FLEXIBLE ADVANCE-RESERVATION (FAR) FOR CLOUDS*

José Luis Lucas
Department of Computing Architecture, Complutense University of Madrid, Madrid, Spain

Carmen Carrión, Blanca Caminero
Department of Computing Systems, University of Castilla-La Mancha, Albacete, Spain

Keywords: Quality of service, Reservations in advance, Virtual machine leasing.

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, a new way of booking resources is proposed, in which Cloud users can specify the minimum and maximum number of virtual resources needed, so that coping with periods of peak load is easier and cheaper. This is supported by means of a general framework which includes modifications and/or additions of several components at different levels of the Cloud architecture. A microbenchmark-based proof-of-concept evaluation is included, which provides a hint on the benefits of the proposal.

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 (Buyya et al., 2009).

Virtualization is one of the key technologies behind the Cloud computing paradigm. Virtualization allows the dynamic instantiation of virtual machines on physical resources, and to allocate them as needed. There are several benefits expected from virtualization, such as high availability, ease of deployment, migration, maintenance, and low power consumption that help to establish a robust infrastructure for Cloud computing (Barham et al., 2003).

With Cloud computing, companies can lease resources on-demand from a virtually unlimited pool. Those 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. A resource is initially 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 often 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 (Marshall et al., 2010). 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, in (Dias de Assuncao et al., 2009) and (Marshall et al., 2010) the benefits of expanding a cluster with EC2 nodes are shown; in (Llorente et al., 2009) the use of Cloud computing

*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.

Luis Lucas J., Carrión C. and Caminero B..
FLEXIBLE ADVANCE-RESERVATION (FAR) FOR CLOUDS.
DOI: 10.5220/0003383306100615
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 610-615
ISBN: 978-989-8425-52-2

for Grid resource provisioning is proposed. Moreover, there are some companies which offer elastic cloud services for an amount of money, like ElasticHosts (Web, 2010b).

The target application for the proposal presented in this paper would be one that exhibits variable loads. The user (i.e., the application provider) can identify the range of Virtual Machines (VMs) needed to support the minimum and maximum expected load. Then, the system will autonomously adjust the resources allocated to that user depending on the real load, up to the specified maximum number of VMs, and allocating always the minimum number of VMs.

More precisely, an extension to the advance-reservation lease model implemented in the Haizea scheduler (Sotomayor et al., 2008) named **FAR (Flexible Advance Reservations)**, is presented, aimed at taking advantage of the elasticity provided by Cloud computing. This extension 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 rationale behind this is similar to the one used to accept a number of Variable Bite Rate (VBR) connections into communication links. VBR connections are characterized by a mean and a peak bandwidth (this is similar to the range of VMs pointed out above). If the peak connection bandwidth is reserved, link bandwidth will be wasted because connections do not transmit at their peak data rate all the time. Translated to our scenario, if the maximum number of VMs is provisioned, it is likely that great part of the time many of them will not be used, thus wasting provider's resources (i.e., energy, CPU power), and maybe preventing other users to be admitted into the system. Thus, several connections can be admitted provided that it is quite unlikely that their "peaks" will happen simultaneously (due to the Law of Large Numbers). In the Cloud scenario, this means that it will be unlikely that all users will need the full range of VMs at the same time. Of course, in both scenarios, some criteria must be applied in order to know if a new connection (user) can be admitted or not, while satisfying its requirements, and not compromising those from already admitted connections (users).

The cost of this type of reservation is intended to be lower that the cost of reserving the maximum number of VMs in the conventional way. So the Cloud provider would get less benefits from every single user, but as a counterpart, it could admit more users into the system. From the users perspective, they could opt for paying a more economic bill for a set of virtual resources, that will very likely be available when needed.

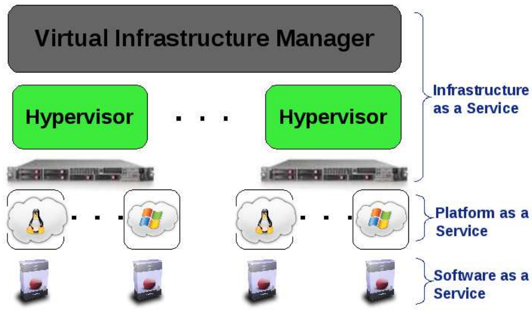


Figure 1: Cloud computing architecture.

