

# Comparison of Cloud and Grid Computing

J. Pourqasem<sup>1</sup>, S. Karimi<sup>1</sup>, S.A. Edalatpanah<sup>2,3,\*</sup>

<sup>1</sup>Department of Computer Science, Faculty of Engineering, University of Guilan, Rasht, Iran

<sup>2</sup>Department of Applied Mathematics, Lahijan Branch, Islamic Azad University, Lahijan, Iran

<sup>3</sup>Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

\*Corresponding author: saedalatpanah@gmail.com

Received October 27, 2013; Revised March 17, 2014; Accepted March 18, 2014

**Abstract** Cloud computing is emerged from internet technology that is distinguished from previous generation of distributed systems i.e. Grid Computing. Grid Computing and Cloud Computing have similarity and difference in basic components. In this paper we present a comparison of Grid Computing and Cloud Computing between principles of them. Furthermore, this paper gives insights into the essential characteristics of both.

**Keywords:** grid computing, cloud computing, comparison

**Cite This Article:** J. Pourqasem, S. Karimi, and S.A. Edalatpanah, "Comparison of Cloud and Grid Computing." *American Journal of Software Engineering*, vol. 2, no. 1 (2014): 8-12. doi: 10.12691/ajse-2-1-2.

## 1. Introduction

Grid computing originated in academia in the mid 1990s with an aim to facilitate users to remotely utilize idle computing power within other computing centers and coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations and in a distributed fashion, which may also involve the aggregation of large-scale cluster computing based systems [1]. With grid computing an organization can transparently integrate, streamline, and share dispersed, heterogeneous pools of hosts, servers, storage systems, data, and networks and sensors into one synergistic system. One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers. General grid middleware and (open) standards allow heterogeneous systems to work collaboratively to deliver the appearance of a large virtual computing environment, while offering a variety of virtual resources. Buyya et al. [2] defined grid as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources.

Cloud computing, is a kind of computing model that came into existence around the end of 2007 [3]. It provides a pool of computing resources which the users can access through Internet. The basic principle of cloud computing is to shift the computing from the local computer into the network [4]. This makes the enterprise use the resource which includes network, server, storage, application, service and so on that is required without huge investment on its purchase, implementation, and maintenance. Resources are requested on-demand without any prior reservation and hence eliminate over provisioning and improve resource utilization. Based on Buyya et al. [5] 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 agreement.

Cloud and grid have same vision; nevertheless still there are highly considerable differences between them. In the literature, only a few comparisons have been appeared in the field of computing.

In this paper we focus on cloud and grid principles and present a comparison between them our comparison involved several aspects such as business model, architecture and compute model and monitoring and security models.

## 2. Grids and Clouds Comparing

In this section, we compare cloud and grid from several aspects that have key role on the performance and reliability of them such as architecture, security model, business model, compute model and monitoring.

### 2.1. Business Model

Traditional business model for software has been a one-time payment for unlimited use (usually on 1 computer) of the software. As a next step in technology, Cloud comes with economy aspects in delivering more abstract resources and services [6,7]. The business model of cloud refers to delivering the services over the Internet via a pay-as-you-go formula. As compared to the cloud-based business model, the business model of grid computing is project-oriented where the organizations that adopt it are sharing the resources in a collaborative manner with other companies that are in the grid model. Grid is a virtual organization with advanced services, where they have direct access to computers, software, data and other resources. This access is possible using condition sharing strategies that realize an organizational process, for users and providers of the resources [8].

Another feature that makes difference between Grid and Cloud is that the access for a resource in Grid could be delayed, because it means that resource was allocated first to other user and the requesting user should wait until the resource will be available again. This delayed allocation doesn't happen in the Cloud environment, users receiving the resources on-demand. Wang and von Laszewski [9] highlight another difference between cloud computing and grid computing, related with the user centric interfaces of cloud computing, that ensure for its users an easy to work environment compared with the Grid resources and services, which are accessed with difficulty by grid users because they are forced to learn new Grid commands and APIs.

## 2.2. Architecture

A Grid architecture (Figure 1) provides an overview of the Grid components, defines the purpose and functions of its components, and indicates how the components interact with one another. The main focus of Grid architecture is on the interoperability and protocols among providers and users of resources in order to establish the sharing relationships.

