University of Castilla-La Mancha

A publication of the

Computing Systems Department

FLEXIBLE ADVANCE-RESERVATION (FAR) FOR CLOUDS

by

Jose Luis Lucas, Carmen Carrión, Blanca Caminero

Technical Report **#DIAB–10-08-01** Agosto. 2010

This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by JCCM under Grants PBI08-0055-2800 and PII1C09-0101-9476.

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete - 02071 - Spain
Phone +34.967.599200, Fax +34.967.599224

Flexible advance-reservation (FAR) for Clouds *

Jose Luis Lucas, Carmen Carrión, Blanca Caminero
Computing Systems Department. Escuela Politécnica Superior
Universidad de Castilla-La Mancha. 02071 - Albacete, SPAIN
{jllucas,carmen,blanca}@dsi.uclm.es

August 27, 2010

Abstract

Cloud computing has gained popularity in recent times. Nowadays several companies are migrating their software towards cloud providers. Using virtual machines as a resource provisioning mechanism offers some benefits, but depending on the applications it could be difficult to preview the exact amount of resources the company needs to provide QoS to its clients. In this paper we propose a new way of booking resources in which the Cloud user can set min and max levels to cope with peak-load. For this proposal we need to modify some components of the cloud infrastructure: a new module for the booked virtual machine, some changes in advance reservation leases from cloud manager and some background for SLA layer.

*This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by JCCM under Grants PBI08-0055-2800 and PII1C09-0101-9476.

1 Introduction

Cloud computing is an emerging topic in the field of parallel and distributed computing. A cloud is a combination of physically and virtually connected resources that aim to power the next generation data centers by architecting them as a network of virtual services [1].

Virtualization is one of the key technologies behind the Cloud computing infrastructure. Virtualization allows us to instantiate virtual machines dynamically on physical machines and allocate them as needed. There are several benefits that we expect from virtualization, such as high availability, ease of deployment, migration, maintenance, and low power consumption that help us to establish a robust infrastructure for Cloud computing [2].

With Cloud computing, companies can lease resources on-demand from a virtually unlimited pool. That resources can be divided into three parts: infrastructure, platform, and software, which are usually made available as subscription-based services in a pay-as-you-go model to consumers. These services are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) in industry. The pay-as-you-go billing model applies charges for the actually used resources per unit time. This way, a business can optimize its IT investment and improve availability and scalability.

Typical computing resources offer a static and finite set of computational capacity to users. Initially, a resource is purchased with an estimate for its peak capacity with the hope that the average load on the resource stays well below that estimate. However, the demand is dynamic, and it is then difficult to estimate the times when the demand will exceed the capacity of the resource (e.g, social networking environment). Most often it is only realized when the system crashes under the load and this results in user

frustration [3]. So a challenge for Cloud providers is how to provide a solution to cope with clients failed estimations.

Several research groups have explored the way of taking advantage of the elasticity that provides Cloud computing. For example,[4] [3] show the benefits of expanding a cluster with EC2 nodes; or [5] shows the use of Cloud computing for Grid resource provisioning. Moreover, there are some companies which offer elastic cloud services for an amount of money, like ElasticHosts [6]. Thus, the target application for our proposal would be one that exhibits variable loads. The user can just identify the range of VMs needed to support the expected load. Then, the system will adjust autonomously the resources allocated to that user depending on the real load.

More precisely, in our proposal we present a extension to the advance-reservation lease for taking advantage of the elasticity provided by Cloud computing. It will allow users to manage their booked flexible resources for improving their business investment. Moreover, this new feature will probably generate new proposals of scheduling policies (due to the new type of lease), smart pricing algorithms, lease priority algorithms and so on...

The remainder of the paper is organized as follows. In Section II we outline the cloud environment we have used for this work. In Section III we discuss our proposal of flexible advance-reservation lease. In Section IV we present some preliminary experiments we have done for testing the usefulness of the proposal. Finally, the paper ends with some conclusions in Section V, where we hypothesize possible guidelines for future work.

