Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Cloud Computing Distilled: What the Practitioner Needs to Know

Harvey Hyman[1]

[1] Library of Congress

Received: 13 December 2015 Accepted: 2 January 2016 Published: 15 January 2016

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.

Index terms— cloud computing, virtualization, hypervisors, provisioning, performance, high availability, fault tolerance.

1 I. Introduction: Defining Cloud Computing

hat 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 [1]. All of these descriptions of cloud are correct, and they reflect the two main impacts of cloud computing upon business: scalability and leverage [2], [3]. 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 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 [2].

II. 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" [1], [4]. While there were a few earlier articles [5], [6], theirs 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." Over last 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 [4].

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 had 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. This is explained in greater detail later.

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. Some technical analysts might distinguish SaaS from PaaS in terms of where the user's data lives. For example, SaaS is sometimes viewed as merely providing processing, and the data itself begins and ends 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. 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 XaaSeverything as a service, also eponymously written as *aaS or EaaS.

2 III. 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 purpose they serve, and how they work.

A quick note to the reader here: Virtualization does not make the cloud work, but it allows for the best features of cloud to be provided. Specifically, virtualization allows for scalability, high availability, fault tolerance, optimization, management, and control. These features will be 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, 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" [7], [8]. 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 [8].

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 reference [9]. Now, for a little history on the origins of the concept of virtualization, early work on virtualization and the use of virtual machines (VMs). The concept of virtualization and the virtual machine dates back to, at least, the 1950s and the use of mainframes, followed by work in the 1970s on "Grid Computing." [7], [8]. See the referenced Popek and Goldberg article for a good discussion of the 1970s view on virtualization and the VM.

A VM is an "isolated duplicate" [7] 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, [7] also sometimes referred to as the "controller program," [7] is the software application that supports the VM environment. The hypervisor is what allows a single server (also called the host) to support multiple guests (VMs) [7]. The number of VMs is only limited by the total number of physical resources available from the host [8]. 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 [8].

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 [9].

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 free virtual machine (typical examples are MS Hyper-V or Oracle Virtual Box) on their laptop or desktop. An application as an appliance is a complete package version of an 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). A type II hypervisor model is also 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.

IV. 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, 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 multitenancy. 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 common supplied by AWS and 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 whom 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.

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 increased CPU for processing power, added storage, or additional 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.

3 V. 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 [10], [11]. 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 [12]. 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.

4 VI. 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 [13].

The main thing to understand when discussing HA is that services are restored quickly, but not instantaneously **??**14]. 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 the VM to another available host during the planned down period.

Two additional things to know here. One, in order for HA to work, there needs to be enough resources available to host the VMs needing to migrate. Two, 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.

5 VII. Fault Tolerance (ft)

Fault Tolerance (FT) is a design method intended to achieve no interruption in service [14]. This topic is currently still ripe for continued system testing and bench marking.

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 [15].

An example of this is the feature known as vLockstep offered by VMware [15]. This feature maintains simultaneous writes to a primary and 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 the service continuity occurs without the need for a reboot **??**14].

The main tradeoff when it comes to HA and FT is the increased cost for the additional resources, just sitting there, waiting to be used. 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 provide, and the cost they are willing to absorb.

6 VIII.

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 scale down dynamically and elastically [16]. 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 taskedcosting 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 [17].

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 [18].

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" [19]. 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, 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.

7 IX. Conclusion

This article set out to present a discussion on the underlying concepts of cloud computing, and point out the most common factors and issues that an IT manager will be confronted with, based on current trends.

The author wishes to thank Henry Chao, Thomas Hull and the IT support staff at Florida Polytechnic University for their support and contributions that made this article possible. [1] [2]

[2] Year 2016 ( ) B Cloud Computing Distilled: What the Practitioner Needs to Know

7 IX. CONCLUSION

[Ibm Knowledge and Center] , Ibm Knowledge , Center . http://www-01.ibm.com/support/knowledgecenter

[Vaquero et al. ()] 'A break in the clouds: towards a cloud definition'. L M Vaquero , L Rodero-Merino , J Caceres , M Lindner . *ACM SIGCOMM Computer Communication Review* 2008. 39 (1) p. .

[Chieu et al. ()] 'A Cloud Provisioning System for Deploying Complex Application Services'. T C Chieu , A Mohindra , A A Karve , A Segal . *7th International Conference on e-Business Engineering (ICEBE)*, 2010.

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

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

[Jadeja and Modi ()] 'Cloud Computing -Concepts, Architecture and Challenges'. Y Jadeja , K Modi . *International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, 2012.

[Malathi ()] 'Cloud Computing Concepts'. M Malathi . *3 rd International Conference on Electronics Computer Technology (ICECT)*, 2011.

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

[Reed ()] 'Clouds, clusters and ManyCore: The revolution ahead'. D A Reed . *Conference on Cluster Computing*, 2008. IEEE International.

[Vaquero et al. ()] 'Dynamically scaling applications in the cloud'. L M Vaquero , L Rodero-Merino , R Buyya . *ACM SIGCOMM Computer Communication Review* 2011. 41 (1) p. .

[Popek and Goldberg ()] 'Formal Requirements Virtualizable Third Generation Architectures'. G J Popek , R P Goldberg . *Communications of the ACM* 1974. 17 (4) .

[Singh et al. ()] 'High Availability of Clouds: Failover Strategies for Cloud Computing using Integrated Checkpointing 14. Algorithms'. D Singh , J Singh , A Chhabra . *International Conference on Communication Systems and Network Technologies*, 2012.

[Chaisiri et al. ()] 'Optimization of resource provisioning cost in cloud computing'. S Chaisiri , B S Lee , D Niyato . *IEEE Transactions on* 2012. 5 (2) p. . (Services Computing)

[Delgado et al. ()] 'Paravirtualization for Scientific Computing: Performance Analysis and Prediction'. J Delgado , A Salah-Eddin , M Adjouadi , S Masoud-Sadjadi . *IEEE International Conference on High Performance Computing and Communications*, 2011.

[Rochwerger et al. ()] 'The reservoir model and architecture for open federated cloud computing'. B Rochwerger , D Breitgand , E Levy , A Galis , K Nagin , I M Llorente , F Galan . *IBM Journal of Research and Development* 2009. 53 (4) p. .

[Hossny et al. ()] 'Towards automated user-centric cloud provisioning: Job provisioning and scheduling on heterogeneous virtual machines'. E Hossny , S Salem , S M Khattab . *8th International Conference Informatics and Systems (INFOS)*, 2012. VMware Press.

[Nuseibeh and Alhayyan ()] 'Trends in the Study of Cloud Computing: Observations and Research Gaps'. H Nuseibeh , K Alhayyan . *Proceedings of the 5 th International Multi-Conference on Complexity, Informatics, and Cybernetics*, (the 5 th International Multi-Conference on Complexity, Informatics, and Cybernetics) 2014. 2014. IMCIC/ICSIT. p. .

[Vandenbeld and Mcdonald ()] *VCA-DCV Official Cert Guide*, M Vandenbeld , J Mcdonald . 2014. VMware Certified Associate Data Center Virtualization.

[VMware's vSphere Availability Guide] *VMware's vSphere Availability Guide*, http://pubs.vmware.com/vsphere51/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-51-availability-guide.pdf