A REVIEW ARTICLE OF CLOUD COMPUTING

Mandeep Kaur [(1)], Amandeep Kaur [(2)], Mandeep Kaur [(3)]

[(1)]Asst. Prof, Ferozepur College of Engineering & Technology, Ferozepur

mandeep.kaur399@gmail.com

[(2)]Lecturer in Dev Samaj College, Ferozepur

amanthind59@gmail.com

[(3)]Lecturer in Shaheed Bhagat Singh State Technical Campus(Poly), Ferozepur

menuk8@gmail.com

Abstract

Cloud computing has become a significant technology trend, and many experts expect it to reshape information-technology processes and the IT marketplace during the next five years.

This keynote paper: presents a 21st century vision of computing; identifies various computing paradigms promising to deliver the vision of computing utilities; defines Cloud computing and provides the architecture for creating market-oriented Clouds by leveraging technologies such as VMs; provides thoughts on market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA oriented resource allocation; presents some representative Cloud platforms especially those developed in industries along with our current work towards realising market-oriented resource allocation of Clouds by leveraging the 3rd generation Aneka enterprise Grid technology; reveals our early thoughts on interconnecting Clouds for dynamically creating an atmospheric computing environment along with pointers to future community research; and concludes with the need for convergence of competing IT paradigms for delivering our 21st century vision.

Introduction to Cloud computing

Cloud Computing is the use of Internet-based technologies for the provision of services [1], originating from the cloud as a metaphor for the Internet, based on depictions in computer network diagrams to abstract the complex infrastructure it conceals [8]. It can also be seen as a commercial evolution of the academic-oriented Grid Computing [9], succeeding where Utility Computing struggled [10], [11], while making greater use of the self-management advances of Autonomic Computing [12]. It offers the illusion of infinite computing resources available on demand, with the elimination of upfront commitment from users, and payment for the use of computing resources on a short term basis as needed [3]. Furthermore, it does not require the node providing a service to be present once its service is deployed [3]. It is being promoted as the cutting-edge of scalable web application development [3], in which dynamically scalable and often virtualised resources are provided as a service over the Internet [13], [1], [14], [15], with users having no knowledge of, expertise in, or control over the technology infrastructure of the Cloud supporting them [16]. It currently has significant momentum in two extremes of the web development industry [3], [1]: the consumer web technology incumbents who have resource surpluses in their vast data centres1, and various consumers and start-ups that do not have access to such computational resources. Cloud Computing conceptually incorporates Software-as-a-Service (SaaS) [18], Web 2.0 [19] and other technologies with reliance on the Internet, providing common business applications online through web browsers to satisfy the computing needs of users, while the software and data are stored on the servers. Figure 1 shows the typical configuration of Cloud Computing at run-time when consumers visit an application served by the central Cloud, which is housed in one or more data centres [20]. Green symbolises resource consumption, and yellow resource provision. The role of coordinator for resource provision is designated by red, and is centrally controlled. Even if the central node is implemented as a distributed grid, which is the usual incarnation of a data centre, control is still centralised. Providers, who are the controllers, are usually companies with other web activities that require large computing 1 A data centre is a facility, with the necessary security devices and environmental systems (e.g. air conditioning and fire suppression), for housing a server farm, a collection of computer servers that can accomplish server needs far beyond the capability of one machine [17].

