International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 5, July - 2012

Cloud & Grid Computing Environments

Balachandrudu K E
Professor, HOD-CSE,
PRRM Engineering College,
JNTUH, SHABAD, R.R Dist

Abstract– 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. This paper strives to compare and contrast Cloud Computing with Grid Computing from various angles and give insights into the essential characteristics of both.

Overview

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 (TeraGrid, 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.

1.1 Defining Cloud Computing

There is little consensus on how to define the Cloud. We add yet another definition to the already saturated list of definitions for Cloud Computing: *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.*

There are a few key points in this definition. First, Cloud Computing is a specialized distributed computing paradigm; it differs from traditional ones in that,

1) it is massively scalable,

2) Can be encapsulated as an abstract entity that delivers different levels of services to customers outside the Cloud,

3) It is driven by economies of scale, and

4) The services can be dynamically configured (via virtualization or other approaches) and delivered on demand. Governments, research infrastructure.

There are three main factors contributing to the surge and interests in Cloud Computing:
1) Rapid decrease in hardware cost and increase in computing power and storage capacity, and
2)the advent of multi-core architecture and modern supercomputers consisting of hundreds of thousands of cores; instrumentation/simulation and Internet publishing and archiving; and
3) the wide-spread adoption of Services Computing and Web 2.0 applications.

1.2 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.

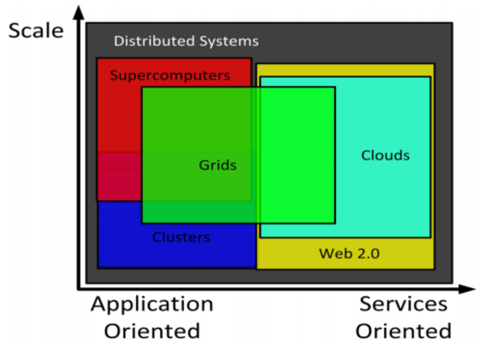

Figure 1: Grids and Clouds Overview

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.

2 Comparing Grids and Clouds Side-by-Side

This section aims to compare Grids and Clouds across a wide variety of perspectives, The Globes 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. Globes address 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 not a Grid is: 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.

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. 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 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 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 TeraGrid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computational power.

2.2 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.
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 General-purpose components such as GARA (general architecture for advanced reservation), and specialized resource management services such as Falkon (although strictly speaking, Falkon also provides services beyond the fabric layer).

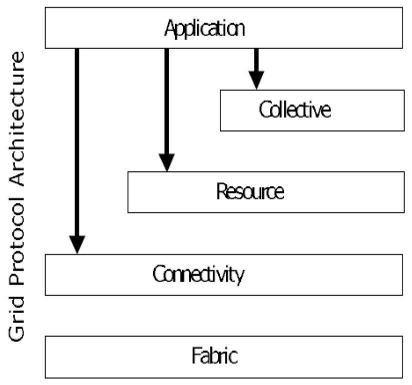

Figure 2: Grid Protocol Architecture

Scheduling and brokering services and MPICH for Grid enabled programming systems and CAS (community authorization service) for global resource policies. The ***connectivity layer*** defines core communication and authentication protocols for easy and secure network transactions. The GSI (Grid Security Infrastructure) 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 GRAM (Grid Resource Access and Management) protocol is used for allocation of computational resources and for monitoring and control of computation on those resources, and GridFTP for data access and high-speed data transfer. The ***collective layer*** captures interactions across collections of resources, directory services such as MDS (Monitoring and Discovery Service) allows for the monitoring and discovery of VO resources, Condor-G and Nimrod-G are examples of co-allocating, ***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 (i.e. Quark Net e-learning environment [52], National Virtual Observatory (http://www.us-vo.org), TeraGrid Science gateway (http://www.teragrid.org)). 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 (WSDL, SOAP), and some advanced Web 2.0 technologies such as REST, RSS, AJAX, 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, a four-layer architecture for Cloud Computing in comparison to the Grid architecture, 1) fabric, 2) unified resource, 3) platform, and 4) application Layers.

