Network and Services Monitoring: A Survey in Cloud Computing Environments

Guilherme da Cunha Rodrigues,
Vinicius Tavares Guimaraes,
Glederson Lessa dos Santos
Federal Institute of Education,Science and
Technology Sul Rio-Grandense
Charqueadas, Brazil
Emails: {grodrigues,vicoguim,gledersons}@charqueadas.ifsul.edu.br

Lisandro Zambenedetti Granville,
Liane M. Rockenbach Tarouco
Federal University of
Rio Grande do Sul

Porto Alegre, Brazil
Emails: {granville,liane}@inf.ufrgs.br

***Abstract*—Cloud Computing are currently seen as a trend in the computing area for companies, institutions and research groups that seek provision of computational resources on demand. This solution usually works out by means of virtual systems which are managed through a specific infrastructure, termed as Infrastructure-as-a-Service or IaaS in the case of cloud computing. In an attempt to identify the network monitoring resources in use, this paper aims at presenting features of some Framework for Cloud Computing operating at the IaaS level.**

Keywords-Cloud Computing; IaaS; Network Monitoring.

I. Introduction

The development of activities in several areas of human knowledge is increasingly dependent on computational resources. These resources must have a minimum of capacity in order to perform these tasks and meet some requirements imposed by the market, such as low cost, availability, flexibility and type of resources.

Hence, concerns with the availability of these resources began in the 80's, with the first Local Area Network (LAN), Wide Area Network (WAN) and high performance computer networks, technologies which have made possible the emergence of connections between computational systems. These systems have been defined as distributed systems, i.e., an independent group of computers that shows itself to their users as a single and coherent system that has as its main characteristics shared resources, transparency, scalability, availability, and flexibility [1].

Thus, computer networks and distributed systems have been developed, and from them other technologies have emerged, which aim to meet these demands, such as grid computing and, more recently, cloud computing.

Nowadays, the concept of Cloud Computing has emerged as a trend in the computer area to professionals, companies and institutions that are concerned both with flexible computational resources with high rates of performance and availability and with the use of hardware devices in rational and sustainable ways [2]. These systems aim to provide services and can be showed at several levels of abstraction, such as software, platform and infrastructure. For example, when it comes to service-oriented systems the term "Software as a Service (SaaS)" has been found in several papers, like Michel Head's 2010, in which he observes that "Cloud computing is a paradigm of computing that offers virtualized resources 'as a service' " [3].

Cloud computing systems usually have a business focus, e.g., resources lease. Some of its characteristics have become important, of which the most prominent appears to be non-functional guarantee provisioning. Applications can thus be executed by considering predefined standards, such as run-time, operations costs, security, privacy, among others. Such guarantees are currently specified in Service Level Agreements (SLA) [4] [5]. However, it will be necessary an effective management in order to maintain the SLAs in a cloud computing environment.

Currently, it is notorious, the main topics in management resources are monitoring and controlling devices; therefore, which activities should be defined and consistent on systems that support cloud computing systems. Thus, this paper aims to demonstrate about the state of art in network monitoring resources for cloud computing systems.

This remaining of this paper is organized as follows. Section 2 presents current technologies in monitoring, such as Simple Network Management Protocol (SNMP) and Management via Web Services. Section 3 presents current systems on Cloud Computing. Section 4 presents the some Frameworks to Cloud Computing. Section 5 presents the network monitoring in Cloud Computing. Finally, conclusions and future works are show in Section 6.

II. Monitoring

A management protocol aims at providing the basis for monitoring and controlling resources. The management process can be divided, according to the OSI management standard, in five functional areas as it follows: fault management, configuration management, accounting management, performance management, and security management [6][7]. Each functional area has the purpose of defining the focus of action of monitoring and controling.

 ISBN: 978-1-61208-183-0

With the growth of computational systems, the demand for better administration methods also increased. The use of Cloud Computing Systems is consequentely affected by such trend and, therefore, the characteristics of each cloud must also be taken into consideration when managing such environments.

In this context, the utilization of management protocols becomes an important tool. Currently, we can cite three relevant management options: Simple Network Management Protocol (SNMP) [6] [8], Network Configuration Protocol (NETCONF) [9] and Management Systems via Web Services [10].

A. SNMP

The SNMP protocol was originally developed for network management. However, due to its flexibility it may be used for other types of management applications. The structure of this protocol is based on managers and agents, where the agents are spread on the resources and the management operations (e.g., read or write of objects) are performed by the manager through direct solicitations to agents. The objects that can be managed are described in a MIB (Manageament Information Base) [11].

