State-of-the-Art on Cloud Ontology

Suruchi Agrawal and Harsh Gupta

Abstract--- Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. A computing Cloud is a set of network enabled service architecture, providing scalability, guaranteed Quality of Service (QoS), reduced information technology overhead for the end-user and great flexibility thereby providing cost effective ownership in a simple and pervasive way. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The resource sharing results in various cloud level offerings such as infrastructure cloud (e.g. hardware, IT infrastructure management), software cloud (e.g. SaaS focusing on middleware as a service, or traditional CRM as a service), application cloud (e.g. Application as a Service, UML modeling tools as a service, social network as a service), and business cloud (e.g. business process as a service). In this paper, we present a peer-review of cloud computing, highlighting its key concepts, architectural principles, state-of-the-art implementation as well as research design challenges.

Keywords--- Cloud Computing, QoS, SaaS, UML, CRM

Suruchi Agrawal, M.Tech (Computer Science), Department of CSE, SEEC, Manipal University Jaipur(MUJ). E-mail: agrawal.suruchi123@gmail.com

Harsh Gupta, M.Tech (Microelectronics), Department of ECE, SEEC, Manipal University Jaipur(MUJ). E-mail:harsh.gupta123@gmail.com

I. INTRODUCTION

The term cloud computing seems to originate from computer network diagrams that represent the internet as a cloud. The working definition summarizes cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2]. In this paper, we have performed a systematic review of all peer-reviewed academic research on cloud computing, and explained the technical challenges being faced.

Research in this field appeared to be split into two distinct viewpoints. One investigates the technical issues that arise when building and providing clouds, and the other looks at implications of cloud computing on enterprises and users. In this paper we discuss the advances and technical aspects of Cloud Computing, such as protocols, interoperability and techniques for building clouds, while discussing the research challenges faced by enterprise users, such as cost evaluations, legal issues, trust, privacy, security, and the effects of cloud computing [1]. This paper is structured as follows: the cloud architecture is shown in the Section 2; Section 3 discusses various cloud standards and interfaces; Section 4 reviews the work on standardized interfaces and Cloud interoperability; Section 5 summarizes various other research done in support of building Cloud infrastructures; while use cases of Cloud computing are reviewed in Section 6; finally Section 7 concludes the review by summing up the state-of-the-art on cloud computing.

II. CLOUD ARCHITECTURE

Cloud computing has 5 essential characteristics, 3 service models, and 4 deployment models. The essential characteristics are (as shown in Figure 1):

- On-demand self-service: computing resources can be acquired and used anytime without the need for human interaction with cloud service providers. Computing resources include processing power, storage, virtual machines etc.
- Broad network access: the previously mentioned resources can be accessed over a network using heterogeneous devices such as laptops or mobiles phones.
- Resource pooling: cloud service providers pool their resources that are then shared by multiple users. This is referred to as multi-tenancy where for ex. a physical server may host several virtual machines belonging to different users.
- Rapid elasticity: a user can quickly acquire more resources from the cloud by scaling out. It can scale back in by releasing those resources once they are no longer required.
- Measured service: resource usage is metered using appropriate metrics such monitoring storage usage, CPU hours, bandwidth usage etc.
- The above characteristics apply to all clouds but each cloud provides users with services at a different level of abstraction, which is referred to as a service model. The 3 most common service models are:

- Software as a Service (SaaS): this is where users simply make use of a web-browser to access software that others have developed and offer as a service over the web. At the SaaS level, users do not have control or access to the underlying infrastructure being used to host the software. For ex. Salesforce's Customer Relationship Management software3 and Google Docs4 use the SaaS model of cloud computing.
- Platform as a Service (PaaS): this is where applications are developed using a set of programming languages and tools that are supported by the PaaS provider. PaaS provides users with a high level of abstraction that allows them to focus on developing their applications and not worry about the underlying infrastructure. Just like the SaaS model, users do not have control or access to the underlying infrastructure being used to host their applications at the PaaS level. Google App Engine5 and Microsoft Azure6 are popular PaaS examples.
- Infrastructure as a Service (IaaS): this is where users acquire computing resources such as processing power, memory and storage from an IaaS provider and use the resources to deploy and run their applications. In contrast to the PaaS model, the IaaS model is a low level of abstraction that allows users to access the underlying infrastructure through the use of virtual machines. IaaS gives users more flexibility than PaaS as it allows the user to deploy any software stack on top of the operating system. However, flexibility comes with a cost and users are responsible for updating and patching the operating system at the IaaS level. For ex. Amazon Web Services EC2 and S37 have IaaS facility.

