International Journal of Engineering, Science and Metallurgy

(General article)

Comparable Study of Grid Computing and Cloud Computing

Deepak Dudeja[a], Sunita Allen[a]

[a] *Assistant Professor, Department of CSE, Technology Education & Research Integrated Institutions, Kurukshetra, Haryana*

Received 11 Nov. 2011; accepted 15 Nov.2011, Available online 1 Dec. 2011

Abstract

This paper 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. [FZRL] 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.

Keywords: *Grid computing, Cloud computing*

1.1. Business Model

In a Cloud Base 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 S2. The former is charged based on per instance-hour consumed for each instance type and the later 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 business model for Grids is project oriented in which the users or community represented by that proposal have certain number of service units they can spend. For example, Tera Grid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computation power. The Tera Grid has more than a dozen Grid sites, all hosted at various institutions around the country [1].

1.2. Architecture

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 form 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 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. Grid provides 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 a mangers.

The connectivity layer defines core communication and authentication protocols for easy and secure network transactions. The Grid Security Infrastructure (GSI) protocol underlies every Grid transaction.

The resource layer defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The Grid Resource Access and Management (GRAM) protocol is used for allocation of computational resources and for monitoring and control of computation on those resources, and Grid FTP for data access and high-speed data transfer. The collective layer captures interactions across collections of resources, directory services such as Monitoring and Discovery Services (MDS) allows for the monitoring and discovery of VO resources, Condor-G and Nimrod-G are examples of co-allocating, scheduling and brokering services and MPIC for Grid enabled programming systems and

---

[*] Corresponding Author
Email: deepakdudeja.cse.teri@terii.in

community authorization service for global resource policies [2].

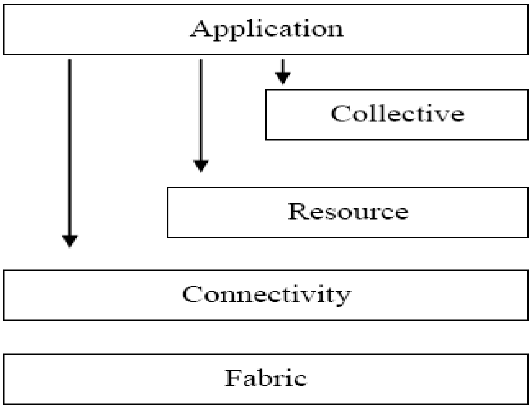

Figure1 Grid Protocol Architecture

The application layer comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments for example Grid workflow systems.

Clouds are developed to address Internet-scale computing problems where some assumptions are different from those of the Grids. Clouds are usually referred to as a large pool of computing and/or storage resources, which can be accessed via standard protocols via an abstract interface. Clouds can be built on top of many existing protocols such as Web Services and some advanced Web 2.0 technologies such as REST, AJAX, RSS, etc. In fact behind the cover, it is possible for Clouds to be implemented over existing Grid technologies leveraging more than a decade of community efforts in standardization, security, resource management, and virtualization support.

There are also multiple versions of definition for Cloud architecture; we define a four-layer architecture for Cloud Computing in comparison to the Grid architecture, composed of fabric, unified resource, platform and application layers.

The fabric layer contains the raw hardware level resources, such as to compute resources, storage resources, and network resources. The unified resource layer contain resources that have been abstracted/encapsulated so that they can be exposed to upper layer and end users as integrated resources, for instance, a virtual computer/cluster, a logical file system, a database system, etc. The platform layer adds on a collection of specialized tools, middleware and services on top the unified resources to provide a development and/or deployment platform. for instance, a Web hosting environment, a scheduling service, etc. Finally, the application layer contains the applications that would run in the Clouds.

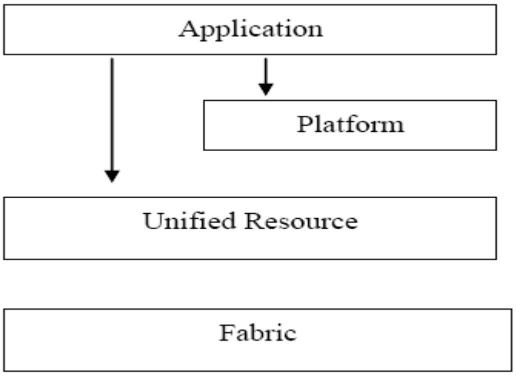

Figure 2 Cloud Architecture

2. Resource Management

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 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 [3].

Cloud Computing compute model will likely look very different, with resources in the Cloud being shared by all users at the same time. This should allow latency sensitive applications to generate 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 the major challenges for Cloud Computing as the Clouds grow in scale, and number of users.