The remainder of the paper is organized as follows. In Section 2 the cloud framework used for this work is outlined. In Section 3 the proposal of flexible advance-reservation leases is introduced and discussed. In Section 4 some preliminary experiments that illustrate the usefulness of the proposal are presented. Finally, the paper ends with some conclusions in Section 5, where possible guidelines for future work are pointed out.

2 CLOUD ENVIRONMENT

Figure 1 depicts the typical Cloud Computing architecture. It includes three layers: *Infrastructure Layer*, *Platform Layer* and *Software Layer*. The *Infrastructure Layer* can be further divided into three parts: *Physical Layer*, where isolated physical resources are placed; *Hypervisor Layer*, which allows each physical resource to be virtualized; 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 consists on applications running on those virtual machines.

The authors have focused on OpenNebula (ONE) (Sotomayor et al., 2009a) as cloud management software, instead of other well-known frameworks such as Eucalyptus (Nurmi et al., 2008) or Nimbus (Keahey and Freeman, 2008), because 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. It is composed of three main components, namely, *OpenNebula Core*, a centralized component that manages the life-cycle of a VM (deploy, monitor, migrate, ...); *Capacity Manager*, a module that governs the functionality provided by the OpenNebula core; and *Virtualizer Access Drivers*, that exposes the basic functionality of the hypervisor to provide an abstraction of the underlying virtualization layer. OpenNebula is able to dynamically scale-out this infrastructure by interfacing with an external cloud. In fact, OpenNebula already includes an Amazon EC2 (Web, 2010a) virtualizer driver.

Besides, Haizea has been used 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. (Sotomayor et al., 2009b). 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" (Sotomayor et al., 2008).

Haizea supports three types of leases (Sotomayor et al., 2009b), being the first one the focus of interest of this paper:

1. *Advance reservation lease*, where the resource must be available at a specific time;
2. *Best-effort leases*, where resources are provisioned as soon as possible;
3. *Immediate leases*, where resources are provisioned when requested or not at all.

3 FLEXIBLE ADVANCE RESERVATION (FAR) LEASES

The main contribution of this paper consists of 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 on the allocated VMs could get overloaded. In order to try to overcome this, a solution consists on adding 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 clients want to get a certain level of Quality of Service (QoS), i.e, a bounded response delay for their application. They also want to save some money and not to waste part of their booked resources. So, with FAR leases, they could purchase enough VMs for an average load and also a flexible range for a peak load. This possibility would be cheaper than purchasing the full range of VMs for the whole leasing period, even though they will be only needed for coping with a unpredictable small peak-load period.

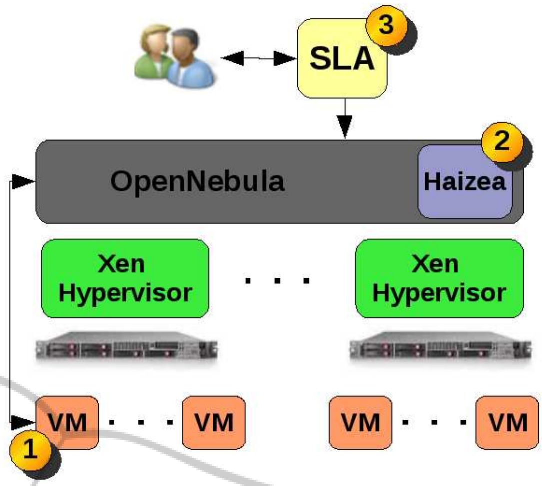


Figure 2: Steps for supporting flexible advance-reservation leases.

As a counterpart, when using FAR leases, the Cloud provider could increase the number of Cloud clients (thus also increasing benefits) since the "flexible" VMs do not be to be allocated the whole reservation time to one specific client. Instead, the scheduler within the Cloud provider could implement strategies to estimate when and how many more flexible VMs need to be allocated over time, and thus decide whether more clients should be admitted or not.