2 Cloud environment

Figure 1 depicts the typically Cloud computing architecture. It includes three layers: *Infrastructure layer*, *Platform layer* and *Software layer*. The *infrastructure layer* can be divided into three parts: *Physical layer*, where isolated physical resources are placed; *Hypervisor layer*, which allows each physical resource to be virtualizated; and *Cloud management layer*, which provides a general view of the system. On top of that, the *Platform layer* is where virtual machines are placed. And finally, the *Software layer*, which consist on applications running into those virtual machines.

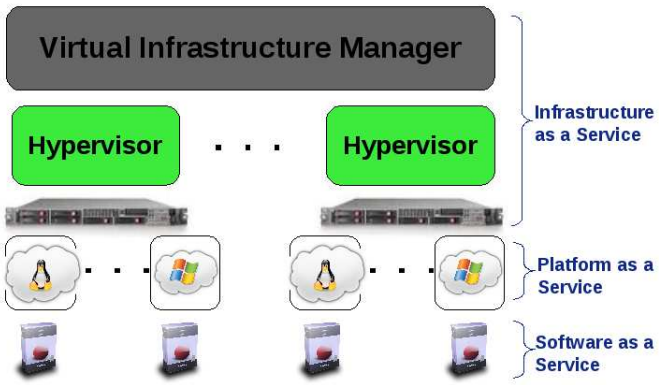


Figure 1: Cloud computing architecture.

The authors have focused on OpenNebula [7] as cloud management software, instead of other well-known frameworks such as Eucalyptus [8] or Nimbus [9], due to the facts that it is one of the most extended cloud manager all over the world, it can be extended and integrated with third-party developments, and its scheduler can be easily replaced. OpenNebula is a virtual infrastructure engine which provides the functionality needed to deploy, monitor and control VMs on a pool of distributed physical resources. OpenNebula is composed of three main components: *The OpenNebula Core*, a centralized component than manages the life-cycle of a VM (deploy, monitor, migrate, ...); *the Capacity Manager* is a module that governs the functionality provided by the

OpenNebula core; *the Virtualizer access Drivers*, than expose the basic functionality of the hypervisor in order to provide an abstraction of the underlying virtualizartion layer. OpenNebula is able to dynamically scale-out this infrastructure by interfacing with an external cloud. In fact, OpenNebula already includes an Amazon EC2 [10] virtualizer driver [11].

Moreover, we have used Haizea [12] as scheduling software, instead of the OpenNebula default scheduler, because it provides a more complete scheduling environment. Haizea is an open source lease management architecture developed by Sotomayor et al. [13] that OpenNebula can use as a scheduling software. Haizea uses leases as a fundamental resource provisioning abstraction. A lease is a "negotiated and renegotiable agreement between a resource provider and a resource consumer, where the former agrees to make a set of resources available to the latter, based on a set of lease terms presented by the resource consumer"[14]. Currently, Haizea supports three types of lease: *advance reservation lease*, where the resource must be available at a specific time:; *best-effort leases*, where resources are provisioned as soon as possible; and *immediate leases*, where resources are provisioned when requested or not at all [13].

3 Flexible advance reservation leases

Our proposal consist on extending the Cloud manager infrastructure for optimizing user cost-benefit ratio and improving the use of resources for variable load applications. Currently, an application provider can book a number of virtual machines during a period of time. However, during the execution, the application which runs into the VM could get overloaded. In order to try to overcome this, we propose to add a range of extra virtual machines which could be used in case the application needs them. A typical use case could be described as follows: The client wants to keep required QoS,

to save some money and not to waste part of their booked resources. They could purchase enough VMs for an average load and a flexible range for a peak load. This possibility would be cheaper than purchasing enough VMs for the whole leasing period for only coping with a unpredictable small peak-load period. In order to support flexible