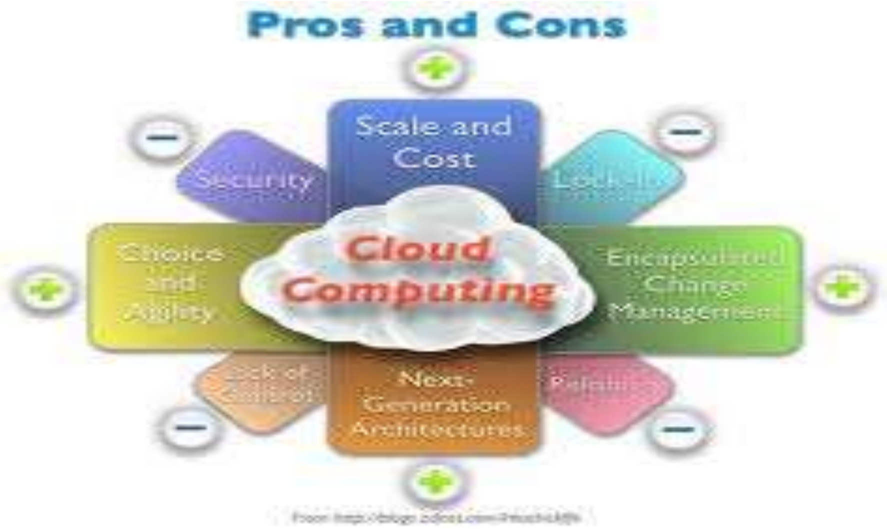

Fig-1: Clouding Computing

Market-Oriented Cloud Architecture

As consumers rely on Cloud providers to supply all their computing needs, they will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations. Cloud providers will need to consider and meet different QoS parameters of each individual consumer as negotiated in specific SLAs. To achieve this, Cloud providers can no longer continue to deploy traditional system-centric resource management architecture that do not provide incentives for them to share their resources and still regard all service requests to be of equal importance. Instead, market-oriented resource management [7] is necessary to regulate the supply and demand of Cloud resources at market equilibrium, provide feedback in terms of economic incentives for both Cloud consumers and providers, and promote QoS-based resource allocation mechanisms that differentiate service requests based on their utility. Figure 2 shows the high-level architecture for supporting market-oriented resource allocation in Data Canters and Clouds. There are basically four main entities involved:

• Users/Brokers: Users or brokers acting on their behalf submit service requests from anywhere in the world to the Data Centre and Cloud to be processed.

• SLA Resource Allocator: The SLA Resource Allocator acts as the interface between the Data Canter/Cloud service provider and service user.

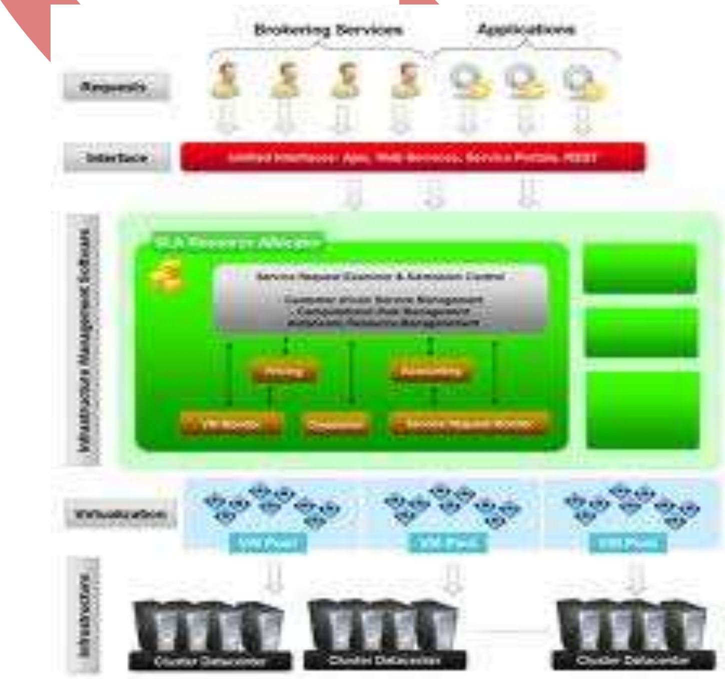

Fig-2: High-level market-oriented cloud architecture.

THE CLOUD ONTOLOGY

Cloud computing systems fall into one of five layers: applications, software environments, software infrastructure, software kernel, and hardware. Obviously, at the bottom of the cloud stack is the hardware layer which is the actual physical components of the system. Some cloud computing offerings have built their system on subleasing the hardware in this layer as a service, as we discuss in subsection IV-E. At the top of the stack is the cloud application layer, which is the interface of the cloud to the common computer users through web browsers and thin computing terminals. We closely examine the characteristics and limitations of each of the layers in the next five subsections.

