A STUDY ON SCIENTIFIC APPLICATIONS IN CLOUD SERVICES

R.Divya,
M.Phil Scholar,
Department of Computer Science and Applications,
Vivekanandha College of Arts and Science College for Women (Autonomous),
Elayampalayam, Namakkal, Tamilnadu.

Dr.G.Kesavaraj,
Assistant Professor,
Department of Computer Science and Applications,
Vivekanandha College of Arts and Science College for Women (Autonomous),
Elayampalayam, Namakkal, Tamilnadu.

Abstract: Clouds are thus emerging as an important class of distributed computational resource, for both data-intensive and compute-intensive applications. There are novel usage modes that can be supported when grids and clouds are used concurrently. Clouds will have a broad impact on legacy scientific applications; because we anticipate that many existing legacy applications will adapt to and take advantage of new capabilities. However, it is unclear if clouds as currently presented are likely to change (many of) the fundamental reformulation of the development of scientific applications. In this paper we have discuss scientific application roles in cloud service models.

Keywords: Cloud Computing, scientific applications, Iaas, Paas, Saas

I. INTRODUCTION

Cloud computing is a kind of parallel and distributed computing systems that delivers infrastructure, platform and software as a service, which are made available as services in a pay-as-you-go model to consumer. These services are referred to as infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS). In [1] Buyya et al. define a cloud as a "type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as a one or more unified computing resources based on service-level agreements". Clouds try to make opportunity to the users all over the world to be able access the services on demand, according to their desired quality of service requirements. So it offers lots of benefits for companies by decreasing management and maintenance costs from leasing IT infrastructure from cloud providers.

Many scientific applications in the field of astronomy, gravitational-physics, computational biology, climate modeling, and life-sciences have used workflow technology to carry out large-scale experiments. Scientific applications are typically modeled as workflows that consist of tasks, data, control sequences and data dependencies [2]. Because of complexity of scientific process, these applications should be usually run on the large and distributed computing environments like cloud environment. Clouds present a chance for scientists whom need high performance computing infrastructure for their experiments [3].

Most of the time, applications are represented as a scientific workflows that can manage many activities and work with lots of data. Scientific already using cloud computing that schedule these workflows onto distributed cloud resources for optimizing various objectives: minimize total makespan of the workflow, minimize cost and usage of network bandwidth, minimize cost of computation and storage, meet the deadline of application, and combination of objectives.

Scientific computing often requires the availability of a massive number of computers for performing large scale experiments. Traditionally, these needs have been addressed by using high-performance computing solutions and installed facilities such as clusters and super computers, which are difficult to setup, maintain, and operate. Cloud computing provides scientists with a completely new model of utilizing the computing infrastructure. Compute resources, storage resources, as well as applications, can be dynamically provisioned (and integrated within the existing infrastructure) on a pay per use basis.

Scientific computing involves the construction of mathematical models and numerical solution techniques to solve scientific, social scientific and engineering problems. These models often require a huge number of computing resources to perform large scale experiments or to cut down the computational complexity into a reasonable time frame. These needs have been initially addressed with dedicated high-performance computing (HPC) infrastructures such as clusters or with a pool of networked machines in the same department, managed by some "CPU cycle scavenger" software such as Condor [4]. With the advent of Grid computing [5] new opportunities became available to scientists: in a complete analogy with the power Grid [6], the computing Grid could offer on demand the horse power required to perform large experiments, by relying on a network of machines, potentially extended all over the world.

Computing Grids introduced new capabilities such as dynamic discovery of services, the ability of relying on a larger number of resources belonging to different administrative domains and of finding the best set of machines meeting the requirements of applications. The use of Grids for scientific computing [7] has become so successful that many international projects led to the establishment of world-wide infrastructures available for computational science.

II. CLOUD COMPUTING

The term Cloud computing encompasses many aspects that range from the experience that end users have with the new opportunities offered by this technology to the implementation of systems that actually make these opportunities a reality. In this section, we will provide a characterization of what Cloud computing is, introduce a reference model for Cloud computing, and identify the key services that this new technology offers. Although, the term Cloud computing is too broad to be captured into a single definition it is possible to identify some key elements that characterize this trend. Armbrust et al. [8] observe that "Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the datacenters that provide those services".

