IBIMA Publishing
Journal of Cloud Computing
http://www.ibimapublishing.com/journals/JCC/jcc.html
Vol. 2016 (2016), Article ID 930208, 9 pages
DOI: 10.5171/2016.930208

Research Article

Cloud Computing Distilled: What the Practitioner Needs to Know

Harvey Hyman

Library of Congress, National Library Service, Washington, DC, USA

Correspondence should be addressed to: Harvey Hyman; hymanphd@gmail.com

Received date: 10 May 2016; Accepted date: 9 June 2016; Published date: 1 September 2016

Academic Editor: Shahida Sulaiman

Abstract

While cloud computing has moved to the forefront of strategic IT initiatives in recent years, only a few articles have focused on the basic fundamentals, and none have been written with the practitioner in mind. This article presents a basic, yet comprehensive discussion on what cloud computing is, and how it works. The article identifies and describes the core concepts that an IT manager should know about cloud computing, and provides simple explanations for how the fundamental methods of cloud are used and their impacts upon business processes. This article divides the technology of cloud computing into four distinct categories: service models, delivery architecture models, virtualization and performance. We describe three service models of SaaS, PaaS and IaaS, three architecture models of public, private and hybrid cloud, explain how virtualization technology works, and discuss the current trends in performance factors of HA, FT, scalability, optimization, control and management.

Keywords: Cloud Computing, Virtualization, Hypervisors, Provisioning, Performance, High Availability, Fault Tolerance.

Introduction

Defining Cloud Computing

What is Cloud Computing? There are three general ways to define it. An operational definition of cloud computing is: "utility computing." A descriptive definition for cloud computing is: "service-based" or "resource-based" computing. A practical definition is simply: "pay as you go" computing (Armbrust et al., 2009). All of these descriptions of cloud are correct, and they reflect the two main impacts of cloud computing upon business: scalability and leverage (Malathi, 2011; Jadeja and Modi, 2012). We discuss these impacts later in the paper, as well as the business and individual advantages to using cloud computing.

In this article we treat the use of cloud computing in terms of "leveraging third party resources, communicated across a network" to support one's individual or organizational

Cite this Article as: Harvey Hyman(2016)," Cloud Computing Distilled: What the Practitioner Needs to Know", Journal of Cloud Computing , Vol. 2016 (2016), Article ID 930208, DOI:10.5171/2016.930208

computing needs. The goal here is to divide the use of cloud computing into four simple groupings: service models, delivery architecture models, virtualization, and performance. We describe and explain each grouping in the sections that follow. We begin with the three main models of cloud computing services: SaaS, PaaS, and IaaS (Malathi, 2011).

Cloud Delivery Models: SaaS, PaaS, IaaS

The general consensus starting point for a framework for discussing concepts in cloud computing is the 2009 article by Armbrust et al., entitled "Above the Clouds" (Nuseibeh and Alhayyan, 2014). While there were earlier articles (Vaqero et al., 2008; Reed, 2008), Armbrust et al., marked the first significant attempt to comprehensively define the emerging landscape coined as *cloud computing* by identifying developing categories in the technology, models, and services evolving as the core constructs that make up the "cloud." In the six years since the release of Armbrust et al., several studies have been working toward a consolidation of the domain constructs into a paradigm to guide academic research and industry practice (Nuseibeh and Alhayyan, 2014).

The first area of consolidation is in regard to service delivery models. For the past several years there has been an overabundance of descriptive service models. This has not been very helpful. Instead of providing a clear path for guiding researchers and practitioners alike, toward the most productive resources to focus upon, a plethora of service delivery models has led to a murky field of definitions and a glorified "thinking out loud" about the next would-be "potential of the day" in cloud computing services. This overabundance has culminated in the catch-all phrase XaaS or "everything as a service" – not very informative or focused. One might as well use the term **AaaS** – for "*anything* as a service."

The general consensus, both in academic and practitioner circles is that cloud service delivery models have been consolidated into three distinct categories: SaaS, PaaS, and IaaS.