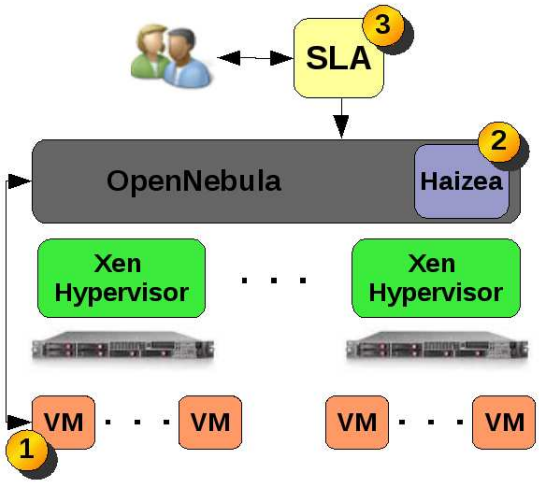


Figure 2: Steps for supporting flexible advance-reservation leases.

advanced-reservation leases, several changes need to be introduced in the generic Cloud infrastructure as depicted in Figure 2.

First of all, an "smart-component" needs to be created. This component needs to be placed in the booked virtual machine and it is able to connect to OpenNebula's node. Figure 3 shows an activity diagram which models its behavior. First of all, a client asks OpenNebula for a virtual machine using an flexible advance-reservation lease. Both parties negotiate an SLA, and ONE gives the VM to the client. After that, depending on the action to do, if the current system is collapsed the component would ask for a new virtual machine to help the system. On the other hand, if the current system is working better than it was previewed, it would tell OpenNebula to delete one existing virtual machine. This procedure works as a loop within the booked period of time. Moreover, the component has the ability to automate the dynamic provisioning of VMs taking into account the negotiated SLA.

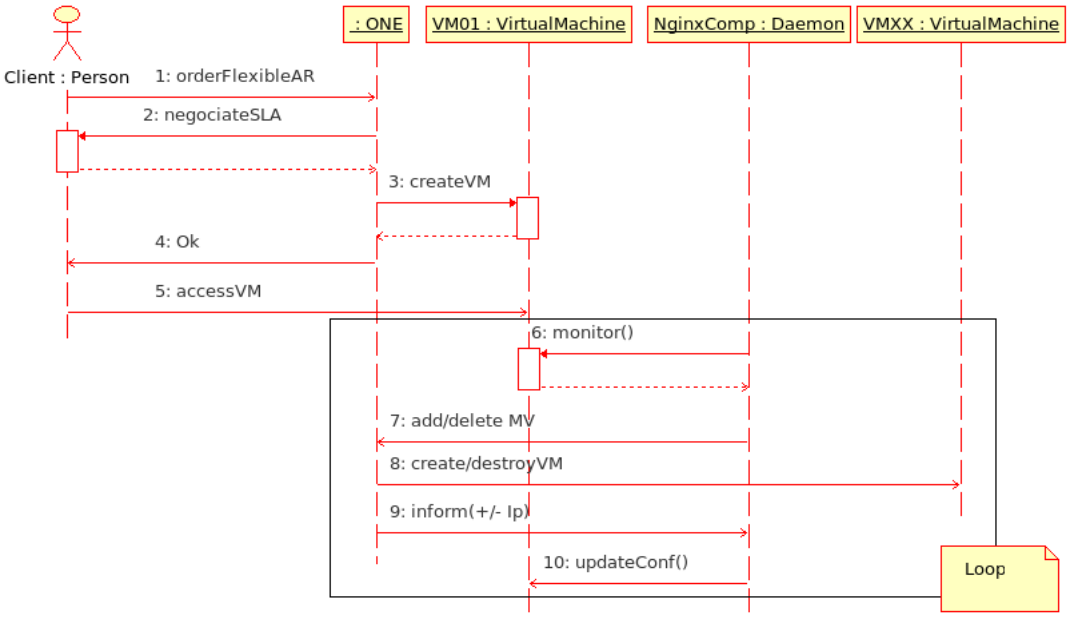


Figure 3: Flexible advance-reservation scenario.

Next, the Haizea MetaScheduler has to be extended to consider the new type of lease. As it has said earlier, Haizea has three types of leases: *best-effort* leases, *immediate* leases and *advance-reservation* leases. We focus our idea on the third one, in which we propose to extend its characteristics for including our flexible feature: a range of virtual machines. Moreover, Haizea would need some control of this flexible feature in order to know when is the system empty of resources to provide or if there are still enough resources to fulfil a client request. Here, some heuristic methods can be used for helping while extending Haizea software.