Erdogmus [3] described Software as a Service as the core concept behind cloud computing, suggesting that it does not matter whether the software being delivered is infrastructure, platform or application, "it's all software in the end" [3].

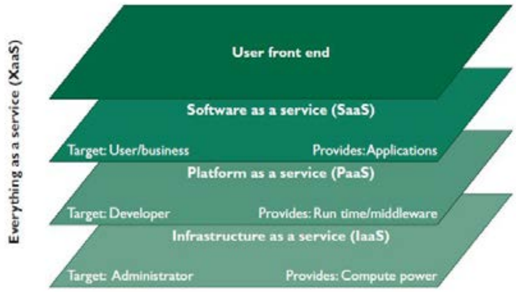

Figure 1: Layered Architecture of Cloud infrastructures

It helps us to distinguish between the types of service being delivered as they have different abstraction levels. The service models are deployed in clouds, but there are different types of clouds depending on who owns and uses them. This is referred to as a cloud deployment model and 4 common models are:

- Private cloud: a cloud that is used exclusively by one organisation. The cloud may be operated by the organisation itself or a third party. The St Andrews Cloud Computing Co-laboratory8 and Concur Technologies [4] are example organizations that have private clouds.
- Public cloud: a cloud that can be used (for a fee) by the general public. Public clouds require significant investment and are usually owned by large corporations such as Microsoft, Google or Amazon.
- Community cloud: a cloud that is shared by several organizations and is usually setup for their specific requirements. The Open Cirrus cloud testbed could be regarded as a community cloud that aims to support research in cloud computing [5].
- Hybrid cloud: a cloud that is setup using a mixture of the above three deployment models. Each cloud in a hybrid cloud could be independently managed but applications and data would be allowed to move across the hybrid cloud. Hybrid clouds allow cloud bursting to take place, which is where a private cloud can burst-out to a public cloud when it requires more resources.

Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs.

Figure 2 provides an overview of the common deployment and service models in cloud computing, where the 3 service models could be deployed on top of any of the 4 deployment models.

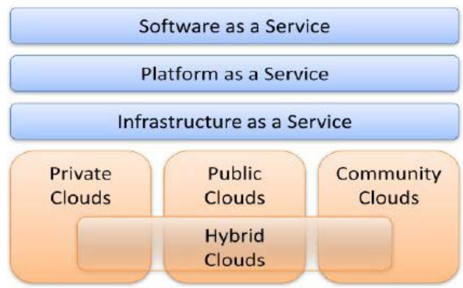

Figure 2: Cloud Computing Deployment and Service Models

The above model includes three of the five characteristics of cloud computing described, namely resource pooling, rapid elasticity and measured service but fails to mention on-demand self-service and broad network access. Youseff et al. [6] described a five-layer stack that can be used to classify cloud services; use composability as their methodology where each service is composed of other services. The 5 layers are applications, software environment, software infrastructure, software kernel, and hardware. This is similar to the SaaS, PaaS and IaaS service models described and only differs in the lower two layers, namely the software kernel and hardware layers. Grid and cluster computing systems such as Globus and Condor are examples of cloud services that fall into the software kernel layer, and ultra large-scale data centers as designed in IBM's Kittyhawk Project [7] are examples of hardware layer services [6].

III. CLOUD STANDARDS AND INTERFACES

It is useful to think of a cloud as a collection of hardware and software that runs in a data centre and enables the cloud computing model [8]. "Scalability, reliability, security, ease of deployment, and ease of management for customers, traded off against worries of trust, privacy, availability, performance, ownership, and supplier persistence" are the benefits of cloud computing [3].

Voas and Zhang [9] identified cloud computing as the next computing paradigm that follows on from mainframes, PCs, networked computing, the internet and grid computing. One of the reasons that prevented grid computing from being widely used was the lack of virtualization that resulted in jobs being dependant on the underlying infrastructure. This often resulted in unnecessary complexity that had an effect on wider adoption [10]. Ian Foster - who was one of the pioneers of grid computing compared cloud computing with grid computing and concluded that although the details and technologies of the two are different, their vision is essentially the same [11]. These design goals included remote terminal access, continuous operational provision, scalability, reliable file systems that users trust to store their only copy of files, information sharing controls, and an ability to support different programming environments [12].