They then identify the Cloud with both the hardware and the software components of a datacenter. A more structured definition is given by Buyya et al. [9] who define a Cloud as 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". One of the key features characterizing Cloud computing is the ability of delivering both infrastructure and software as services. More precisely, it is a technology aiming to deliver on demand IT resources on a pay per use basis. Previous trends were limited to a specific class of users, or specific kinds of IT resources. Cloud computing aims to be global: it provides the aforementioned services to the mass, ranging from the end user that hosts its personal documents on the Internet, to enterprises outsourcing their entire IT infrastructure to external datacenters.

III. CLASSIFICATION OF SCIENTIFIC APPLICATIONS AND SERVICES IN THE CLOUD

Common models of clouds [10] introduce composite hierarchies of different layers, each implementing a different service model (see Fig. 1). The services of each layer can be composed from the services of the layer underneath, and each layer may include one or more services that share the same or equivalent levels of abstraction. The proposed layers consist of the following: the Software as a Service (SaaS) layer, the platform as a service (PaaS) layer, and the Infrastructure as a Service (IaaS) layer. The IaaS layer can be further divided into the computational resources, storage, and communications sub layers, the software kernel layer, and the hardware/firmware layer that consists of the actual physical system components. As shown in Fig. 1, clouds can also be classified according to their deployment model into public and private clouds [11]. A public cloud is generally available on pay-per-use basis. Several infrastructures have emerged that enable the creation of so-called private clouds—that is, clouds that are only accessible from within an organization. Based on the proposed service layers, we will derive a classification from the application's perspective, with our aim to provide suggestions and raise further discussions on how scientific applications could possibly foster in the cloud environment. Although our taxonomy is targeted toward specific cloud environments, we strongly believe that a scientific application should and must remain interoperable regardless of the execution backend or the initial development infrastructure.

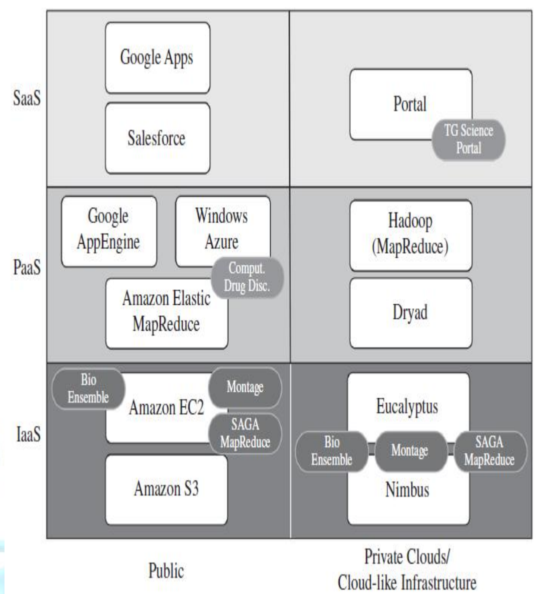

Fig. 1: Cloud taxonomy and application examples: Clouds provide services at different levels (IaaS, PaaS, SaaS).

The identification of how cloud application services fit into the layers may allow software developers to better comprehend the nature of parameters introduced in each layer. Such an assumption could lead into easier and more efficient implementation of cloud-operable scientific applications. Research work from the traditional cluster/grid era systems has already determined important features like scalability, extensibility, and high availability that should play an integral role in a distributed application's core functionality [12].

a) Software as a Service (SaaS) Layer

The software as a service layer is the highest layer in the proposed model. SaaS provides ready-to-run services that are deployed and conFig.d for the user. In general, the user has no control over the underlying cloud infrastructure with the exception of limited configuration settings. Regarding scientific applications, such a layer may represent an access point for the end user to reach a service, like a portal or a visualization tool. Scientific portals have been used by many grid services. A strong characteristic of SaaS services is that there is no client side software requirement. All data manipulated in such systems are held in remote infrastructures where all the processing takes place. One of the most prominent advantages of applications that are presented in this layer is universal accessibility regardless of the client system's software availability. This scheme provides flexibility to the end user and transparency of any complex mechanism involved. Some widely used examples of services that belong to this category are Google Apps and Salesforce [13]. These gateways provide among other things several domain specific web portals, which can be used to access computational and data services.