The *fabric layer* contains the raw hardware level resources, such as compute resources, storage resources, and network resources. The *unified resource layer* contains resources that have been abstracted/encapsulated (usually by virtualization) 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 of 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. Clouds in general provide services at three different levels

(*IaaS*, *PaaS*, and *Saas*) as follows, although some providers can choose to expose services at more than one level

Infrastructure as a Service (IaaS) provisions hardware, software, and equipments (mostly at the unified resource layer, but can also include part of the fabric layer) to deliver software application environments with a resource usage-based pricing model. Infrastructure can scale up and down dynamically based on application resource needs. Typical examples are Amazon EC2 (Elastic Cloud Computing) Service and S3 (Simple Storage Service) where compute and storage infrastructures are open to public access with a utility pricing model; Eucalyptus is an open source Cloud implementation that provides a compatible interface to Amazon's EC2, and allows people to set up a Cloud infrastructure at premise and experiment prior to buying commercial services.

Platform as a Service (PaaS) offers a high-level integrated environment to build, test, and deploy custom applications. Generally, developers will need to accept some restrictions on the type of software they can write in exchange for built-in application Engine, which enables users to build Web applications on the same scalable systems that power Google applications.

Software as a Service (SaaS) delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. Sales force is an industry leader in providing online CRM (Customer Relationship Management) Services.

2.3 Resource Management

This section describes the resource management found in Grids and Clouds, covering topics such as the computer 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.

2.4 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. We briefly discuss here some general programming models in Grids. MPI (Message Passing Interface) is the most commonly used programming model in parallel computing, in which a set of tasks use their own local memory during computation and communicate by sending and receiving messages. MPICH-G2 is a Grid enabled implementation of MPI.

2.5 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.

2.6 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.

3 Conclusions and lights to the future

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. What does the future hold? We will hazard a few predictions, based on our beliefs that the economics of computing will look more and more like those of energy. Neither the energy nor the computing grids of tomorrow will look like yesterday's electric power grid. "Cloud" or "Grid", we will need to support on-demand provisioning and configuration of integrated "virtual systems" providing the precise capabilities needed by an end-user. We will need to define protocols that allow users and service providers to discover and hand off demands to other providers, to monitor and manage their reservations, and arrange payment. We will need tools for managing both the underlying resources and the resulting distributed computations. We will need the centralized scale of today's Cloud utilities, and the distribution and interoperability of today's Grid facilities. Unfortunately, at least to date, the methods used to achieve these goals in today's commercial clouds have not been open and general purpose, but instead been mostly Company.

4 References

[1] S. Ahuja, N. Carriero, and D. Gelernter. "Linda and Friends", IEEE Computer 19 (8), 1986, 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. "Data Management and Transfer in High Performance Computational Grid Environments", Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.

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

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

[5] B. Bode, D.M. Halstead, R. Kendall, Z. Lei, W. Hall, D. Jackson. "The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters", Usenix, 4th Annual Linux Showcase & Conference, 2000.

[6] J. Brodkin. "Gartner: Seven cloud-computing security risks", http://www.networkworld.com/news/2008/070208-cloud.html, 2008.

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

[8] R. Buyya, K. Bubendorfer. "Market Oriented Grid and Utility Computing", Wiley Press, New York, USA, 2008.

[9] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi, "Storing and Querying Scientific Workflow Provenance Metadata Using an RDBMS", in Proc. of the 2nd International

Workshop on Scientific Workflows and Business Workflow Standards in e- Science, in conjunction with e-Science, pp.611-618, Bangalore, India, December, 2007.
[10] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke.“The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets”, Jrnl. of Network and Computer Applications, 23:187-200, 2001.
[11] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke. “From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution,” March 5, 2004.
[12] S. Davidson, S. Boulakia, A. Eyal, B. Ludäscher, T. McPhillips, S. Bowers, M. Anand, J. Freire, “Provenance in Scientific Workflow Systems”, IEEE Data Eng. Bull. 30(4): 44-50 (2007).
[13] J. Dean, S. Ghemawat. “Map Reduce: Simplified Data Processing on Large Clusters”, Sixth Symposium on Operating System Design and Implementation (OSDI04), 2004.
[14] C. Dumitrescu, I. Raicu, I. Foster. “The Design, Usage, and Performance of GRUBER: A Grid uSLA-based Brokering Infrastructure”, Intl. Journal of Grid Computing, 2007.
[15]Eucalyptus, http://eucalyptus.cs.ucsb.edu/, 2008.
[16] I. Foster, C. Kesselman. “Globes: A Metacomputing Infrastructure Toolkit”, Intl J. Supercomputer Applications, 11(2):115-128, 1997.
[17] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy. “A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation”, Intl Workshop on Quality of Service, 1999.