Cloud Application Layer: The cloud application layer is the most visible layer to the end-users of the cloud. Normally, the users access the services provided by this layer through web-portals, and are sometimes required to pay fees to use them. This model has recently proven to be attractive to many users, as it alleviates the burden of software maintenance and the ongoing operation and support costs. Furthermore, it exports the computational work from the users' terminal to data centres where the cloud applications are deployed. This in turn lessens the restrictions on the hardware requirements needed at the users' end, and allows them to obtain superb performance to some of their cpu-intensive and memory-intensive workloads without necessitating huge capital investments in their local machines. As for the providers of the cloud applications, this model even simplifies their work with respect to upgrading and testing the code, while protecting their intellectual property. Since a cloud application is deployed at the provider's computing infrastructure (rather than at the users' desktop machines), the developers of the application are able to roll smaller patches to the system and add new features without disturbing the users with requests to install major updates or service packs. Configuration and testing of the application in this model is arguably less complicated, since the deployment environment becomes restricted, i.e., the provider's data canter. Even with respect to the provider's margin of profit, this model supplies the software provider with a continuous flow of revenue, which might be even more profitable on the long run. This model conveys several favourable benefits for the users and providers of cloud applications, and is normally referred to as Software as a Service (SaaS). Sales force Customer Relationships Management (CRM) system [7] and Google Apps [8] are two examples of SaaS. As such, the body of research on SOA has numerous studies on compostable IT services which have direct application to providing and composing SaaS. Our proposed ontology illustrates that cloud applications can be developed on the cloud software environments or infrastructure components (as discussed in the next two subsections). In addition, cloud applications can be composed as a service from other cloud services offered by other cloud systems, using the concepts of SOA. For example, a payroll application might use another accounting SaaS to calculate the tax deductibles for each employee in its system without having to implement this service within the payroll software. In this respect, the cloud applications targeted for higher layers in the stack are simpler to develop and have a shorter time-to market. Furthermore, they become less error-prone since all their interactions with the cloud are through pretested APIs. Developed for a higher cloud-stack layer, the flexibility of the applications is however limited and this may restrict the developers' ability to optimize their applications' performance. Despite all the advantageous benefits of this model, several deployment issues hinder its wide adoption. Specifically, the security and availability of the cloud

Cloud Software Environment Layer: The second layer in our proposed cloud ontology is the cloud software environment layer (also dubbed the software platform layer). The users of this layer are cloud applications' developers, implementing their applications for and deploying them on the cloud. The providers of the cloud software environments supply the developers with a programming-language-level environment with a set of well-defined APIs to facilitate the interaction between the environments and the cloud applications, as well as to accelerate the deployment and support the scalability needed of those cloud applications. The service provided by cloud systems in this layer is commonly referred to as Platform as a Service (PaaS). One example of systems in this category is Google's App Engine [5], which provides a python runtime environment and APIs for applications to interact with Google's cloud runtime environment. Another example is Sales Force Apex language [9] that allows the developers of the cloud applications to design, along with their applications' logic, their page layout, workflow, and customer reports. Developers reap several benefits from developing their cloud application for a cloud programming environment, including automatic scaling and load balancing, as well as integration with other services (e.g. authentication services, email services, user interface) provided to them through the PaaS-provider. In such a way, much of the overhead of developing cloud applications is alleviated and is handled at the environment level. Furthermore, developers have the ability to integrate other services to their applications on-demand. This in turn makes the cloud application development a less complicated task, accelerates the deployment time and minimizes the logic faults in the application. In this respect, a Hadoop [10] deployment on the cloud would be considered a cloud software environment, as it provides its applications' developers with a programming environment, i.e. map reduce framework for the cloud. Similarly, Yahoo's Pig [11], a high-level language to enable processing of very large files on the hadoop environment may be viewed as an open-source implementation of the cloud platform layer. As such, cloud software environments facilitate the process of the development of cloud applications. 4 C. Cloud Software Infrastructure Layer The cloud software infrastructure layer provides fundamental resources to other higher-level layers, which in turn can be used to construct new cloud software environments or cloud applications. Our proposed ontology reflects the fact that the two highest levels in the cloud stack can bypass the cloud infrastructure layer in building their system. Although this bypass can enhance the efficiency of the system, it comes at the cost of simplicity and development efforts.

REFERENCES

[1] M. Haynie, "Enterprise cloud services: Deriving business value from Cloud Computing," Micro Focus, Tech. Rep., 2009.
[2] R. L. Peter Lucas, Joseph Ballay, "The wrong cloud," MAYA Design, Inc, Tech. Rep., 2009.[Online].Availablehttp://www.maya.com/filedownload/126/The%20Wrong%20Clouddf
[3] 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," University of California, Berkeley, 2009. [Online]. Available: http://d1smfj0g31qzek.cloudfront.net/ abovetheclouds.pdf
[4] J. Hayes, "Cred - or croak?" IET Knowledge Network, Tech. Rep., 2008. [Online]. Available: http://kn.theiet.org/magazine/ issues/0820/cred-croak-0820.cfm?SaveToPDF