SaaS, stands for Software as a Service. This delivery model is most commonly associated with the term *thin client* and software accessed via a web browser. Think of this model as *user-facing*. This is the choice of delivery for end-users who wish to access a hosted application such as common business applications. The operative description here is *software applications that are subscription based and internet hosted*.

The main significant factor with SaaS as a service delivery model is the centrality and control in distribution. Instead of deploying multiple copies of a software application, SaaS allows for a single copy to be accessed by the end-user. The specific advantages here are versioning and policy – both of which can be controlled almost instantly by updating the application host. In technical terms, SaaS supports a *multi-tenant architecture*. This means that all customers of the service use the same single version of the software application with the same single configuration of hardware, OS, and communication network. If we want to support more than one version of an application, another means of SaaS distribution would be the use of individual virtual machines (VMs), each supporting a different configuration or version of the application. The practitioner might see this in the form of virtual desktop image (VDI).

PaaS stands for Platform as a Service. Think of this model as *developer-facing*. In practical terms, it is really just a more robust variety of SaaS. This model is most commonly associated with providing development, deployment and maintenance support for a web-based application. You might also think of this model as a lifecycle approach – beginning with developing the application and ending with hosting and maintaining the application. This is the type of solution a business would use to deploy an application to be used as a SaaS by its customers. The practitioner should note that the PaaS model

is a supporting mechanism for the previous mentioned VDI deployment configuration.

Some technical analysts might distinguish SaaS from PaaS in terms of where the user's data live. For example, SaaS is sometimes viewed as merely providing processing, and the data itself begin and end on the user's local machine, whereas the "platform" in PaaS is viewed as providing the container for everything, including the residence of the user's data.

This distinction highlights a significant concern when relying on cloud computing for a business solution – when the connection is down, so is the service, there are no local copies as with the traditional client deployment model. There is also the potential for "data lock," not discussed in this article.
IaaS stands for Infrastructure as a Service. This model is the most aligned with the cloud definition of "utility computing." In this service delivery model, we are no longer focused on the individual end-user. In this model, the customer is defined as an organization, and the service is defined as computing resources. Think of this model as the configuration of a computing backbone that supports an entire enterprise. An IaaS provider supplies a pool of resources for the three computing components of CPUs, memory, and storage. Once again, the practitioner should note that a VDI deployment scheme will also likely follow an IaaS service model – in fact it will stretch across all three delivery models mentioned here. The technologies most closely associated with IaaS are virtualization, provisioning, and instances. These are discussed in the sections that follow.

In the interest of providing complete information on the subject of service delivery models, we need to acknowledge that over the past several years there have been numerous variations of service oriented computing such as IDaaS (Identification), BaaS (backend), STaaS (storage), EaaS (email, enterprise, everything). However, the industry has consolidated service models into the main three described above, with all other variations of service deliveries over the cloud having been merged into the main catch term **XaaS** – everything as a service, also eponymously written as *aaS or EaaS. A significant reason for this consolidation has been the move toward "micro-services" computing, whereby a large scale enterprise application is broken into smaller atomic modules, with each module encompassing a dedicated service. For example, user account authentication, credit card processing, and content search, would each be defined as separate services living on separate servers, and therefore, capable of individual software releases and scalable hosts.

How Does Cloud Computing Work? Virtual Machines and Hypervisors

What the IT manager needs to know is that there are three main components to a Cloud Computing solution: Virtual Machines (VMs), Hypervisors (VMM), and Hosts (servers). The remainder of this section will explain what these components are, the purposes they serve, and how they work.

A quick note to the reader here: Virtualization does not make the cloud work. A better way to understand it is that it allows the best features of the cloud to work. Specifically, virtualization allows for performance features of scalability, high availability, fault tolerance, optimization, management, and control. These features are explained later in the article.

Remember, cloud computing is a construct, meaning it is not a technique in and of itself. Instead, one should think of cloud computing as a *collection of computing technologies* that support the goal of resource-based, service-based, pay as you go, utility computing.

Remember also that, cloud computing is the ability to access computing services over a network. Cloud computing provides these services by making use of the core technology of virtualization.