The MIB serves as reference for tracking objects that are part of the system with the aim of getting information about resources. It contains a hierarchical structure set according to a numeral sequence [12].

The Simple Network Management Protocol (SNMP) is currently used to refer a collection of specifications for network management, which includes the protocol itself (SNMP), the specification to describe management objects (SMI - Structure of Management Information), and the management objects (MIB). All the operations supported by SNMP are related to reading/writing management objects. The main operations available are the following:

- Get: this operation is used to read an object value.
- Set: this operation is used to write object values.
- Get-Next: this operation is used to read the value of the next available object.
- Trap: this operation is used for communicating special events.

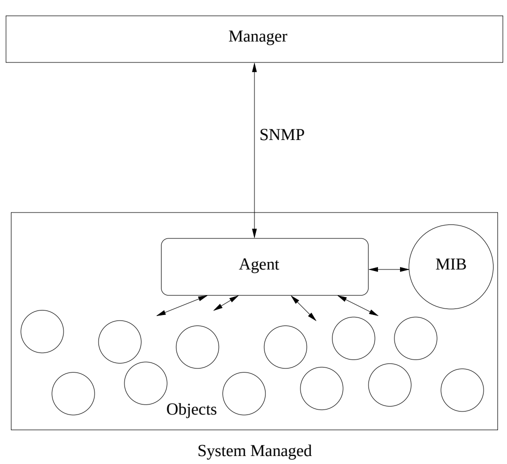

Fig. 1. SNMP Architecture.

The amount of operations is limited, making the protocol simple, flexible and easy to implement. Such operations are used between manager and agents.

The architecture of the SNMP is shown in Figure 1, where it is possible to see the manager and agent interacting. In this figure, the manager sends solicitations using SNMP to the agent, then it observes the reference within the MIB, it accesses the object and it sends a reply to the manager using SNMP.

The development of the agent follows the features of the managed system, being one extension of system. We can notice that the features on the system influence directly the complexity for agent development.

Because of these characteristics, we can define that SNMP is effective to monitoring resources, but is limited to controlling. Thus, there is a need for others technologies. For this, currently, we find the Netconf and Management via Web Services.

B. Web Services

In general, Web Services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks [13]. Based on this concept, Web Services emerge with different potentials in the network management context. According to Klie et al. [14], in the network management community, many people see Web Services as a possibility to solve some of the most important problems:

- First of all, since Web Services are platform independent and use standard Internet protocols, they help to deal with the heterogeneity.
- Second, they offer a unified communication model for network, application, and management systems.
- Third, with Web Service composition mechanisms, automation can be supported.

Vianna et al. [15] highlights that the motivation for using Web Services in the network management area is that these technologies in fact address problems that SNMP investigations have been trying to solve in years. Basically, a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Figure 2 shows the general process of engaging a Web service, emphasizing the involved entities.

As shown, the requester and provider entities become known to each other. The requester and provider entities somehow agree on the service description and semantics that will govern the interaction between the requester and provider agents. The service description and semantics are realized by the requester and provider agents. Finally, the requester and provider agents exchange messages, thus performing some task on behalf of the requester and provider entities.

The core technologies of Web Services include Simple Object Access Protocol (SOAP) [16], Web Service Definition Language (WSDL) [17] and Universal Description Discovery and Integration (UDDI) [18], they are expressed in the standard form of XML documents and are built XML-based

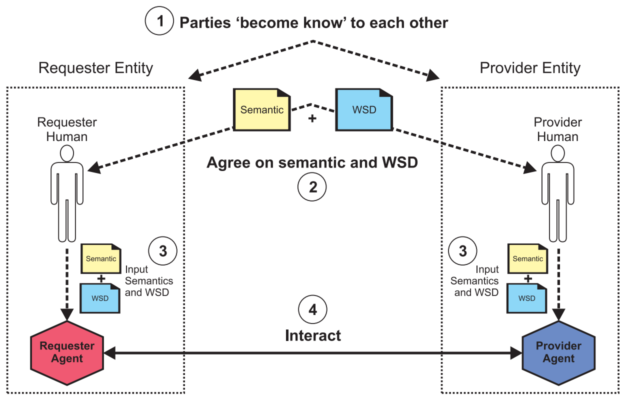

Fig. 2. The general process of engaging a Web service [13].

specification, this ensures the platform-independent, language-independent and human-computer interaction of the architecture [9]. Basically, SOAP is the communication protocol, WSDL describes the service, how to access it and the available operations, while the UDDI allows publish and discover of Web Services directories.