b) Platform as a Service (PaaS) Layer

The Platform as a Service (PaaS) layer provides the capability to deploy custom applications on the cloud providers' infrastructure. These applications are developed using the programming languages and APIs defined by the cloud provider. Similar to SaaS, the user has only limited control over the underlying cloud infrastructures: A developer does not have to worry about complex programming details, scalability, load balancing, or other system issues that may hinder the overall process of building an application. All such criteria are already specified by the given API that abstracts underlying architectural parameters. A well-known PaaS example is the Google App Engine [14] that equips developers with a Python and Java API and runtime environment for the implementation of web applications. Windows Azure [15] is Microsoft's PaaS platform and offers different types of runtime environments and storage services for applications. While, in particular, Google App Engine is primarily geared toward Web applications (such as science portals), Windows Azure is also well-suited for compute- and data-intensive applications.

Watson et al. [16] use Windows Azure—in particular the data storage and VM execution environment—to conduct data mining for computational drug discovery. Another PaaS abstraction that is used for parallel processing of large amounts of data is Map Reduce (MR) [17]. The framework solely requires the user to define two functions: the map and the reduce function. Both functions operate on key/value pairs: The map function transforms an input key/value pair representing a data row to an output key/value pair; the reduce function is used to merge all outputs of the map functions. Generally, the MapReduce framework handles all complexities and orchestrates the distribution of the the data as well as of the map and reduce tasks. Hadoop [18] is a well known example of an open-source MapReduce framework. Amazon's Elastic MapReduce provides a hosted MapReduce service. Another example of an environment for data-intensive computing is Microsoft Dryad. The framework allows the programmer to efficiently use resources for running data parallel applications. In Dryad a computation has the form of a directed graph (DAG), where the program instances that compose the computation are represented as graph vertices and the one-way communication channels between the instances are represented as graph edges.The Dryad infrastructure includes computational frameworks like Google's MapReduce. A port of Dryad to Windows Azure is planned, but at the time of writing is not available. PaaS clouds provider higher-level abstractions for cloud applications, which usually simplifies the application development process and removes the need to manage the underlying software and hardware infrastructure. PaaS offers automatic scalability, load balancing, and failure tolerance. However, the benefits are also associated with some drawbacks: Generally, PaaS services usually provide highly proprietary environments with only limited standard support. App Engine, for example, supports parts of the Java Enterprise API, but uses a custom BigTable-based data store.

c) Infrastructure-as-a-Service Layer

The infrastructure-as-a-service (Iaas) layer provides low-level, virtualized resources, such as storage, networks, and other fundamental computing resources via self-services to the user. In general, the user can deploy and run arbitrary software, which usually includes operating systems as well as applications. However, the user has no knowledge of the exact location and specifics of the underlying physical resources. Cloud providers usually offer instant elasticity; that is, new resources can be rapidly and elastically provisioned to scale-up or scale-out applications dynamically.

Computational cloud resources are represented through virtual machine instances (VMs), where the user is usually granted full administrative access and has the ability to build and deploy any kind of service infrastructure. Such VMs usually come with an OS already installed. The developer may choose a VM to rent that has the OS she wants. Amazon EC2 [19] is the prime example of such a service and currently offers a variety of VM images, where one may choose to work on a Windows platform or on some Linux-based platforms.

The developer can further configure. and add extra libraries to the selected OS to accommodate an application. Rackspace and GoGrid provide similar services. Eucalyptus and Nimbus offer EC2 compatible infrastructures, which can be deployed in-house in a private cloud. Several scientific clouds utilize these frameworks—for example, Science Cloud and Future Grid. VMs are provided to the user under SLAs, where the cloud provider guarantees a certain level of system's performance to their clients. They usually involve fees on behalf of the user utilizing the leased computational resources, while open source/research cloud infrastructures don't include any financial requirement. When a team of scientists rents some virtual resources to run their experiments, they usually also lease data storage to store their data/results remotely and access them within the time limits of their agreement with the service provider. Examples of public cloud storage service are Amazon S3 and Rackspace Cloud Files . Walrus is a S3 interface compatible service, which can be deployed on private cloud infrastructures. Another common cloud-like infrastructure is distributed file systems, such as the Google File System (GFS) and the Hadoop File System (HDFS) [20]. Both systems are optimized for storing and retrieving large amounts of data.