The definition of **virtualization** is "software acting like hardware" or "software taking the place of hardware" or "software emulating hardware" or "software functioning as hardware" (Popek and Goldberg, 1974; Delgado et al., 2011). Take your pick, but the concept remains the same. The purpose of virtualization is to increase the capacity of physical hardware by creating multiple virtual environments (VMs) on top of them (Delgado et al., 2011).

The modern virtualization model focuses primarily on the virtual machine (VM) and the hypervisor (VMM), but, like cloud computing, virtualization is more akin to a collection of technologies than a specific technique itself. Meaning, the significance of virtualization is not limited to its application of the VM and the VMM. There are also vLANs to take the place of physical LANs, and vSwitches to take the place of physical switches. Both of which are examples of using software to take the place of hardware with the goal of supporting easier configuration, greater control, scalability, and increased capacity. For an excellent discussion on virtualization techniques applied across the enter enterprise resource pool see the reference paper by Vandenbeld and McDonald (2014).

It is important to inform the reader that the concepts of virtualization and the virtual machine date back to at least the 1950s and the use of mainframes; this was followed by exploratory work in the 1970s on "Grid Computing." (Popek and Goldberg, 1974; Delgado et al., 2011). For a good discussion on the 1970s view on virtualization and the VM, take a closer look at the Popek and Goldberg article.
A VM is an "isolated duplicate" (Popek and Goldberg, 1974) of a physical machine. In the practical sense, think of a VM as a separate, isolated container that provides an entire computing environment – this is also known as *provisioning* an *instance*, explained in more detail later.

The hypervisor, which has historically also been called the VMM or virtual machine monitor, also sometimes referred to as the "controller program," is the software application that supports the VM environment (Popek and Goldberg, 1974). The **hypervisor** is what allows a single server (also called the host) to support multiple guests (VMs). The number of VMs is only limited by the total number of physical resources available from the host. So, for example, if there are 20 CPUs available on the host, then up to 20 CPUs can be allocated to the various VMs to be created. An important thing to remember here is that there must always be some reserve for the host itself and for the hypervisor itself, as well as the VMs they are supporting (Delgado et al., 2011).

There are two models for running a hypervisor: **Type I and Type II**. In a Type I installation, the host is the bare metal server. In this case, the hypervisor "presents" the environment to the VM and serves all resource requests from the OS residing in the VM container (Vandenbeld and McDonald, 2014).

In a Type II installation, the hypervisor sits on top of the existing OS and relates resource requests to the underlying OS, which then fulfills the requests. A common example of this use case is an individual user who downloads an application as an appliance, and runs that appliance in a virtual machine environment (typical examples are Microsoft's "Hyper-V" or Oracle's Virtual Box) on their laptop or desktop.

Now an individual might find a type II hypervisor helpful, particularly in a situation where the user is presented with an *appliance* by the software manufacturer. An application as an appliance is a complete packaged version of the application that includes everything it needs to run on a bare minimum OS environment – hence why it is a very popular choice for individuals who prefer an application that comes preconfigured for a virtual environment (think a VM wizard).

There is a recent commercial addition to the Type II hypervisor model: *Docker* and *Containers*. This paper will not get into "Docker" or the "Docker Engine" but the practitioner should be aware that this product is out there and that the concept of "containers" is gaining momentum in the domain of cloud computing and VMs. In particular, a type II hypervisor model is very handy for presenting a testing environment for a development team that may wish to install specific versions of an OS or a legacy application without impacting the native host or larger resource pool.

Likewise, moving along the software lifecycle process phases, (from development to testing to staging to production), the trend is shifting the software paradigm toward the use of "containers," with orchestration tools and repositories, to migrate developer's source code from phase to phase. Virtualization and the use of cloud computing resources are the main drivers supporting this trend.

Types of Clouds: Public, Private, Hybrid

Public, private, and hybrid clouds are three leading types of cloud architectures that have emerged as the main conceptual descriptions for the deployment of cloud computing configurations.

In the past several years there have been other variations of cloud architecture offered to explain ad hoc configurations such as *community cloud, federated cloud, distributed cloud, inter-cloud,* and *multi-cloud*. However, for the most part, the industry has consolidated cloud provider models to the main three described here.