The Fabric layer comprises the physical resources which are shared within the Grid. According to Foster and Kesselman [10], this includes computational resources, storage systems, network resources, catalogues, software modules, sensors and other system resources.

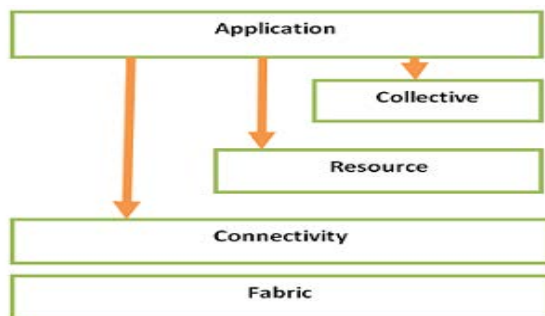


Figure 1. Grid Protocol Architecture

The connectivity layer defines communication and authentication protocols for easy and secure network transactions. This layer enables the exchange of data between resources of fabric layer. The identity of users and resource is verified by authentication protocols which have the following characteristics; single sign on, delegation, integration with various local security solution and user-based trust relationships. The GSI (Grid Security Infrastructure) [11] protocol underlies every Grid transaction.

The Resource layer uses the communication and security protocols (defined by the connectivity layer) to control secure negotiation, initiation, monitoring, accounting, and payment for the sharing of functions of individual resources. It comprises mainly information and management protocols. Information protocols are used to obtain information about the structure and state of available resources. Management protocols are used to negotiate access to resources and serve as a “policy application point” by ensuring that the usage of the resources is consistent with the policy under which the resource is to be shared. The GRAM (Grid Resource Access and Management) [11,12] protocol is used for

allocation of computational resources and for monitoring and control of computation on those resources, and Grid FTP [13] for data access and high-speed data transfer.

The Collective layer is responsible for all global resource management and for interaction with collections of resources (Foster and Kesselman [10]) [14]. Collective layer protocols implement a wide variety of sharing behaviors. The most important functionalities of this layer are: directory services like MDS (Monitoring and Discovery Service) [15] allows for the monitoring and discovery of VO resources, collocation, scheduling and brokering services, monitoring and diagnostics services and data replication services. The services of the collective layer are usually invoked by programming models and tools: Grid-enabled programming systems, workflow systems, software discovery services and collaboration services. This layer also addresses community authorization together with accounting and payment services.

The application layer comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments. Two examples are Grid workflow systems, and Grid portals [16].

Now, we define four-layer architecture for Cloud Computing in Figure 2 which is composed of fabric, unified resource, platform, and application layers [17].

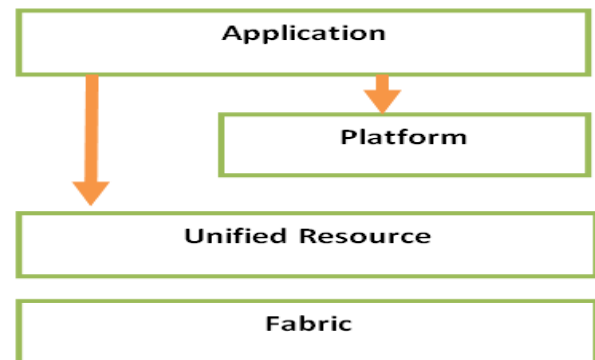


Figure 2. Cloud Protocol Architecture

The fabric layer consists of the raw hardware resources, such as the basic computing units, storage disks, and network bandwidths. Similar to grid computing, at this layer, most resources are heterogeneous. For example, in a cloud data center, the underlying physical machines can be commodity PCs, workstations, and supercomputers. The unified resource layer consists of heterogeneous resources which are usually in the form of virtualized resources. In this layer, the underlying physical machines have been abstracted/encapsulated usually by virtualization tools so that they can be exposed to upper layer and end users as integrated resources, for example, a virtual computer/cluster, a logical file system, a database system, and so on. The platform layer consists of a set of resource management tools and middleware services on top of the unified resources. The platform layer can provide a development and/or deployment platform, for example, a web hosting environment, a workflow modeling service, and a scheduling service, and so on. Finally, the application layer consists of the user applications which can be any kind of applications such as cloud workflow applications, social networking tools, and e-commerce websites.