IV. CONCLUSION

Several scientific applications from different domains (e.g., life sciences, high energy physics, astrophysics, computational chemistry) have been ported to cloud environments. The majority of these applications rely on IaaS cloud services and solely utilize static execution modes: A scientist leases some virtual resources in order to deploy their testing services. One may select different number of instances to run their tests on. An instance of a VM is perceived as a node or a processing unit. There can be a multiple number of instances under the same VM, depending on the SLA one has agreed on. Once the service is deployed, a scientist can begin testing on the virtual nodes; this is similar to how one would use a traditional set of local clusters. This paper discussed about scientific application roles in cloud service models.

V. REFERENCES

[1]. Buyya R, Yeo CS, Venugopal S, Broberg J and Brandic I "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the5th utility. Future Generation Computer System", Vol. 25, No. 6, Pp. 599–616, September 2009.

[2]. Pandy S and Buyya R, "Scheduling and Management Techniques for Data-Intensive Application Workflows", IGI Global, USA, September 2009.

[3]. Lee J, Tierney B and Johnston WE, " Data Intensive Distributed Computing; A Medical Application Example. in HPCN Europe '99: Proceedings of the 7th International Conference on High-Performance Computing and Networking". London, UK:Springer-Verlag, Pp. 150–158.

[4] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: The condor experience," Concurrency and Computation: Practice and Experience, Vol. 17, Pp. 323–356, February 2005.

[5] Foster and C. Kesselman, "The Grid: blueprint for a new computing infrastructure", I.Foster and C. Kesselman, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., November 1998.

[6] M. Chetty and R. Buyya, "Weaving Computational Grids: How Analogous Are They with Electrical Grids?," Computing in Science and Engineering (CiSE), Vol. 4 Pp. 61– 71, doi:10.1109/MCISE.2002.1014981, July-August 2002.

[7] M. J. Chin, S. Harvey, S. Jha, and P. V. Coveney, "Scientific Grid Computing: The First Generation," Computing in Science and Engineering, Vol. 7, Pp. 24–32, August 2005.

[8] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the Clouds: "A Berkeley View of Cloud computing", Technical Report No. UCB/EECS-2009-28, University of California at Berkley, USA, February 2009.

[9] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", Keynote Paper, in Proc. 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008), IEEE CS Press, Dalian, China. Pp. 25–27, September 2008.

[10] P. Mell and T. Grance, The NIST definition of cloud computing.

[11] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing", in Proceedings of the Grid Computing Environments Workshop, GCE '08, November 2008.

[12] M. Armbrust et al., Above the clouds: "A Berkeley View of Cloud Computing", Technical Report UCB/EECS-2009_28, EECS Department, University of California, Berkeley, February 2009.

[13] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam, "TeraGrid science gateways and their impacton science Computer", Vol. 41(11), Pp. 32_41, june 2008.

[14] Google App E ngine, http://code.google.com/appengine/.

[15] WindowsAzure, http://www.microsoft.com/windowsazure/

[16] P. Watson, D. Leahy, H. Hiden, S. Woodman, and J. Berry, An Azure "Science Cloud for Drug Discovery", Microsoft External Research Symposium, 2009.

[17] J. Dean and S. Ghemawat, MapReduce: "Simplified data processing on large clusters", in Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation, Berkeley, CA, USENIX Association, Pp. 137_150, September 2004.

[18] Hadoop: Open Source Implementation of MapReduce, http://hadoop.apache.org/.

[19] Amazon EC2 Web Service, http://ec2.amazonaws.com.

[20] HDFS http://hadoop.apache.org/common/docs/current/hdfs_design.html.