As cloud computing delivers IT as a service, cloud researchers can also learn from service oriented architecture (SOA). In fact, the first paper that introduced PaaS [13] described PaaS as an artefact of combining infrastructure provisioning with the principles of SaaS and SOA. Sedayao [14] built a monitoring tool using SOA services and principles, and describe their experience from building a robust distributed application consisting of unreliable parts and the implication for cloud computing. As design goal for distributed cloud computing analogous to "routers in a network, any service using other cloud services needs to validate input and have hold down periods before determining that a service is down"[14].

There are three groups currently working on standards for cloud computing: The Cloud Computing Interoperability Forum9, the Open Cloud Consortium10, and the DMTF Open Cloud Standards Incubator11. Cloud computing can benefit from standardized API interfaces as generic tools that manage cloud infrastructures to be developed for all offerings. For IaaS there are developments towards standards and Eucalyptus is looking to become the de-facto standard. Achieving standardized APIs appears to be rather politically than technically challenging, hence there seems to be little space for academic involvement. However, standardized interfaces alone do not suffice to prevent vendor lock-in. For an open cloud, there is a need for protocols and software artifacts that allow interoperability to unlock more of the potential benefits from cloud computing.

IV. CLOUD INTEROPERABILITY AND NOVEL PROTOCOLS

The next steps from compatible and standardized interfaces towards utility provisioning are universal open and standard protocols that allow interoperability between clouds and enable the use of different offerings for different use cases. There are ways of [15] allowing cloud services to interoperate with other clouds and highlight many goals and challenges, such as that cloud services should be able to implicitly use others through some form of library without the need to explicitly reference them, e.g. with their domain name and port. The collection of protocols inside and in-between the clouds that solve interoperability in the cloud are termed intercloud protocols. The intercloud protocol research agenda is made up of several areas: addressing, naming identity and trust, presence and messaging, virtual machines, multicast, time synchronization, and reliable application transport.

For cloud computing, each of these areas contains several issues. In addressing for example, the research problem is that there is the limited address space in IPv4 and that its successor IPv6 might be an inappropriate approach in a large and highly virtualized environment, as the cloud, due to its static addressing scheme: Bernstein et al criticize that IP addresses traditionally embody network locations for routing purposes and identity information, but in the cloud context identifiers should allow the objects to move into different subnets dynamically. This problem of static addresses is addressed by Ohlman et al. [16]. They recommend the usage of Networking of Information (NetInf) for cloud computing systems. Unlike URLs which are location-dependent, NetInf uses a location-independent model of naming objects, and offers an API that hides the dynamics of object locations and network topologies.

V. CLOUD BUILDING

Building cloud offerings requires management software, hardware provision, simulators to evaluate the design, and evaluating management choices. Sotomayor et al. [17] presents two tools for managing cloud infrastructures: OpenNebula, a virtual infrastructure manager, and Haizea, a resource lease manager. To manage the virtual infrastructure, OpenNebula provides a unified view of virtual resources regardless of the underlying virtualization platform, manages the full lifecycle of the VMs, and support configurable resource allocation policies including policies for times when the demand exceeds the available resources. Sotomayor et al. argue that in private and hybrid clouds resources will be limited, in the sense that situations will occur where the demand cannot be met, and that requests for resources will have to be prioritized, queued, pre-reserved, deployed to external clouds, or even rejected. Haizea can act as a scheduling backend for OpenNebula, and together they advance other virtual infrastructure managers by giving the functionality to scale out to external clouds, and providing support for scheduling groups of VMs, such that either the entire group of VMs are provided resources or no member of the group. In combination they can provide resources by best-effort, as done by Amazon EC2, by immediate provision, as done by Eucalyptus, and using advance reservations (shown in Figure 3).