2.1 Data Model

The 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 shown in figure 3. Internet Computing will be centralized around data. Cloud Computing, as well as Client Computing. Cloud Computing and Client Computing will coexist and evolve hand in hand, while data management will become more important for both. Cloud Computing and Client Computing with the increase of data-intensive applications. The critical role of Cloud Computing goes without saying, but the important of Client Computing cannot be overlooked [4] either for several reasons:

i) For security reasons, people might not be willing to run mission-critical applications on the Cloud and send sensitive data to the Cloud for processing and storage;

ii) Users want to get their things done even when the Internet and Cloud are down or the network communication is slow;

iii) With the advances of multi-core technology, the coming decade will bring the possibilities of having a desktop supercomputer with 100s to 1000s of hardware threads/cores.

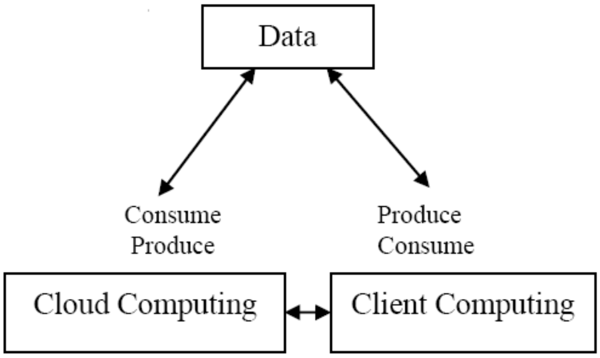

Figure 3 The Triangle model of next-generation Internet Computing

Data Grids have been specifically designed to tackle data intensive applications in Grid environments, with the concept of virtual data playing crucial role. Virtual data captures the relationship between data, programs and computations and prescribes various abstractions that a data grid can provide: location transparency where data can be requested without regard to data location, a distributed metadata catalog is engaged to keep track of the locations of each piece of data across grid sites, and privacy and access control are enforced; 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. There is also 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.

2.2 Data Locality

In Grids, data storage usually relies on shared file systems, where data locality cannot be easily applied. One approach is to improve schedulers to be data-aware, and to be able to leverage data locality information when scheduling computational tasks; this approach has shown to improve job turn-around time significantly.

To achieve good scalability at Internet scales for Clouds, Grids and their applications, data must be distributed over many computers and computations must be steered towards the best place to execute in order to minimize the communication costs.

2.3 Virtualization

Virtualization has become an indispensable ingredient for almost every Cloud; the most obvious reasons are for abstraction and encapsulation. Just like threads were introduced to provide users the "illusion" as if the computer were running all the threads simultaneously, and each thread were using all the available resources, Clouds need to run multiple user applications, and all the applications appear to the uses if they were running simultaneously and could use all the available resources in the Cloud. Virtualization provides the necessary abstraction such that the underlying fabric can be unified as a pool of resources and resource overlays can be built on top of them. Virtualization also enables each application to be encapsulated such that they can be configured, deployed, started, migrated, suspended, resumed, stopped, etc., and thus provides better security, manageability and isolation [5].

There are also many other reasons that Clouds tend to adopt virtualization:

a) Server and application consolidation, as multiple applications can be run on the same server; resources can be utilized more efficiently;

b) Configurability, as the resource requirements for various applications could differ significantly, some require large storage, some compute, in order to dynamically configure and bundle resources for various needs, virtualization is necessary as this is not achievable at the hardware level;

c) Increased application availability, virtualization allows quick recovery from unplanned outages, as virtual environments can be backed up and migrated with no interruption in service;

d) Improved responsiveness: resource provisioning, monitoring and maintenance can be automated, and common resources can be cached and reused.

Grids do not rely on virtualization as much as Clouds do, but that might be more due to policy and having each individual organization maintain full control of their resources. However, there are efforts in Grids to use Virtualization as well, such as Nimbus (Virtual Workspace Service), which provide the same abstraction and dynamic deployment capabilities.

2.4 Monitoring

Many Grids also enforce restrictions on what kind of sensors or long-running services a user can launch, 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 Cloulds. In a Cloud, different levels of services can be offered to an end use, the user is only exposed to a predefined API, and the lower level resources are opaque to the user. The user does not have the liberty to deploy his own monitoring infrastructure, and the limited information returned to the user may not provide the necessary level of details for him to figure out what the resource status is [5].

2.5 Provenance

Provenance refers to the derivation history of a data product, including all the data sources, intermediate data products, and the procedures that were applied to produce the data product. In Grids, provenance management has begun general built into a workflow system, from early pioneers such as Chimera, to modern scientific workflow systems, such as Swift, Kepler, and VIEW to support the discovery and reproducibility of scientific results. It has also been built as a standalone service, such as PreServ, to facilitate the integration of provenance component in more general computing models, and deal with trust issues in provenance assertion.

Clouds are become the future playground for e-science research, and provenance management is extremely important in order to track the processes and support the reproducibility of scientific results. Provenance is still an unexplored area in Cloud environments, in which we need to deal with even more challenging issues such as tracking data production across different service providers.

In other words, capturing and managing provenance in Cloud environments may prove to be more difficult than in Grids, since in the latter there are already a few provenance systems and initiatives, however scalable provenance querying and secure access of provenance information are still open problems for both Grids and Cloud environments.