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

1. A "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. Its mission is to monitor the QoS offered by the service running in that VM, triggering the creation/deletion of VMs within the specified flexible range as needed.
2. The Haizea Metascheduler has to be extended to consider the new type of lease. As stated earlier, Haizea has three types of leases: *advance-reservation* leases, *best-effort* leases, and *immediate* leases. This proposal is related to the first one, extending its characteristics to include the "flexibility" feature: a range of virtual machines. Moreover, Haizea would need some control over this flexibility feature in order to know when the sys-

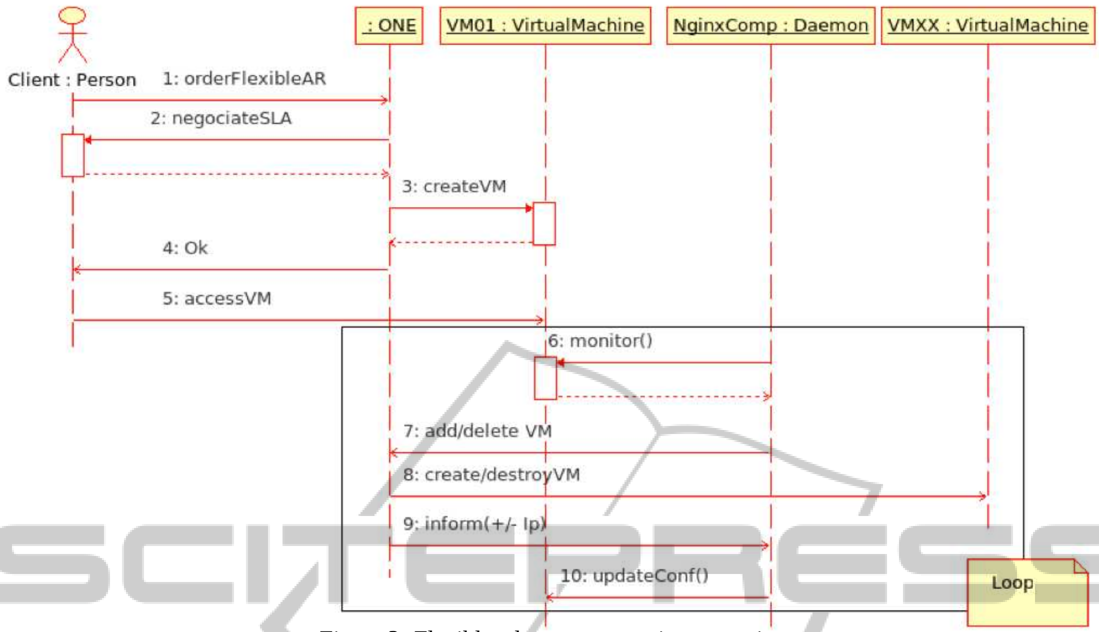


Figure 3: Flexible advance-reservation scenario.

tem is empty of resources to provide, or if there are still enough resources to fulfill a client request. Here, a range of heuristics can be applied.

3. Support for a 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. Cloud computing providers would need new ways to generate SLA templates which include the new flexible advance-reservation lease.

Figure 3 shows an activity diagram which models the intended behavior of a FAR lease. First of all, a client asks OpenNebula for a (number of) virtual machine(s) using an Flexible Advance-Reservation lease. Both parties negotiate an SLA, and ONE allocates the VM to the client. The client sets up its service on the newly allocated VM, and monitoring on the performance of such service starts. After that, and depending on the action to do, if the service becomes collapsed, the smart component would ask for a new virtual machine to help supporting the service. On the other hand, if the service is working better than it was previewed, the component would tell OpenNebula to delete one existing virtual machine. This procedure works as a loop within the booked period of time. Moreover, the smart component has the ability to automate the dynamic provisioning of VMs taking into account the negotiated SLA.

Of course, this basic scenario could be improved by implementing some criteria (i.e., prediction techniques) to anticipate the service collapse, and have the additional VM(s) ready before collapse really occurs. Also, the time needed to set up additional VMs should be considered, when fine-tuning the heuristics inside the smart component.