The **public cloud** describes a computing architecture whereby the user's instance is drawn from a shared pool of resources. This is also called multi-tenancy. The main advantage here is for the small business user who wants to "spin up" or "tear down" a web based computing application or service, and does not require specific hardware or software configurations or have particular security concerns. The public cloud is most closely associated with the "pay as you go" business model for cloud computing. The security aspect is particularly important to note here given that the user's instance is based on the shared pool of resources and not dedicated to the user as would be in a single tenant model. This model is associated with managed solutions commonly supplied by Google, Amazon (under the brand name AWS), IBM (under the brand name Softlayer), and Microsoft (under the brand name Azure).

The **private cloud** describes a computing architecture whereby the hardware, storage and communication network is dedicated to the organization. This is called single tenant. This model is associated with users who require custom configurations, have specific hardware, software or network requirements, have higher level security concerns such as HIPAA or other compliance issues, and desire greater control and management of their computing resources. This model is not a "pay as you go" model due to the dedicated nature of the hardware and software pool. However, there has been an increasing trend on the part of cloud providers (such as Google, Amazon, Microsoft and IBM) to provide a "virtual private cloud" whereby a resource pool is dedicated for scalability and performance, but still has some elements of "pay as you go" pricing.

The **hybrid cloud** is more of a scalability solution than an architectural model. Hybrid cloud covers situations whereby an organization requires the flexibility to temporarily expand their computing needs such as bursting requirements for increased CPU processing power, storage, or memory allocation.

In this use case, an organization may have a custom configuration of dedicated hardware, software and network communication, but requires additional resources to temporarily scale up a service or process. The organization can achieve this temporary increase in scale by extending into the public cloud. This allows the organization to take

advantage of the "pay as you go" approach for the additional scale up, and release those resources when no longer needed, all the while maintaining their custom configuration to meet their unique business needs.

This model is also a possible solution for organizations who feel the need to keep part of their services located on premise, due to particular security concerns yet still be able to take advantage of higher scale delivery service performance from cloud resources.

Provisioning and Instances

What is Provisioning? Simply put, provisioning is the allocation of computing resources. The resources are configured as an **instance**, which is the specific operational deployment of computing resources to support a particular business process (Hossny and Khattab 2012; Chieu et al., 2010). A good use case example is an organization that wants to run a Hadoop application. The cloud provider will "provision" an "instance" of a Hadoop cluster, or "spin up a VM" configured for this purpose. This allows the organization to horizontally scale the Hadoop application, using as much compute, memory and storage as needed.

When we think of provisioning an instance from a pool of resources, we begin to think of this implementation as a computing cluster (Awad et al., 2014). A cluster is comprised of nodes. Think of nodes as individually configured CPU, memory and storage elements combined in a manner to support the required business process. This is an example of optimization, whereby a provisioned instance is matched to a specific performance requirement. This is discussed in more detail in a later section.

The cluster approach differs from the Grid model in that each node of a cluster is running its own instance. It is this simple distinction that allows for the support of features known as HA and FT – also associated with functionalities called load balancing and fail over.

High Availability (HA)

High Availability (HA) is a design method for system continuity by switching services over to alternative hosts in the case of server or hardware failure, generating a new VM in the case of software or OS failure, or generating a new node in the case of a node failure (Singh et al., 2012).

The main thing to understand when discussing HA is that services are restored quickly, but not instantaneously (IBM Knowledge Center). However, the vast majority of applications and services will appear to be seamless to the end user. HA is often associated with minimizing downtime due to maintenance, upgrades, and software application failures (the most common source of failure). In the use cases of maintenance and upgrades, HA is used to "migrate" a VM to another available host during the planned down period.

Two additional things to know here: (1) In order for HA to work, there needs to be enough resources available to host the VMs needing to migrate, and (2) in the case of an application failure, there is a moment of unavailability (however short), until the VM is restarted (this is reboot time).

If services are so critical that the end user cannot accept even a momentary lapse, then FT offers an added level of robustness.