[5] P. Mckenna, “Can we stop the internet destroying our planet?” New Scientist, vol. 197, no. 2637, pp. 20–21, 2008.
[6] J. Kaplan,W. Forrest, and N. Kindler, “Revolutionizing data center energy efficiency,” McKinsey&Company,Tech.Rep.,2008.[Online].Availablehttp://www.mckinsey.com/clientservice/bto/ pointofview/pdf/Revolutionizing Data Center Efficiency.pdf
[7] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, and S. Wolff, “A brief history of the internet,” Institute for Information Systems and Computer Media, Tech. Rep., 2001. [Online]. Available: http://www.iicm.tugraz. at/thesis/cguetl diss/literatur/Kapitel02/References/Leiner et al. 2000/brief.html?timestamp=1197467969844
[8] J. Scanlon and B. Wieners, “The internet cloud,” The Industry Standard, Tech. Rep., 1999. [Online]. Available: http://www.thestandard.com/article/0,1902,5466,00.html
[9] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-degree compared,” in Grid Computing Environments Workshop, 2008, pp. 1–10.
[10] T. Foremski, “Sun services CTO says utility computing acceptance is slow going,” ZDNet,CBSInteractive,Tech.Rep.,2006.[Online].Available:http://blogs.zdnet.com/Foremski/?p=33
[11] A. Orlowski, “The Cell chip - how will MS and Intel face the music?” The Register, Tech. Rep., 2005. [Online]. Available: http://www.theregister.co.uk/2005/02/03/cell analysis part two/
[12] J. Kephart, D. Chess, I. Center, and N. Hawthorne, “The vision of autonomic computing,” Computer, vol. 36, no. 1, pp. 41–50, 2003.
[13] G. Gruman and E. Knorr, “What Cloud Computing really means,” InfoWorld Inc., Tech. Rep., 2008. [Online]. Available: http://www.infoworld.com/article/08/04/07/ 15FE-cloud-computing-reality 1.html
[14] Gartner, “Cloud Computing will be as influential as ebusiness,” Gartner, Tech. Rep., 2008. [Online]. Available: http://www.gartner.com/it/page.jsp?id=707508
[15] P. Gaw, “What’s the difference between Cloud Computing and SaaS?” Proofpoint, Tech. Rep., 2008. [Online]. Available: http://blog.fortiva.com/fortivablog/2008/05/what-is-the dif.html
[16] K. Danielson, “Distinguishing Cloud Computing from Utility Computing,” ebizQ, Tech. Rep., 2008. [Online]. Available: http://www.ebizq.net/blogs/saasweek/2008/03/ distinguishing cloud computing/
[17] M. Arregoces and M. Portolani, Data center fundamentals. Cisco Press, 2003.
[18] M. Turner, D. Budgen, and P. Brereton, “Turning software into a service,” Computer, vol. 36, no. 10, pp. 38–44, 2003.
[19] T. Oreilly, “What is Web 2.0: Design patterns and business models for the next generation of software,” O’Reilly Media, Tech. Rep., 2008. [Online]. Available: http://www.oreillynet.com/ pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
[20] R. Buyya, C. Yeo, and S. Venugopal, “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,” in Conference on High Performance Computing and Communications. IEEE, 2008.
[21] A. Newman, A. Steinberg, and J. Thomas, Enterprise 2.0 Implementation. McGraw-Hill Osborne Media, 2008.
[22] Amazon, “Amazon Elastic Compute Cloud (EC2),” Amazon Web Services LLC, Tech. Rep., 2009. [Online]. Available: http://aws.amazon.com/ec2/
[23] Mosso, “Deploy and scale websites, servers and storage in minutes,” Rackspace, Tech. Rep., 2009. [Online]. Available: http://www.mosso.com/
[24] R. Buyya, C. Yeo, and S. Venugopal, “Market-oriented Cloud Computing: Vision, hype, and reality for delivering it services as computing utilities,” in High Performance Computing and Communications. IEEE Press, 2008.
[25] Google, “Google App Engine: Run your web apps on Google’s infrastructure.” Google, Tech. Rep., 2009. [Online]. Available: http://code.google.com/appengine/
[26] T. Bain, “Is the relational database doomed?” ReadWriteWeb.com, 2008. [Online]. Available: http://www.readwriteweb.com/archives/ is the relational database doomed.php
[27] F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, “Bigtable: A distributed storage system for structured data,” in USENIX Symposium on Operating Systems Design and Implementation, 2006.
[28] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazon’s highly available key-value store,” in Symposium on Operating Systems Principles. ACM, 2007, pp. 205–220.
[29] B. Johnson, “Cloud Computing is a trap, warns GNU founder Richard Stallman,” The Guardian, Tech. Rep.,2008.[Online]. Available: http://www.guardian.co.uk/technology/2008/ sep/29/cloud.computing.richard.stallman
[30] J. McCabe, Network analysis, architecture, and design. Morgan Kaufmann, 2007. [Online]. Available: http://books.google.co.uk/ books?id=iddGPgR48 MC