total processors on resource respectively.

Resources Allocation in Cloud Technology-

Resource allocation and provisioning is the process of mapping the available resources to different applications of the Cloud over the Internet. It must be managed precisely in a way that prevents overloading and resources wastage (wastage of bandwidth, processing unit, memory, etc.). The service provider is responsible for managing its resources, while the Cloud's users are responsible

for specifying their application requirements and the Service Level Agreement (SLA).

Cloud resource allocation can be identified by four phases:

Phase 1: Virtual Machine (VM) Creation Each VM is created according to user's request, which is usually specified in SLA document. Given that Cloud could accept hundreds of users' requests simultaneously, hundreds of VMs can be created simultaneously too.

Phase 2: VM Deploying This is the process of mapping a VM to a suitable available physical machine. The scheduler should have detailed information about the physical machines: which machines are overloaded and which ones are able to host new VMs. Each VM will be accommodated by one host (Physical Machine (PM)). This process is similar to resource selection in Grid resource allocation with extra difficulties. Many VMs can be deployed to one PM. Therefore, the deploying decision should be handled in a way that maintains good resources utilization and avoid performance degradation.

Phase 3: Task Encapsulation this is the process of encapsulating tasks to be processed into the VM. In reality, the VM may be allocated for a specific user or it can be used for multiple users depending on Cloud service level. User task should be encapsulated in its VM. This commonly happens in the Cloud models of Infrastructures-as-a-Service (IaaS) and Platform-as-a-Service (PaaS). Pre-created VMs can be used to serve user incoming tasks, which are usually taking place in the Software -as-a- Service (SaaS) model. Generally, Cloud applications are executed on VMs. Each application has specified requirements of processing power, and the created VM must provide the processing power to the application.

Phase 4: VM Usage In this phase VM will be running to process the encapsulated tasks. Public, private and hybrid Clouds all have the ability to provide VMs with unlimited scalability. The cost of purchasing, operating and maintaining the physical resources of these Cloud datacenters is enormous. Thus Cloud providers intend to optimize the usage of these resources, through planned allocation of VMs in hosts. Cloud service providers are also keen to maintain a positive client experience. The difficulty of resources allocation problem comes in twofold: maintaining the performance requirement of VMs on one side while reducing the operational cost on the other side.

CONCLUSION

In this paper, we show that Clouds and Grids share a lot commonality in their vision, architecture and technology, but they also differ in various aspects such as security, programming model, business model, compute model, data model, applications,

and abstractions. We also identify challenges and opportunities in both fields. We believe a close comparison such as this can help the two communities understand, share and evolve infrastructure and technology within and across, and accelerate Cloud Computing from early prototypes to production systems.

In conclusion, cloud computing as mentioned above is a new technology of computer network, providing the web services in high quality and lower cost comparing to normal technique. Using cloud computing might contribute to improvement of services in other related technologies specially the previous generations such as Grid computing. Cloud computing is almost certainly set to be developed and become an attractive option for many corporations. The future of computing is set to be successful. It is the next generation and can provide tremendous value to companies. It can help companies achieve more efficient use of their IT hardware and software investments, which in turn can lead the acceleration and adoption of innovations.

REFERENCES

1. A. Al-Shaikh, H. Khattab, A. Sharieh and A. Sleit (2016). "Resource Utilization in Cloud Computing as an Optimization Problem" International Journal of Advanced Computer Science and Applications(Ijacs), 7(6).
2. A. Prathibha, D. B. Latha and D. G. Sumathi (2015). "Issues in adapting cluster, grid and cloud computing for HPC applications," International Journal of Concepts on Computing and Information Technology, vol. 2, no. 1, pp. 12-16.
3. A. Weiss (2007). Computing in the Clouds, netWorker 11 (4) pp. 16-25
4. B. Malet and P. Pietzuch (2010). "Resource allocation across multiple cloud data centres", 8th International Workshop on Middleware for Grids, Clouds and e-Science, Copyright 2010 ACM 978-1-4503-0453-5. DIO:10.1145/1890799.1890804.
5. E. R. Kaur (2015). "A Review of Computing Technologies: Distributed, Utility, Cluster, Grid and Cloud Computing," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 2, pp. 144-148.
6. Foster, Y. Zhao, I. Raicu and S. Lu, (2008). "Cloud Computing and Grid Computing 360-Degree Compared", Proc. IEEE Grid

Computing Environments Workshop, pp. 1-10.

7. H.Stockinger. (2007). "Defining the grid: a snapshot on the current view". The Journal of Supercomputing, (1): pp. 3–17.
8. K. Kaur and A. K. Rai (2014). A Comparative Analysis: Grid, Cluster and Cloud Computing," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, no. 3, pp. 5730-5734.
9. M. Armbrust et. al. (2010). "A View of Cloud Computing", Comm. ACM, vol. 53, no. 4, 2010, pp. 50–58.
10. Members of EGEE-II. An egee comparative study: Grids and clouds - evolution or revolution. Technical report, Enabling Grids for E-science Project, June 2008. Electronic version available at <https://edms.cern.ch/document/925013/>
11. N. Sadashiv (2011). "Cluster, Grid and Cloud Computing: A Detailed Comparison," 6th International Conference on Computer Science & Education (ICCSE), pp: 477 – 482, Singapore,, DOI: 10.1109/ICCSE.2011.6028683.
12. P. Dabas and A. Arya (2013). Grid Computing: An Introduction," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 3, pp. 466-470.

Corresponding Author

Ankit Bansal*

Assistant Professor, Gulzar Group of Institutes,
Ludhiana

erankitbansal@gmail.com