4 PRELIMINARY RESULTS

In order to have an insight into the foreseen benefits of our proposal, a proof-of-concept experiment has been devised which shows the system behavior when VMs are automatically provisioned and freed on workload fluctuations.

The physical infrastructure used in this work consists of one central host in which OpenNebula is installed, and two more hosts where VMs will be deployed. In both hosts the Xen Hypervisor (Web, 2010c) is used to manage VMs. As Figure 4 depicts, 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.

It is supposed that there is a client who has booked one virtual machine for a period of time. This VM has an Nginx web server (ngi, 2010) with PHP installed, and also some PHP scripts which consume CPU time.

For testing that virtual machine, a variable workload has been designed. Taking into account that the arrival rate is one request per second, three types of requests have been created: *Low load requests*, which

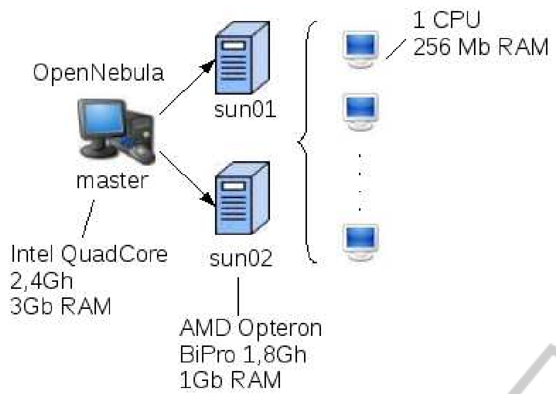


Figure 4: Infrastructure used in the experiment.

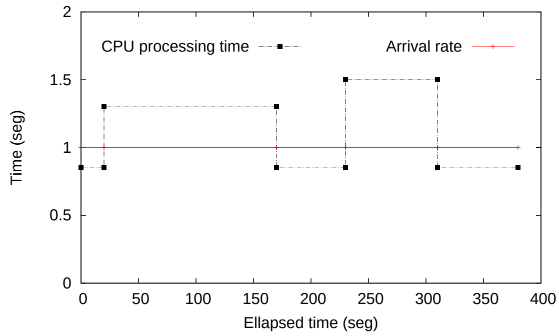


Figure 5: Workload profile.

take less than a second to finish; *High load requests*, which take more than a second to finish; *Very high load requests*, which take much more than one second to finish.

The workload submitted to the web server has been created by mixing the three types of requests within a period of time. As Figure5 shows, the workload starts with a period of low load requests, being followed by a period of high load requests. This sequence of requests will load the web server. After that, it comes another easy low load period before introducing the highest load requests into the web server. The test period ends with another low request period. The rationale behind this workload variations is to mimic a workload that could need more resources than the initially foreseen.

The response time of the jobs is the metric used in our proof-of-concept evaluation. This parameter is a measure of the QoS provided by the web server. More parameters will be considered in a future such as CPU usage or energy consumption, which could be also used to set up trade-offs between users' service guarantees and provider's interests.

Next, some plots which illustrate the evolution of the web server performance without and with FAR leases are shown. Figure 6a shows the web server performance under the specified workload without FAR leases. It can be observed that the web server gets overloaded during the period of high load requests, because lots of them are being queued. When introducing low load requests, the web server is most of the time overloaded due to previously queued requests. Then, while introducing very high load and then again low load requests, the web server shows the same performance as before.

When FAR leases are applied, performance degradation is avoided just by booking a range of virtual machines in the same period of time. As we can see in Figure 6b when the main virtual machine gets overloaded, the smart 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 web server is working fine and it decides to delete the extra VM. When the very high load part of the workload appears, the component asks for one extra VM and it also immediately asks for another one for preventing the web server to get overloaded. When this part finishes and the second low part arrives, the web server works fine again and the component deletes those extra virtual machines.