Finally, background for new SLA type should be added so that clients can use the new type of lease. SLA are negotiated between two parties following the specific negotiation protocols, and are generated using the SLA templates available on both the consumers and the providers sides. With this new feature Cloud computing providers would need new ways to generate SLA templates which include the flexible advance-reservation.

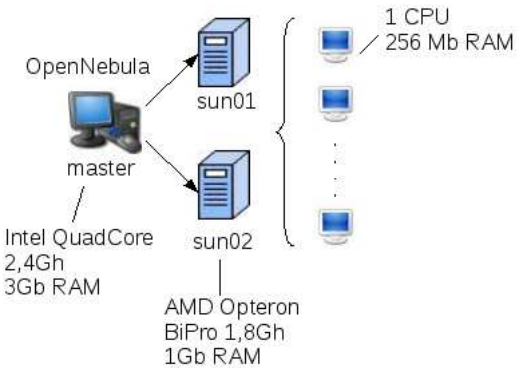


Figure 4: Overview of the infrastructure used for this work.

4 Preliminary results

In order to have an insight into the foreseen benefits of our proposal, we have devised a proof-of-concept experiment which shows the system behavior when VMs are automatically provisioned and freed on workload fluctuations. The physical infrastructure used in this work consist on one central host in which OpenNebula is installed, and two more hosts where VMs will be deployed. In both hosts we have used Xen Hypervisor [15] to manage our VMs. As can be observed in Figure 4, the virtual machines have a single core 1.8 GHz AMD Opteron processor and 256 MB of RAM, and all of them are connected to the Internet.

We suppose there is a client who has booked one virtual machine for a period of time. This VM has the Nginx web server and PHP installed, and also some PHP scripts which use CPU time.

For testing that virtual machine we have designed a variable workload. Having into account than our arrival rate is one request per second, we have created three types of requests: *low request*, which takes less than a second to finish, *high request*, which takes more than a second, and *very high request*, which takes much more than one second. We have mixed those types of requests in a period of time, creating our

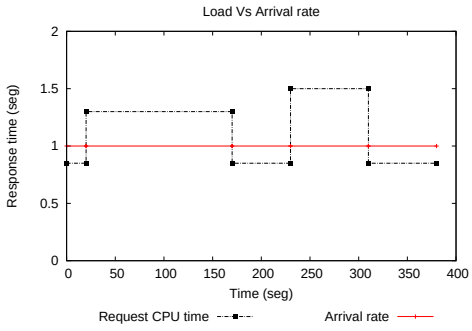
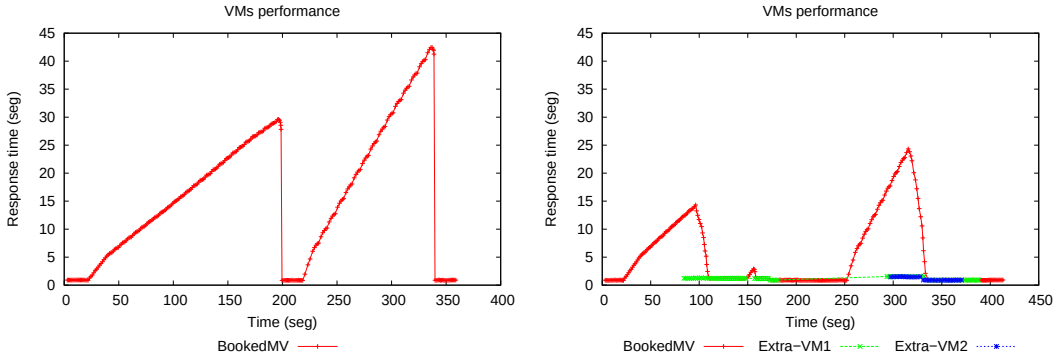


Figure 5: Defined workload for testing the system.

test workload. As Figure 5 shows, we start with low requests being followed with high request which load the system. After that, it comes another easy low-load period before introducing the highest load requests to the system, and ending with another low request period. Our purpose is to simulate a real workload that could need more resources than the booked virtual machine gives us. That is a possible situation that would arise when there has been a bad booking.