In the network management context, the main standards are the Management Using Web Services (MUWS) [10] by the Organization for the Advancement of Structured Information Standards (OASIS) and the Web Services for Management (WS-Management) [19] by the Desktop Management Task Force (DMTF). Nowadays, specifically in the cloud computing context, the use of Web Services is presented as a potential alternative given the heterogeneity and diversity of environments that composes this new paradigm. This way, the Amazon Web Services (AWS) [20] is a popular example.

III. Cloud Computing

Nowadays, there are many different definitions to Cloud Computing. Rajkumar Buyya defines it as a paralel and distributed system composed of clustered virtual machines interconnected which are allocated dynamically and presented as a unified system of computing resources based on SLAs estabilished through business trading between service provider and clients [21].

On the other hand, Ian Foster understands that the paradigm of cloud computing is not necessarily a new concept, but the result of a symbiosis among different paradigms, such as: grids, clusters, distributed systems and others [22].

This paper does not aim at discussing different concepts as to what cloud computing is. However, in order to present a focus to this work, some definitions need to be taken into account such as types of cloud computing (Deployment models) and service models.

A. Types of Cloud Computing (Deployment Models)

Computing clouds can be classified according to their creation, use and purpose. They may come from the need of short-time computing resources within an organization, which in this context, utilizes resources from another business partner. This classification may also come from underutilization of resources within an enterprise, generating an opportunity to lease its computing availability on demand to a third party through a cloud. The peculiarities described above result in consolidated types of cloud computing described as: Private, Public, Community and Hybrid Cloud [23].

B. Service Models

The concept of cloud computing encompasses different levels of computing services. They may vary from allocated applications within a cloud to making storage resources available and data processing. Such levels are regarded as service models as it follows [24]:

- Software as a Service (SaaS): applications of interest to a large amount of users may be hosted within a cloud as an alternative to local processing. Applications are offered as a service by providers and accessed by clients through applications such as a browser. Managing and controlling of networking infrastructure, operational systems, servers and storage is done by the service provider. Google Apps is a example of SaaS.
- Platform as a Service (PaaS): the structure provided to the user by the provider to develop applications which are going to be executed and made available within clouds. On that regard, another concept emerges known as Utility Computing, used to denominate the whole support platform to the developing and providing of applications in clouds.
- Infrastructure as a Service (IaaS): it is the capacity a provider has to offer a processing infrastructure and storage in a transparent way. The user does not have control of the physical infrastructure, but through virtuliazation mechanisms it is possible to control operational systems, storage, installed applications and possibly a limited control of network resources.

Such virtualized infrastructure must follow some guidelines which enable an effective resource management. Such effectiveness relies on [25]:

- Providing a uniform and homogenous vision of the virtualized resources, regardless of the platform, be it Xen, KVM or Vmware, for example.
- Managing a VM life cycle, including networking communication configurations, in a dynamic way for virtual machine clusters.
- Managing system resource storage.
- Adjustable support to allocation of resources in order to cater for the specific needs of each and every enterprise which comprises or utilizes the cloud, for each one uses a cloud according to its needs, such as: availability, server consolidation, decrease in energy, among others.
- Adapting the organization in order to choose necessary resources, including peaks where the local ones do not cater for. The new choice of resources should include subsidies taking into account the addition of new physical resources, in a dynamic way, or containing a buffer zone tolerant of faults originated by physical resources.

 ISBN: 978-1-61208-183-0

The characteristics above should be presented by frameworks which enable the managing of virtual resources in cloud computing systems. Currently we can use the Amazon EC2 [26], Eucalyptus [27], OpenNebula [28] and Nimbus [29] as examples of available utility computing clouds as an IaaS.

IV. Frameworks

A. Amazon EC2

The Amazon Elastic Compute Cloud (Amazon EC2) is a private solution for Cloud Computing. This solution was a pioneer on solutions to cloud, thus, become reference for others frameworks.

The Amazon EC2 is mainly characterized by [26]:

- Resource and instances storage capacity in different and distributed allocations.
- Safe cloud computing environment.
- Automatic load scheduling and balancing.
- The ability to import external virtual machines.