5 CONCLUSIONS AND FUTURE WORK

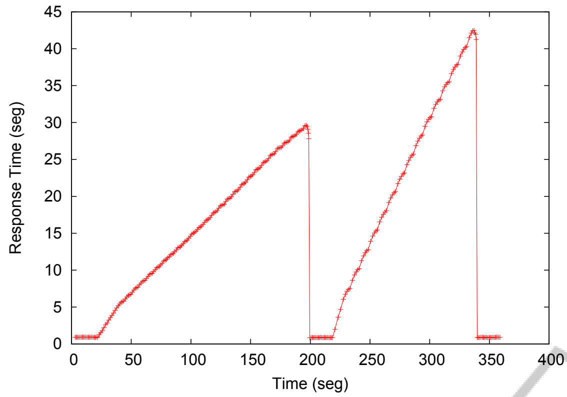
In this work, a new type of Virtual Machine lease, named Flexible Advance Reservations (FAR) lease, is proposed in order to provide Cloud clients with a cost-effective way to cope with periods of peak load, while allowing the Cloud provider to make a better utilization of resources, thus increasing benefits.

The general framework for supporting FAR leases has also been described, which includes three parts: 1) a smart-component implemented within the provisioned VM, 2) extensions to the Haizea scheduler, and 3) support from the SLA manager.

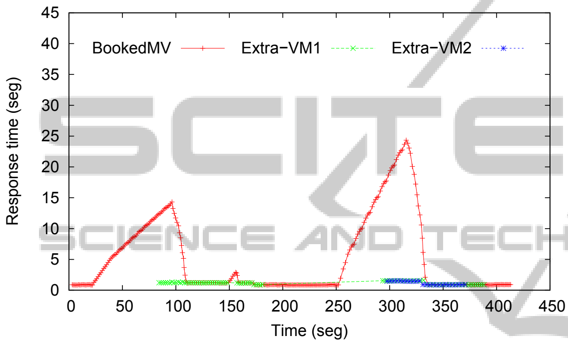
The smart-component has already been created. Its aim is to monitor the QoS offered by the services provided by the booked VMs, and decide whether more VMs are needed, or the system works fine.

Finally, a proof-of-concept evaluation has been included, showing the benefits of the automatic and dynamic provision of resources on applications demand.

Up to now, the current implementation is an initial prototype focused specifically on the Nginx web server. As future work, more intensive testing, over larger environments, need to be carried out. Also, it



(a) Without FAR lease.



(b) Using FAR lease.

Figure 6: System performance.

is intended to develop the full framework described here. The first step will be extending the Haizea scheduler with the new type of lease. Finally, research will be carried out on algorithms for improving the smart component performance, and on strategies for VM scheduling in a Cloud supporting FAR leases.

REFERENCES

(2010a). Amazon Elastic Compute Cloud (EC2), URL: http://aws.amazon.com/ec2/.

(2010b). Elastic Hosts - Flexible servers in the cloud, URL: http://www.elastichosts.com/.

(2010). Nginx web server, URL: http://nginx.org.

(2010c). Xen Hypervisor, URL: http://xen.org/.

Barham, P. et al. (2003). Xen and the Art of Virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*.

Dias de Assuncao, M., di Costanzo, A., and Buyya, R. (2009). Evaluating the cost-benefit of using Cloud computing to extend the capacity of clusters. In *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*.

Keahey, K. and Freeman, T. (2008). Science Clouds: Early Experiences in Cloud Computing for Scientific Applications. In *Proceedings of the Workshop on Cloud Computing and Its Applications*.

Llorente, I., Moreno-Vozmediano, R., and Montero, R. (2009). Cloud Computing for on-Demand Grid Resource Provisioning. *Advances in Parallel Computing, Volume 18 (2009): High Speed and Large Scale Scientific Computing. IOS Press.*

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

Nurmi, D., Wolski, R., and Grzegoczyk, C. (2008). The Eucalyptus Open-source Cloud-computing System. In *Proceedings of the Workshop on Cloud Computing and Its Applications*.

Sotomayor, B., Keahey, K., and Foster, I. (2008). Combining Batch Execution and Leasing Using Virtual Machines. In *Proc. of the 17th Intl. Symp. on High-Performance Distributed Computing (HPDC).*

Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I. (2009a). Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 13(5):14–22.

Sotomayor, B., Santiago, R., Martn, I., and Foster, I. (2009b). Resource Leasing and the Art of Suspending Virtual Machines. In *Proceedings of the 11th IEEE Intl. Conf. on High Performance Computing and Communications*.