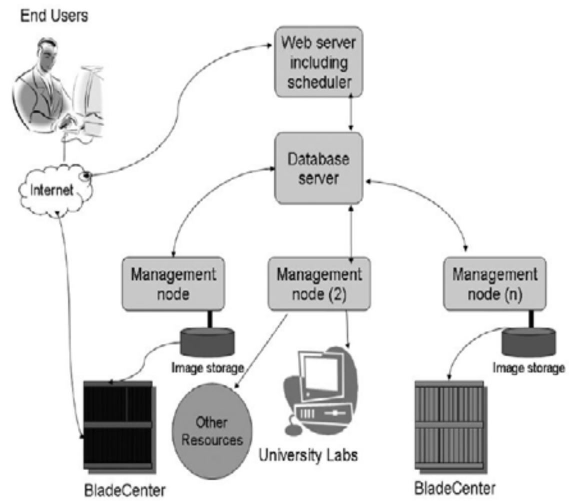

Figure 3: Cloud Building States

Sriram [18] discusses some of the issues with scaling the size of data centers used to provide cloud computing services. It presents the development and initial results of a simulation tool for predicting the performance of cloud computing data centers which incorporates normal failures, failures that occur frequently due to the sheer number of components and the expected average lifecycle of each component and that are treated as the normal case rather than as an exception. It shows that for small data centers and small failure rates the middleware protocol does not play a role, but for large data centers distributed middleware protocols scale better. CloudSim, another modelling and simulation toolkit has been proposed by Buyya et al. [19]. CloudSim simulates the performance of consumer applications executed in the cloud. The topology contains a resource broker and the data centers where the application is executed. The simulator can then estimate the performance overhead of the cloud solution. It is built on top of a grid computing simulator (GridSim) and looks at the scheduling of the execution application, and the impact of virtualization on the application's performance.

VI. USE CASES IN CLOUD COMPUTING

Chun and Maniatis [20] describe one such use-case, where cloud computing enables a technology which otherwise would not be possible: to overcome hardware limitations and enable more powerful applications on smartphones, they use external resources. This is done by partially off-loading execution from the smartphone and using cloud resources. But, Chun and Maniatis also include laptops or desktops near the phone in their "cloud" because of the network latency for phones. Depending on the use case, their model offloads entire computations or parts thereof, and only has the remainder executed locally.

Another use-case that becomes feasible and affordable through the use of cloud computing is large-scale non-functional requirements testing, as described by Ganon and Zilbershtein [21]. They tested Network Management Systems for systems where much of the functionality is in the endpoints, such as in voice over IP software.

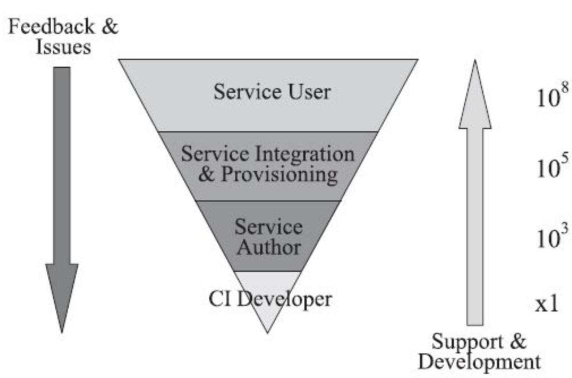

Figure 4: Cloud User Hierarchy

In a cloud that offers IaaS, the number of VMs and thus instances of operating systems that need to be managed increases significantly. To avoid having to deploy software and updates into each virtual machine, and to avoid lengthy installation processes, entire so called "virtual appliances" will be managed. This means, in cloud computing the operating system will no longer be viewed separated from the applications deployed, but rather both will be deployed and maintained jointly. For service providers this means, they now have the ability to offer a virtual appliance, as functional disc image, instead of having to create lengthy installation procedures to guarantee compatibility with other applications in the VM. Wilson [22] describes Coronary, a software configuration management tool for virtual appliances. Coronary takes the idea of incremental updates from configuration management software such as CVS or subversion, and uses this technology to manage virtual appliances over their lifecycle as shown in Figure 4.

VII. CONCLUSION

To conclude, this paper discusses the research and technological aspects of cloud computing, and highlights the attempts been made at building unified APIs to access clouds which seem to be more politically than technically challenging. Cloud computing uses a number of characteristics, service models and deployment models. The socio-technical aspects of cloud computing that were reviewed included the costs of using and building clouds, the security, legal and privacy implications that cloud computing raises as well as the effects of cloud computing on the work of IT departments. The technological aspects that were reviewed included standards, cloud interoperability, lessons from related technologies, building clouds, and use-cases that presented new technological possibilities enabled by the cloud. The technical challenges in cloud computing involves the adoption of cloud computing, such as availability of service and data lock-in, lack of scalable storage, data transfer and performance unpredictability. [23] Cloud computing could benefit from the functionality modelling issues from service computing, and the context-sensitivity issues from pervasive computing.