B. Eucalyptus

Eucalyptus is a framewrok designed for cloud computing systems which operate on IaaS level. This system enables users and cloud computing system administrators the manipulation of virtual machines through functionalities such as creation, control and finalization [27].

It was developed aiming at the broadening of academic studies of cloud computing, it is one of the first systems of this kind and it has an open code and its structure adapted to a wide range of resources which utilize physical infrastructure (processing, storage, network, among others). It is usually found and available within academic research groups.

In this regard, it is worth pointing out that the functional struture of Eucalyptus is flexible and modular, enabling possible adaptations to the system which aim at best suiting it to testing scenarios, experiments, analysis or studies.

By being modular, Eucalyptus is formed by components which interact among themselves through interfaces. This system modularity enables it to be altered, updated and perfected.

Apart from being modular, Eucalyptus also has a hierarchical structure (as described in the Architecture subsection), which facilitates the usage of resources available in labs, clusters, workstations and servers.

1) Architecture: The architecture of Eucalyptus is simple, flexible and modular. The structure of the system is hierarchical and it is presented with a friendly operational environment. The system enables functionalities such as beginning, access, control and ending of VMs, using a presentation system similar to Amazon's EC2 [30].

By definition, implementing each component of high level in the system is a stand alone web service. Such definition brings some benefits such as:

- Each web service is presented with a defined language from an API (Application Programming Interface) in the shape of a WSDL (Web Service Description Language) containing the operations that the service is able to execute and the structures of an in/out database.
- The system enables the implementation of security policies regarding the communication among its components.

Eucalyptus is formed currently by three components of high level, Instance Manager (IM), Group Manager (GM) and Cloud Manager (CM).

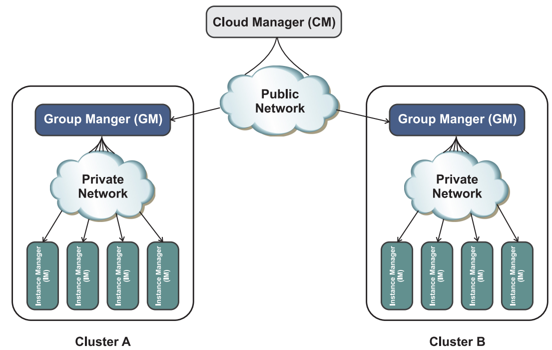

Fig. 3. Eucalyptus Architecture.

Figure 3 shows Eucalyptus Architecture, highlighting the high-level components.

C. OpenNebula

OpenNebula is a virtual structure manager which can be used to deploy and manage virtual machines (VM) or manage VM clusters that can be co-scheduled in local resources or even in external public clouds.

The automatic configuration of virtual machines, the preparation of disc images and the communication network configuration among other functionalities occur independently of the virtualized platform (e.g., Xen [31], KVM [32], Vmware [33]) or the external cloud (e.g.,EC2); therefore, the Opennebula works independently of the virtual system being used [34].

The platform independence is one of the main characteristics of Opennebula, being only possible given the modularity of the system. This is one of the peculiarities of Opennebula.

This framework has functionalities similar to the ones found in similar systems, its focus, however, lies on the attempt to fulfill some gaps still found in other frameworks for cloud computing systems. Among these gaps we can highlight the following [25]:

- The inability to scale external clouds.
- Monolithic and closed architectures that are hard to extend or interface with other applications, not allowing seamless integration with existing storage and network management solutions deployed in data centers.
- A limited choice of preconfigured placement policies, such as First Fit and Round Robin.
- A lack of support for scheduling, deploying and configuring groups of VMs (for example, a group of VMs representing a cluster, where all the nodes either deploy entirely or do not deploy at all, and where some VMs configuration depends on others, such as the headworker relationship in compute clusters).

 ISBN: 978-1-61208-183-0

1) Architecture: The architecture of the Opennebula has modular and specialized components which execute functions to fulfill requirements regarding virtual infrastructure management.

In this regard, the virtual machine life cycles is executed by Opennebula's core, which manipulates three managing modules: Image and storage technology module, Network factory and a thrid module called Underlying Hypervisor's [25].

Among these three modules, we highlight the network factory, which is composed of virtual network equipment like DHCP servers, firewalls and switches, among others equipments. They provide for the VMs an virtual communication network environment.