### 2.3. Compute Model

Most Grids use a batch-scheduled compute model, in which a local resource manager (LRM) [18], such as PBS, Condor, SGE; manages the computer resources for a Grid site, and users submit batch jobs (via GRAM) to request some resources for some time.

Here, the scheduling software identifies a processor on which to run a specific grid job that has been submitted by a user. After available resources have been identified, the follow-on step is to schedule the individual jobs to run on these resources. Advanced schedulers monitor the progress of active jobs, managing the overall workflow. If the jobs were to become lost due to system or network outages, a high-end scheduler would automatically resubmit the job elsewhere. Generally grids have policies associated with the users to enforce security and accounting purposes. These policies actually help to identify user's needs, decide the number of processors, duration of allocation of the processors for the batch jobs submitted by the users [19]. Some grids also have a *reservation system*. These systems allow one to reserve resources on the grid. These are a calendar-based mechanism for reserving resources for specific time periods, and preventing others from reserving the same resource at the same time. In a *scavenging* grid environment, any processor that becomes idle reports its idle status to the grid management node. The management node in turn assigns to this processor the next job that is satisfied by the processor's resources. If the processor becomes busy with local nongrid work, the grid job is usually suspended or delayed.

Due to the expensive scheduling decisions, data staging and potentially long queue times; many Grids don't encourage interactive applications. Although, interactive applications are not impossible for Grids. Compared to Grid, Cloud Computing compute model supports many operations effectively which are not suitable for Grids. Like, Cloud Computing can effectively handle latency sensitive interactive applications to operate natively on Clouds with QoS assurance.

Each service provider of cloud registers its services to an UDDI [20] which in turn is used by service consumers for searching specific services. Once such a service is found its interface can be used for submitting tasks and for retrieving their results. The scheduling heuristic will select the service which is most likely to minimize the execution costs. Usually the negotiation is linked to the phase of service selection and involves a scheduler request for a particular service characteristic. Negotiation can also involve the decision on what data/tasks are allowed to be submitted to the service and whether the service provider can further use the submitted data/tasks for its own purposes or not. As a result it is said that scheduling between services is accomplished by a meta-scheduler [21] that deals with tasks at service level, leaving the resource level scheduling to the internal Virtual Organization (VO) schedulers. These internal schedulers handle tasks assignments depending on their own policies and thus there is no guarantee that the task submitted by the meta-scheduler will be executed inside the cost constraints negotiated at the time of the submission.

Many SAs require some sort of user estimates in order to provide improved scheduling solutions. User given

estimates are dependent on the user's prior experience with executing similar tasks. Users also tend to overestimate task execution times knowing that schedulers rely on them.

### 2.4. Monitoring

Grid monitoring is a complex task, since the nature of the Grid means heterogeneous systems and resources the users want to monitor their applications running on the grid. Also, the resource or service owners need to surface certain states so that the user of those resources or services may manage the usage using the state information. It is to be expected that the user should be able to query the grid system to determine how the application (and possibly the constituent micro jobs) are progressing toward task completion. MDS [22] provides a uniform framework for providing and accessing system configuration and status information such as server configuration, network status, or the locations of replicated datasets.

Grid monitoring supports the discovery of *sensors* in a distributed environment, the collection and analysis of information from these sensors, the generation of alerts when unusual conditions are detected, and so forth. Many Grids (such as Tera Grid [23]) also enforce restrictions on what kind of sensors or long-running services a user can launch. A grid system, particularly the job scheduler, needs to provide automatic monitoring and recovery of micro jobs that happen to fail; failures may be due to programming errors (including infinite loops) or hardware failures of various kinds in the grid resources and/or networking system.

Cloud monitoring is not as straightforward as in Grids, because Grids in general have a different trust model in which users via their identity delegation can access and browse resources at different Grid sites, and 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 her own monitoring infrastructure, and the limited information returned to the user may not provide the necessary level of details for her to figure out what the resource status is. The same problems potentially exist for Cloud developers and administrators, as the abstract/unified resources usually go through virtualization and some other level of encapsulation, and tracking the issues down the software/hardware stack might be more difficult. Essentially monitoring in clouds requires a fine balance of business application monitoring, enterprise server management, virtual machine monitoring, and hardware maintenance, and will be a significant challenge for Cloud Computing as it sees wider adoption and deployments.