2.6 Programming Model

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 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. Clouds have generally adopted Web Services APIs where users access, configure and program Cloud services using pre-defined APIs exposed as Web Services, and HTTP and SOAP are the common protocols chosen for such services. Although Clouds adopted some common communication protocols such as HTTP and SOAP, the integration and interoperability of all the services and applications remain the biggest challenge [6], as users need to tap into a federation of Clouds instead of a single Cloud provider.

3. Application Model

Grids have 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 [7].

Cloud Computing could in principle caters to a similar set of applications. the one exception that will likely be hard to achieve in Cloud Computing are HPC applications that require fast and low latency network interconnects for efficient scaling to many processors as Cloud Computing is still in its infancy, the applications that will run on clouds are not well defined, but we can certainly characterize them to be loosely coupled, transaction oriented and likely to be interactive.

4. 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.

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 [8], 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. Currently the security model for Clouds seems to be relatively simpler and less secure than the security model adopted by Grids. Clouds infrastructure typically rely on Web forms 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. To contrast this [9], Grids are stricter about its security. Security is one of the largest concerns for the adoption of Cloud Computing. We outline seven risks a Cloud user should raise with vendors before committing:

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 his data will be stored, it is important that the Cloud provider commit to store and processing data in specific jurisdictions and to obey local privacy requirements on behalf of the customer [10].

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) Investigative support: Cloud services are especially difficult to investigate, it this is important for a customer, and then such support needs to be ensured with a contractual commitment.

g) Long-term viability: Your data should be viable even the Cloud provider is acquired by another company.

5. Job Scheduling

Job Scheduling is the core value and aim of grid technology, its aim is to use all kinds of resources. It can divide a huge task into a lot of independent and no related sub tasks, and then let every node do the jobs. Even any node fails and doesn't return result, it doesn't matter; the whole process will not be affected. Even one node crashes, the task will be reassigned to other nodes. Just like Grid Computing, Cloud Computing will make a huge resource pool through grouping all the resources. But the resources provided by cloud are to complete a special task. Cloud Computing is designed to meet general application and there are not grid for a special field.

Conclusion

The main aim of this work is to present a difference between Grid Computing and Cloud Computing. Cloud computing has many advantages over Grid Computing, clouds will not replace grids, as grids have not replaced capability HPC, over the last 10 years as some have predicated. All three technologies have their place, what we will see over the next couple of years is that these different computing nodes will more and more grow together with the WWW and the Internet, until all these resources become one global infrastructure for information, Knowledge, computation and communication, the WWW. We think it is more likely that grids will be re-branded or merge into cloud computing, Grid Computing helped create a certain technology reality which made clouds possible. And when it comes to IaaS (infrastructure as s service), We think in five years something like 80 to 90 percent of the computation are doing could be cloud-based. In a word, the concept of Cloud Computing is becoming more and more popular. Now, Cloud Computing is in the beginning stage. All kinds of companies are providing all kinds of Cloud computing service, from software application to net storage and mail filter. We believe cloud computing will become main technology in our information life. Cloud has owned all conditions. Now the dream of Grid Computing will be realized by Cloud Computing. It will be a great event in the IT history [11].

A Comparison such as this helped the two communities understand, share and evolve infrastructure and technology within and across and accelerate Cloud Computing from early prototypes to production systems.

References

1. S. Ahuja, N. Carriero, and D. Gelernter (1986). Linda and Friend, *IEEE Computer* 19 (8), pp. 26-34.
2. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster,C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke (2002), Data Management and Transfer in High PerformanceComputational Grid Environments, *Parallel Computing Journal*, Vol. 28 (5), pp. 749-771.
3. Amazon Elastic Compute Cloud (Amazon EC2) (2008), http://aws.amazon.com/ec2.
4. Amazon Simple Storage Service (Amazon S3)(2008), http://aws.amazon.com/s3.
5. B. Bode, D.M. Halstead, R. Kendall, Z. Lei, W. Hall, D. Jackson (2000).The Portable Batch Scheduler and the Maui Scheduler on Linux Cluster, *Usenix, 4th Annual LinuxShowcase & Conference*.
6. J. Brodkin. Gartner: Seven cloud-computing security risks",http://www.networkworld.com/news/2008/070208-cloud.html.

7. R. Buyya, D. Abramson, J. Giddy, Nimrod/G (2000), 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).
8. R. Buyya, K. Bubendorfer (2008), Market Oriented Grid and Utility Computing, *Wiley Press*, New York, USA.
9. A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi (2007),Storing and Querying Scientific Workflow Provenance Metadata Using an RDBM, *Proc. of the 2nd International Workshop on Scientific Workflows and Business Workflow Standards in e- Science, in conjunction with e-Science,* Bangalore, India, December, pp.611-618.
10. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke (2001), The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets, *Jrnl. of Network and Computer Applications*, 23, pp.187-200.
11. K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke, (2004), From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring &Evolution.