The Opennebula core communicates with the storage devices, network and virtual systems through the so-called pluggable drivers, so that Opennebula is not bound to a specific environment, providing a uniform managing to the underlying infrastructure layers.

Apart from managing virtual machine life cycles, Opennebula's core has deployment support, which includes interconnected component clusters, such as web services and requested data base in several virtual machines. That way a group of virtual machines can be treated as first class entities in Opennebula. Additional to the managing of virtual machines as a single unit, the core can also be presented with information regarding the context, such as digital certificates and virtual machine software licensing.

A resource scheduler, usually Haizea, determines how the virtual machine allocations are going to be accomplished [25]. More specifically, the scheduler has access to all information about Opennebula requests and based on them it determines future and current allocations, creating and updating resource programming and sending the deployment commands to the Opennebula core.

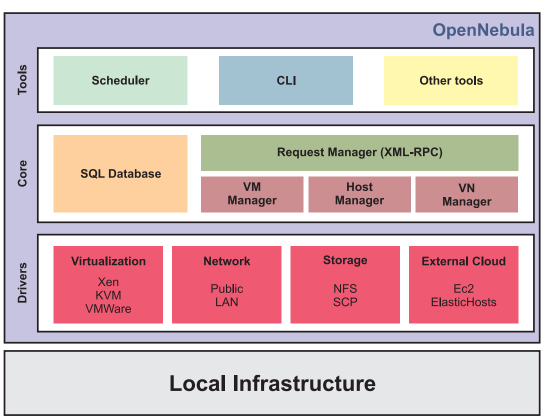

Fig. 4. OpenNebula Architecture.

Figure 4 shows the Opennebula's architecture and its components, highlighting the core, the Haizea scheduler and the drivers which interconnect the system modules and external clouds.

D. Considerations

We will hold a brief comparison between Eucalyptus and Opennebulla, for this, we defined some functions and topics will be analyzed in both systems, such as, Architecture, Virtual Systems, Virtual Infrastructure Management and Allocation and Resources Allocation.

1) Architecture: We understand architecture as organization structure for operation of framework, in this way, both, Eucalyptus and Opennebulla have diferent organization forms. The architecture of Eucalyptus is hierarchical and have three high level components. The components communicate with each other through networks public and private. On the other hand, Opennebula have a modular architecture, because uses the modular resources (e.g., drivers, cores) to interoperate between them.

2) Virtual Systems: Clouds are composed by sets of virtual machines. Thus, frameworks for cloud computing need have the capacity to manage virtual machines. Nowadays, we found several systems to deploy the virtualization, such as, Xen and KVM, for example. In this context, a good framework need provide support for these systems. The Eucalyptus, currently, have support to Xen, KVM, and VMware, among others. Additionally, Eucalyptus, have support to Virtual Machine Manager (VMM). The Virtual Machine Manager is a device that allow operate and manage a virtual infrastructure over virtual machine (e.g., memory, hard disk). On the other hand, Opennebula have support to several virtual systems, such as, Xen, KVM and Vmware, among others, but, we can not found reference to VMM support.

3) Virtual Infrastructure Management and Allocation: Virtual infrastructure management and allocation in cloud is related with framework capacity to manipulated a virtual machines sets. These virtual machines, that interoperates in a virtual environments, such as, virtual networks.

Both, Eucalyptus and Opennebula have capacity to begin, access, manage and kill virtual machines, individually or in groups. In addition, have support to management of virtual networks and SLAs standards. However, the Opennebula have a capacity to scheduling a virtual machine hosted in a external cloud. Currently, the Opennebula have drivers for utilization on frameworks, such as, Amazon EC2, and Eucalyptus, for example.

4) Resource Allocation: Resource Allocation (e.g., memory, hard disk) on clouds is related with framework capacity to manipulate these resources in virtual machines and work with SLAs. The Opennebula uses a scheduler, usually Haizea, to do it. On the other hand, the Eucalyptus uses yours hierarchical architecture to make this.

V. Monitoring in Cloud

A. Monitoring in Eucalyptus

As seen before, the Eucalyptus system enables users and administrators to manipulate an infrastructure of physical and virtual resources through open code solutions for cloud computing systems.

 ISBN: 978-1-61208-183-0

This system is hierarchically structured and has three components of high level which communicate among themselves through communication networks.

In order to achieve that, Eucalyptus utilizes public and private network structures, being the latter heterogeneous and comprised of physical and virtual network devices. The implementation of these virtual networks is achieved through Virtual Distributed Ethernet (VDE), a system which has the ability of creating virtual network according to ethernet standards and provides the use of switches, cables and other virtual network devices for the Eucalyptus.