### 2.5. Security Models

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 individual resources of users 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 grid security infrastructure (GSI) supports uniform authentication, authorization, and message-protection mechanisms in multi institutional settings. Specifically, the grid security infrastructure provides single sign-on, delegation, and identity mapping using public key technology (X. 509 certificates). A certificate authority (CA) can be utilized to establish the identity of the *donor* processor, as well as the users and the grid itself. Some grid systems provide their own log-in to the grid, whereas other grid systems depend on the native operating systems for user authentication.

CAS [24] (Community Authorization Service) is designed for advanced resource authorization within and across communities. Gruber (A Grid Resource Usage SLA Broker) [25] 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.

Currently, the security model for clouds seems to be relatively simpler and less secure than the security model adopted by Grids. Cloud infrastructure typically rely on web forms (over SSL) to create and manage account information for end-users, and allows users to reset their passwords and receive new passwords via emails in an unsafe and unencrypted communication. Note that new users could use clouds relatively easily and almost instantly, with a credit card and/or email address.

The usage of Clouds is associated also with higher security and privacy risks related to data storage and management in two ways: first because of the need to transfer data back and forth to a Cloud so that it can be processed in a Cloud; second because data is stored on an external infrastructure and the data owner relies on the Cloud provider's assurance that no unauthorized access takes place. A typical Cloud is outside a company's or other organization's firewall, which have significant impact on a company's decision to move use Cloud Services.

Legal and regulatory issues also need attention. When data are moved into the Cloud, providers may choose to locate them anywhere on the planet. The physical location of data centers determines the set of laws that can be applied to the management of data. For example, specific cryptography techniques could not be used because they are not allowed in some countries.

We outline several risks which a Cloud user should raise with vendors before committing [26]:

A. Privileged user access: sensitive data processed outside the enterprise needs the assurance that they are only accessible and propagated to privileged users.

B. Regulatory compliance: a customer needs to verify if a Cloud provider has external audits and security certifications and if their infrastructure complies with some regulatory security requirements.

C. Data location: since a customer will not know where her data will be stored, it is important that the Cloud provider commit to storing and processing data in specific jurisdictions and to obey local privacy requirements on behalf of the customer.

D. Data segregation: one needs to ensure that one customer's data is fully segregated from another customer's data.

E. Recovery: it is important that the Cloud provider has an efficient replication and recovery mechanism to restore data if a disaster occurs.

F. Long-term viability: your data should be viable even the Cloud provider is acquired by another company.

### 3. Conclusion

In this paper, we have presented a comparison between principle components of two computing models, grid and cloud computing. Such a comparison in different perspectives will make easy to understand the computing models since the features of these computing models seems to be similar conceptually. It also helps in identifying the similarities and differences from each other. Grid and cloud computing appears to be a promising model especially focusing on standardizing APIs, security, interoperability, new business models, and dynamic pricing systems for complex services. Hence there is a scope for further research in these areas.

### References

- [1] A Iosup, D Epema, "grid computing workloads", internet computing, (volume: 15, issue: 2, page: 19-26), IEEE, 2011.
- [2] Michael Hauck, Matthias Huber, Markus Klems, Samuel Kounev, Alexander Pretschner, Ralf Reussner, Stefan Tai, "challenges and opportunities of cloud computing", Karlsruhe Institute of technology, technical report, (page: 1-11), 2010.
- [3] Lizhe Wang, Gregor von Laszewski, Andrew Younge, Xi He, Marcel Kunze, Jie Tao, Cheng Fu, "cloud computing: a perspective study", generation computing, (volume 28, issue 2, page: 137-146), 2010.
- [4] N Sadashiv, SMD Kumar, "cluster, grid and cloud computing: A detailed comparison", computer science & education (ICCSE), 6th international conference on, (page: 477-482), 2011.
- [5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility, future gener", computer systems, (volume 25, issue 6, page: 599-616), 2009.
- [6] I. Foster, Yong Zhao, Raicu. I. Shiyong Lu, "cloud computing and grid computing 360-degree compared", grid computing environments workshop, GCE '08. (Page: 1-10), 2008.
- [7] Thomas Rings, Geoff Caryer, Julian Gallop, Jens Grabowski, Tatiana Kovacicova, Stephan Schulz, Ian Stokes-Rees, "grid and cloud computing: opportunities for Integration with the next generation network", Journal of grid computing, (volume 7, issue 3, page: 375-393), 2009.
- [8] A. Szalay, A. Bunn, J. Gray, I. Foster, I. Raicu, "the importance of data locality in distributed computing applications", NSF workflow workshop, 2006.
- [9] Lizhe Wang, Gregor von Laszewski, Andrew Younge, Xi He, Marcel Kunze, Jie Tao, Cheng Fu, "cloud computing: a perspective study", new generation computing (volume 28, issue 2, page: 137-146), 2010.