Here we present some graphics which show the evolution of our system performance without and with our proposal. Looking at Figure 6.a we can observe the system performance under the specified workload without our proposal. It can be observed that the system gets overloaded with high requests, because lots of them are being queued. When introducing low requests, the system is most of the time overloaded due to queued requests. Then, while introducing very high and then low requests, the system shows the same performance as before.

With our proposal we could avoid those problems just booking a range of virtual machines in the same period of time. As we can see in Figure 6.b when our main virtual machine gets overloaded, our component asks OpenNebula for another virtual machine for helping it. In this point, the first peak load has been fixed. Later, in the low part of the workload, the component knows that the system is working fine and it decides to delete the extra VM.



(a) Without FAR lease.

(b) Using FAR lease.

Figure 6: System performance.

When the very high part of the workload appears, the component asks for one extra VM and also it immediately asks for another one for preventing the system to get overloaded. When this part finishes and the second low part arrives,the system works fine again and the component deletes those extra virtual machines.

5 Conclusion and Future work

In this work we propose an extension to advance-reservation leases in order to fix bad booking decisions, which could make the system get overloaded with the corresponding loss of QoS. We create a smart-component which monitors the VM booked and decide if it needs another VM or if the system works fine. We propose some extensions in Haizea software to add this new feature to the current ones. Finally, we evaluate an example of workload without and with that proposal to have a view of the benefits of the automatic and dynamic provision of resources on applications demand.

Our current implementation is an initial prototype focused specifically on Nginx web server. As future work we will extend the Haizea code for adding the new type of lease. Also we will test several algorithms for improving the smart component performance.

References

[1] 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 Generation Computer Systems*, June 2009.

[2] P. Barham *et al.*, "Xen and the Art of Virtualization," *SOSP, Symposium on Operating Systems Principles*, 2003.

[3] P. Marshall, K. Keahey, and T. Freeman, "Using Clouds to Elastically Extend Site Resources," *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing (CCGrid), Melbourne, Australia,*, 2010.

[4] M. Dias de Assuncao, A. di Costanzo, and R. Buyya, "Evaluating the cost-benefit of using Cloud computing to extend the capacity of clusters," *High Performance Distributed Computing*, 2009.

[5] I. Llorente, R. Moreno-Vozmediano, and R. Montero, "Cloud Computing for on-Demand Grid Resource Provisioning," *Advances in Parallel Computing, IOS Press (in press)*, 2009.

[6] "Elastic Hosts - Flexible servers in the cloud," June 2010. [Online]. Available: http://www.elastichosts.com/

[7] "OpenNebula home page," June 2010. [Online]. Available: http://www.opennebula.org/

[8] D. Nurmi, R. Wolski, and C. Grzegoczyk, "The Eucalyptus Open-source Cloud-computing System," *In Proceedings of Cloud Computing and Its Applications*, 2008.

[9] K. Keahey and T. Freeman, "Science Clouds: Early Experiences in Cloud Computing for Scientific Applications," *Cloud Computing and Its Applications*, October 2008.

[10] "Amazon Elastic Compute Cloud (EC2)," June 2010. [Online]. Available: http://aws.amazon.com/ec2/

[11] B. Sotomayor, R. Santiago Montero, I. MartÃn Llorente, and I. Foster, "Capacity Leasing in Cloud Systems using the OpenNebula Engine," *Workshop on Cloud Computing and its Applications*, 2008.

[12] "Haizea - An Open-Source VM-Based Lease Manager," June 2010. [Online]. Available: http://haizea.cs.uchicago.edu/

[13] B. Sotomayor, R. Santiago, I. MartÃn, and I. Foster, "Resource Leasing and the Art of Suspending Virtual Machines," *11th IEEE Intl. Conf. on High Performance Computing and Communications*, 2009.

[14] B. Sotomayor, K. Keahey, and I. Foster, "Combining Batch Execution and Leasing Using Virtual Machines," *HPDC,High-Performance Distributed Computing*, 2008.

[15] "Xen Hypervisor," June 2010. [Online]. Available: http://xen.org/