It is also worth mentioning that the communication between group and instance managers is provided by a private network influencing directly on the communication among the final components of a system. This peculiarity is vital to the load balance within a network, given the adequate structure of a communication network has direct influence on the system's performance altogether.

A virtual network is created from physical devices that host virtual ones which posses as main characteristics isolation capability and migration, which enable the definition of different and flexible use scenarios, cost reduction and effective security policies.

In this respect, it is understood that the effective monitoring of these networks is of crucial importance to framework researches and the own perfecting of it, as well as any other cloud computing system.

The current managing standards such as SNMP protocol, Netconf and managing via Web Services can be applied to Eucalyptus as a whole, although studies which evaluate its performance, efficency and application of these standards have not yet been found in cloud computing system.

The avouchment above highlight the fact that it is still incipient the task of monitoring network resources in Eucalyptus, such notions can only be proved through research, studies and experiments which can evaluate specifically and proficiently these types of solutions.

B. Monitoring in OpenNebula

As observed before, the Opennebula system has a flexible and modular architecture and it is able to manage virtual machines individually or conjoined. This system possesses a core and a scheduler which work with virtual support systems (e.g., Xen, Vmware, among others), storage devices and virtual communication networks, and it is also capable of scheduling external clouds if necessary.

In this regard, virtual network image devices are stored in the so-called network factory, devices which are allocated and deallocated according to momentary communication needs.

Such communication needs may come from within a cloud, which occur between virtual systems initiated and set in motion to cater for applications defined by the core and scheduler in Opennebula.

The monitoring of virtual network devices which serve to the virtual systems are compatible with the solutions already mentioned in this paper, such as SNMP, Netconf and managing through Web Services, given the fact that virtual systems can support these managing solutions, as well as the monitoring of virtual devices.

As the Eucalyptus, it is understood that the virtual monitoring activity is very important to the perfecting of Opennebula, although no research, studies or experiments regarding it have been found, as well as currently in respect to this framework.

VI. Conclusion and Future Work

The cloud computing emerges as an old dream of computer science, called utility computing. From this perspective, customers, companies, government institutions, among others, start to migrate their applications, platforms and infrastructure to the cloud, creating new types of pricing, services, resources use, etc.

This way, virtualization is the most important assumption to achieve the goals addressed by this new paradigm. Specifically, virtualization can enable dynamism and scalability to the cloud, maximizing the potential to services offering and optimizing resources allocation by providers. In the other hand, this flexibility brings an inherent complexity to infrastructure orchestration and management, since constitute an increasingly heterogenous enviroment.

In this work, we investigated the cloud monitoring process in the main cloud plataforms. We can observe that the concerns are more focused on the cloud offering models, i.e., the convergence in a model, which directs the efforts to management tasks, still seems remote. Obviously, we can diagnose easily the well-known management challenges, but the innovations introduced by the cloud require specific researches.

Based on this context, we verify the need for well-defined mechanisms to monitoring process. Such mechanisms should provide the features for low-level metrics measurement (e.g., measure the rate of memory and processor usage in an array of devices that make up the infrastructure for a particular application) and the capacity to diagnose high-level behaviors, such as quality of services from customers point-of-view.

As future works, we intend to investigate the applicability of traditional monitoring mechanisms in order to diagnose the strengths and weaknesses of them in the cloud enviroment. From this point, we aim propose a basic monitoring framework to cover the different models of service provided in the cloud.

References

[1] A. S. Tanenbaum and A. S. Woodhull, *Operating Systems: Design and Implementation*, 2nd ed. Bookman, 2000.

[2] B. Hayes, "Cloud computing," *Commun. ACM*, vol. 51, pp. 9–11, July 2008. [Online]. Available: http://doi.acm.org/10.1145/1364782.1364786

[3] M. R. Head, A. Kochut, C. Schulz, and H. Shaikh, "Virtual hypervisor: Enabling fair and economical resource partitioning in cloud environments," in *NOMS*, 2010, pp. 104–111.

[4] I. Brandic, "Towards self-manageable cloud services," in *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 02*, ser. COMPSAC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 128–133. [Online]. Available: http://dx.doi.org/10.1109/COMPSAC.2009.126

[5] G. Reig, J. Alonso, and J. Guitart, "Prediction of job resource requirements for deadline schedulers to manage high-level slas on the cloud," *Network Computing and Applications, IEEE International Symposium on Cluster Computing and the Grid*, vol. 1, pp. 162–167, 2010.