Fault Tolerance (FT)

Fault Tolerance (FT) is a design method intended to achieve no interruption in service (IBM Knowledge Center). This topic is currently still ripe for continued system testing and benchmarking.

The main thing to understand here is the tradeoff between robustness and cost. FT is achieved through redundancy. In the case of HA, as long as resources are available a VM can be restarted (the delay is in the required boot time). FT by comparison not only reserves a redundant amount of resources to

absorb the failure, but also sustains a shadow copy of the VM, thereby maintaining a primary and secondary VM (VMware's vSphere Availability Guide).

An example of this is the feature known as vLockstep offered by VMware (VMware's vSphere Availability Guide). This feature maintains simultaneous writes to a primary and to a secondary VM. When the primary VM fails, the secondary continues on. As far as the customer is concerned, nothing has happened. The main technical difference here is that service continuity occurs without the need for a reboot (IBM Knowledge Center).

When it comes to HA and FT, the main tradeoff is one of economic cost versus service level performance – the increased cost for the additional resources that are on standby, waiting to be used, "just in case." This becomes a significant economic as well as architectural issue that must be considered by the stakeholders of the service in terms of how much risk they are willing to accept, the level of service they are committed to providing, and the cost they are willing to absorb.

Performance: Scalability, Optimization, Management and Control

Current trends in cloud computing are exploring performance factors in scalability, optimization, management, and control. These factors are used to assist organizations when choosing the type of cloud service model (SaaS, PaaS, IaaS), architecture model (Public, Private, Hybrid), and continuity model (HA, FT) best suited for their business processes.

When we discuss scalability in the cloud, we are referring to the ability to scale up as well as to scale down dynamically and elastically (Vaquero et al., 2011). Scaling up refers to the ability to add resources to handle additional workloads without reducing performance. Scaling down refers to the ability to release or eliminate resources when workloads are reduced.

Under the physical model for computing, increased scalability is achieved through the time consuming acquisition, installation and configuration of hard assets. When workloads are reduced, idle computing assets sit unused, waiting to be tasked – costing money.

The cloud model (applying virtualization) allows for dynamic scaling of resources. The advantage here is the ability to instantly respond to increased workload demands and yet, only pay for what is needed, when it is needed. Unused assets are released. In this use case, "elasticity rules" are applied to establish the optimal use of computing resources for an organization's needs (Rochwerger et al., 2009).

The practical definition of optimization is *the ability to match resources to workload needs as closely as possible*. In the realm of cloud computing this refers to *the dynamic adjustment of provisioned resources* to meet the exact needs of the business process at any given moment. Optimization is a constant exercise in the avoidance of two problems: *overprovisioning* and *underprovisioning* (Chaisiri et al., 2012).

Overprovisioning refers to the problem of having more computing resources than needed, resulting in idle assets.

Underprovisioning refers to the problem of having too few computing resources available for the current workload, and may result in reduced performance below service level agreements (SLAs), outages, or even complete system failures due to lack of compute, memory or storage.

Cloud management refers to the "fundamental support of users of cloud services" (Lonea, et al., 2012). From the practical IT management point of view, this issue refers to how much direct control the organization wants to exert on its cloud solution. This is a strategic IT decision that will impact the organization's choice of

service model (SaaS, PaaS, IaaS) and delivery architecture (Public, Private, Hybrid).

The degree of control to which the organization wishes to maintain over its computing resources is largely a factor of balancing the preference of having portions of computing resources, or even the entire computing infrastructure, managed by a provider versus maintaining those resources by internal personnel.

Issues effecting this strategic calculation normally include: the level of expertise on hand, whether the organization's IT solution is starting from scratch or is migrating an existing solution, the ability to control policy and protocols, specific regulatory requirements, unique security concerns, and the organization's culture and comfort with risk and third party provided services.

Conclusion

This article sets out to present a discussion on the underlying concepts of cloud computing. This article is intended to provide a light summary on the subject matter, with a significant reference list that the reader may use to better inform themselves in this domain. The article points out the most common factors and issues that an IT manager will be confronted with at their organization, based on recent and predicted trends.