- [10] Foster, Ian, Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", 2 end edition, Morgan Kaufmann Publishers, 2004.
- [11] David Villegas, Ivan Rodero, Liana Fong, Norman Bobroff, Yanbin Liu, Manish Parashar, S. Masoud Sadjadi," handbook of cloud computing; the role of grid computing technologies in cloud computing", springr, (page: 183-218), 2010.
- [12] I. Foster, "globus toolkit version 4: software for service oriented systems." journal of computer science and technology, (volume 21, issue 4, page: 513-520), 2006.
- [13] Sunita Rani, P.K. Suri, "grid computing and cloud computing: description and comparison", international journal of advanced research in computer science and electronics engineering (IJARCSEE), (volume 1, issue 7, page: 136-150), 2012.
- [14] Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, "grid and cloud computing: a Business perspective on technology and applications", springer, 2010.
- [15] A Goscinski, M Brock, "toward dynamic and attribute based publication, discovery and selection for cloud computing", future generation computer systems, (volume 26, issue 7, page: 947-970), 2010.
- [16] Y. Zhao, M. Wilde, I. Foster, J. Voeckler, J. Dobson, E. Gilbert, T. Jordan, E. Quigg., "virtual data grid middleware services for data-intensive science", concurrency and computation: practice and experience, (volume 18, issue 6, page: 598-608), 2006.
- [17] Xiao Liu, Dong Yuan, Gaofeng Zhang, Wenhao Li, Dahai Cao, Qiang He, Jinjun Chen, Yun Yang," the design of cloud workflow systems", springer, 2012.
- [18] R Giordanelli, C Mastroianni, "the cloud computing paradigm: characteristics, opportunities and research issues", istituto di calcolo e reti ad alte. grid. deis. unical. It, (page: 1-23), 2010.
- [19] N. Wilkins-Diehr, "science gateways - common community interfaces to grid resources", concurrency and computation: practice and experience, (volume 19, issue 6, page: 743-749), 2007.
- [20] Iosup. A, Epema. D, "grid computing workloads", internet computing, (volume: 15, issue: 2 page: 19-26), IEEE, 2011.
- [21] Borko Furht, Armando Escalante, handbook of cloud computing: scheduling service oriented workflows inside clouds using an adaptive agent based approach", springer, 2010.
- [22] X Meng, Y Wang, W Zhang, J Meng, automatic discovery of data resources in the e-government grid", WSEAS transactions on computers,(volume 8, issue 3, page: 569-578), 2009.
- [23] Daniel S. Katz, Scott Callaghan, Robert Harkness, Sudhakar Pamidighantam, Marlon Pierce, Beth Plale, Carol Song, John Towns, "science on the teragrid", computational methods in science and technology, (page: 81-97), 2010.
- [24] Yongqiang Zou, Li Zha, Xiaoning Wang, Haojie Zhou, Peixu Li, "a layered virtual organization architecture for grid", the journal of super computing, (volume 51, issue 3, page: 333-351), 2010.
- [25] Jiangyan Zheng, Shijie Cao, Jianhui Zheng, Zhihui Chen, Liang Hong, "information technology education based on cloud computing", information computing and applications communications in computer and information science, (volume 391, page: 417-426), 2013.
- [26] S. Subashini, V. Kavitha, "a survey on security issues in service delivery models of cloud computing", journal of network and computer applications, (volume 34, issue 1, page: 1-11), 2011.