[6] W. Stallings, *SNMP,SNMPv2,SNMPv3 and RMON1 and RMON2*, 3^a ed. Addison-Wesley, 1996.

[7] R. Stephan, P. Ray, and N. Paramesh, "Network management platform based on mobile agents," *Int. J. Netw. Manag.*, vol. 14, no. 1, pp. 59–73, 2004.

[8] M. A. Miller and Atwood, *Managing Internetworks with SNMP; The Definitive Guide to the Simple Network Management Protocal, SNMPV2, RMON, and Rmon2*, 2nd ed. Hungry Minds, Incorporated, 1997.

[9] D. Xiaochao and W. Jinpeng, "Netconf network management model based on web services," in *IT in Medicine Education, 2009. ITIME '09. IEEE International Symposium on*, vol. 1, Aug. 2009, pp. 160 –164.

[10] W. Vambenepe and V. Bullard, "Web services distributed management: Management using web services (muws 1.1) part 1," Available http://docs.oasis-open.org/wsdm/wsdm-muws1-1.1-spec-os-01.htm, Nov, 2011.

[11] Y.-C. Chen and I.-K. Chan, "SNMP getrows: an effective scheme for retrieving management information from MIB tables," *Int. J. Netw. Manag.*, vol. 17, no. 1, pp. 51–67, 2007.

[12] M. T. Rose, "Management information base for network management of tcp/ip-based internets: Mib-ii," 1990. [Online]. Available: http://rfc.net/rfc1158.html

[13] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web services architecture," 2004. [Online]. Available: http://www.w3.org/TR/ws-arch/, Sep, 2011.

[14] T. Klie, T. U. Braunschweig, A. Belger, T. U. Braunschweig, L. Wolf, and T. U. Braunschweig, "A peer-to-peer registry for network management web services," 2009.

[15] R. Lemos Vianna, M. Almeida, L. Tarouco, and L. Granville, "Investigating web services composition applied to network management," in *Web Services, 2006. ICWS '06. International Conference on*, Sep, 2006, pp. 531 –540.

[16] J. Snell, D. Tidwell, and P. Kulchenko, *Programming Web services with SOAP*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002.

[17] *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[18] S. Ran, "A model for web services discovery with qos," *SIGecom Exch.*, vol. 4, pp. 1–10, March 2003. [Online]. Available: http://doi.acm.org/10.1145/844357.844360

[19] D. M. T. Force, "Web services for management specification," Available http://dmtf.org/sites/default/files/standards/documents, Nov, 2011.

[20] Amazon, "Amazon web services (aws)," Available http://aws.amazon.com, Nov, 2011.

[21] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, pp. 599–616, June 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1528937.1529211

[22] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," *2008 Grid Computing Environments Workshop*, vol. abs/0901.0, no. 5, pp. 1–10, 2008. [Online]. Available: http://arxiv.org/abs/0901.0131

[23] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[24] L. M. Vaquero, L. Rodero-merino, J. Caceres, M. Lindner, and T. I. Y. Desarrollo, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, pp. 50–55, 2009.

[25] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, pp. 14–22, 2009.

[26] "Amazon elastic compute cloud: Ec2," http://aws.amazon.com/documentation/ec2/. Last access Sep. 2011.

[27] "Eucalyptus cloud," http://www.eucalyptus.com/. Last access Sep. 2011.

[28] "Opennebula project," http://www.opennebula.org. Last access Sep. 2011.

[29] P. Marshall, K. Keahey, and T. Freeman, "Elastic site: Using clouds to elastically extend site resources," *IEEE International Symposium on Cluster Computing and the Grid*, pp. 43–52, 2010.

[30] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, ser. CCGRID '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 124–131. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2009.93

[31] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauery, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of SOSP'03*, 2003.

[32] The KVM Project Home Page, "KVM-kernel based virtual machine," Available http://www.linux-kvm.org, Nov, 2011.

[33] THE VMWARE Project Home page, "The vmware project," RFC 1832. Available http://www.vmware.com, Nov, 2011.

[34] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Capacity leasing in cloud systems using the opennebula engine," *Cloud Computing and its Applications, CCA´08*, pp. 1–5, 2008.