Acknowledgements

The author wishes to thank computer engineer Henry Chao and CIO Thomas Hull, at Florida Polytechnic University for their support and contributions that made this article possible.

The content and opinions expressed in this article are solely that of the author, and are not endorsed in any way by the Library of Congress or National Library Service. No support or resources of the Library of Congress or National Library Service were utilized to produce this work.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M., (2009), "Above the Clouds: A Berkeley View of Cloud Computing," UC Berkeley Reliable Adaptive Distributed Systems Laboratory, http://radlab.cs.berkeley.edu.

2. Awad, O.M.O., Artoli, A.M.A., Ahmed, A.H.A., (2014), "Cloud computing versus in-house clusters: a comparative study," World Congress on Computer Applications and Information Systems (WCCAIS).

3. Chaisiri, S., Lee, B. S., Niyato, D. (2012). "Optimization of resource provisioning cost in cloud computing," Services Computing, IEEE Transactions on, 5(2), 164-177.

4. Chieu, T. C., Mohindra, A., Karve, A. A., Segal, A., (2010), "A Cloud Provisioning System for Deploying Complex Application Services," 7th International Conference on e-Business Engineering (ICEBE).

5. Delgado, J., Salah-Eddin, A., Adjouadi, M., Masoud-Sadjadi, S., "Paravirtualization for Scientific Computing: Performance Analysis and Prediction," (2011), IEEE International Conference on High Performance Computing and Communications.

6. Hossny, E., Salem, S., Khattab, S. M., (2012), "Towards automated user-centric cloud provisioning: Job provisioning and scheduling on heterogeneous virtual machines," 8th International Conference Informatics and Systems (INFOS).

7. IBM Knowledge Center. Cited as: http://www-01.ibm.com/support/knowledgecenter/SSPHQG_6.1.0/com.ibm.hacmp.concepts/ha_concepts_fault.htm.

8. Jadeja, Y., Modi, K., (2012), "Cloud Computing - Concepts, Architecture and Challenges," International Conference on

Computing, Electronics and Electrical Technologies (ICCEET).

9. Lonea, A.M., Popescu, D.E., Prostean, O., (2012) "A survey of management interfaces for eucalyptus cloud," IEEE 7th International Symposium on Applied Computational Intelligence and Informatics (SACI).

10. Malathi, M., (2011), "Cloud Computing Concepts," 3rd International Conference on Electronics Computer Technology (ICECT).

11. Nuseibeh, H., Alhayyan, K., (2014), "Trends in the Study of Cloud Computing: Observations and Research Gaps," Proceedings of the 5th International Multi-Conference on Complexity, Informatics, and Cybernetics, (IMCIC/ICSIT, 2014), pages 38 – 43.

12. Popek, G. J., Goldberg, R. P., (1974), "Formal Requirements Virtualizable Third Generation Architectures," Communications of the ACM, Volume 17, Number 4.

13. Reed, D.A., (2008) "Clouds, clusters and ManyCore: The revolution ahead," Conference on Cluster Computing, IEEE International.

14. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Galan, F. (2009). "The reservoir model and architecture for open federated cloud computing," IBM Journal of Research and Development, 53(4), 4-1.

15. Singh, D., Singh, J., Chhabra, A., (2012) "High Availability of Clouds: Failover Strategies for Cloud Computing using Integrated Checkpointing Algorithms," International Conference on Communication Systems and Network Technologies.

16. Vaquero, L. M., Rodero-Merino, L., Caceres, J., Lindner, M. (2008) "A break in the clouds: towards a cloud definition." ACM SIGCOMM Computer Communication Review, 39(1), 50-55.

17. Vaquero, L. M., Rodero-Merino, L., Buyya, R. (2011) "Dynamically scaling applications in the cloud," ACM SIGCOMM Computer Communication Review, 41(1), 45-52.

18. Vandenbeld, M., McDonald, J., (2014), "VCA-DCV Official Cert Guide." VMware Certified Associate Data Center Virtualization, VMware Press.

19. VMware's vSphere Availability Guide. Cited as: http://pubs.vmware.com/vsphere51/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-51-availability-guide.pdf.