REFERENCES

[1] KHAJEH-HOSSEINI, A., SOMMERVILLE, I. and SRIRAM, I. "Research Challenges for Enterprise Cloud Computing," (unpublished).(Submitted to 1st ACM Symposium on Cloud Computing, Indianapolis, Indiana, USA, June 2010, under paper id 54)

[2] MELL, P. and GRANCE, T. 2009. Draft NIST Working Definition of Cloud Computing.

[3] ERDOGMUS, H. 2009. Cloud Computing: Does Nirvana Hide behind the Nebula? Software, IEEE 26, 2, 4-6.

[4] LEMOS, R. 2009. Inside One Firm's Private Cloud Journey. Retrieved December 1, 2009, fromhttp://www.cio.com/article/506114/Inside_One_Firm_s_Private_Cloud_Journey

[5] Open CirrusTM: the HP/Intel/Yahoo! Open Cloud Computing Research Testbed. Retrieved December 1, 2009, from https://opencirrus.org/

[6] YOUSEFF, L., BUTRICO, M. and DA SILVA, D. 2008. Toward a Unified Ontology of Cloud Computing. In Grid Computing Environments Workshop, 2008. GCE '08, 1-10.

[7] J. Appavoo, V. Uhlig, and A. Waterland, "Project Kittyhawk: Building a Global-Scale Computer," vol. 42, 2008, pp. pp. 77-84.

[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, 2009.

[9] VOAS, J. and ZHANG, J. 2009. Cloud Computing: New Wine or Just a New Bottle? IT Professional 11, 2, 15-17.

[10] VOUK, M. A. 2008. Cloud computing — Issues, research and implementations. In Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on, 31-40.

[11] FOSTER, I., ZHAO, Y., RAICU, I. and LU, S. 2008. Cloud Computing and Grid Computing 360-Degree Compared. In Grid Computing Environments Workshop (GCE '08), Austin, Texas, USA, November 2008, 1-10.

[12] Corbató, F. J., Saltzer, J. H., and Clingen, C. T. 1972. Multics: the first seven years. In Proceedings of the May 16-18, 1972, Spring Joint Computer Conference, Atlantic City, New Jersey, May 1972, 571-583.

[13] CHANG, M., HE, J., and E. Leon, "Service-Orientation in the Computing Infrastructure," 2006, pp. 27-33.

[14] SEDAYAO, J. 2008. Implementing and operating an internet scale distributed application using service oriented architecture principles and cloud computing infrastructure. In iiWAS '08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, 417-421.

[15] BERNSTEIN, D., LUDVIGSON, E., SANKAR, K., DIAMOND, S. and MORROW, M. 2009. Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability. In Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on, 328-336.

[16] OHLMAN, B., ERIKSSON, A. and REMBARZ, R. 2009. What Networking of Information Can Do for Cloud Computing. In Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on, 78-83.

[17] SOTOMAYOR, B., RUBÃ©, A., LLORENTE, I. and FOSTER, I. 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. Internet Computing, IEEE 13, 5, 14-22.

[18] SRIRAM, I. 2009. A Simulation Tool Exploring Cloud-Scale Data Centres. In 1st International Conference on Cloud Computing (CloudCom 2009), pp. 381-392.

[19] BUYYA, R., RANJAN, R. and CALHEIROS, R. N. 2009. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In High Performance Computing & Simulation, 2009. HPCS '09. International Conference on, 1-11.

[20] CHUN, B.-G. and MANIATIS, P. 2009. Augmented Smart Phone Applications Through Clone Cloud Execution. In Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII).

[21] GANON, Z. and ZILBERSHTEIN, I. E. 2009. Cloud-based Performance Testing of Network Management Systems. In Computer Aided Modeling and Design of Communication Links and Networks, 2009. CAMAD '09. IEEE 14th International Workshop on, 1-6.

[22] WILSON, M. 2009. Constructing and Managing Appliances for Cloud Deployments from Repositories of Reusable Components.

[23] MEI, L., CHAN, W. K. and TSE, T. H. 2008. A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues. In Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE, 464-469.