A Comparison of Private Cloud Systems

Cristian Chilipirea, Ghita Laurentiu, Mirona Popescu, Sorin Radoveanu, Vladimir Cernov, Ciprian Dobre
Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Bucharest, Romania
{ghita.laurentiu; mirona.popescu; sorin.radoveanu; vladirmir.cernov}@cti.pub.ro; {cristian.chilipirea; ciprian.dobre}@cs.pub.ro

***Abstract* — Cloud Computing represents one of the most popular new paradigms making utility computing a reality. In recent years, this concept has received a lot of attention because of the enormous benefits it brings: for instance the capability known as elasticity permits users to change the number of machines they use according to their needs and in a timely manner. Many large companies deployed public cloud systems such as Amazon EC2 or Microsoft Azure. Even though the popularity and usefulness of cloud systems is clear, there are still concerns around the use of public clouds: availability, data safety or privacy. Because of these concerns there is a need to deploy private cloud systems. Systems that run on machines owned by the organization that requires them, offering similar services as public clouds to different parts of the organization while maintaining a large degree of trust.**

With the need for private cloud systems a number of software solutions appeared. We concentrate here on Infrastructure as a Service, cloud systems where users are given full access to virtual machines. These are generally Open Source and capture large interest from the public. In this paper we make a through comparison between three of the most popular open source cloud software systems and describe our experience with installing and maintaining small deployments using these three software suits.

Keywords — cloud computing; performance; openstack; opencloud; opennebula

I. INTRODUCTION

Cloud system [1] use has grown dramatically in recent times. There are more and more companies deploying public and private clouds. With the rise in cloud usage multiple type of clouds have appeared: Metal as a Service - Where users are given access to physical machines; Infrastructure as a Service – Where users are given access to virtual machines that run on a infrastructure maintained by the cloud owner (Here we have systems such as Amazon EC2 or Microsoft Azure); Platform as a Service – Users are getting access to execution runtime, databases or web servers; and finally Software as a Service - where users get access to an application such as games, e-mail, virtual desktops or online document editors.

We are specifically interested in Infrastructure as a Service (IaaS) systems. These systems benefit on overprovisioning to offer low costs to virtual machines used by many different users. When we compare clouds with previous systems such as cluster or grids, one important advantage is elasticity. Elasticity permits cloud users to increase and lower the number of machines they use, and with them the cost, depending on their needs. A high, abrupt increase in the number of clients means a need to deploy more machines in a time sensitive manner. In the classical model, where all machines were owned by the company offering said service, this would only be possible if other physical machines were available. This was generally not the case, because this would mean an extremely high increase in cost while the benefits would be only temporary. For instance a shop would have an increase in clients using their services during the winter holiday period far higher than the rest of the year. With elasticity, more machines can be acquired during the holiday season and discarded after, for little cost.

Clouds open the way for small groups to take advantage of large computing power in order to enable their applications. The need for large processing power is felt even more now, with the rise of Big Data. These systems require more and more processing power to run complex machine learning algorithms that extract information of the data set.

Trust is the main problem of public clouds. A user of a cloud may need assurances about availability, data privacy and data security, especially with sensitive data sets. These assurances cannot be guaranteed. Not only there has to be trust in the public Cloud provider, but also the government and the regulations of the country the cloud provider operates in. This is made especially clear by the NSA issues in the USA.

Private clouds, and more recently hybrid clouds, are a possible solution. Companies that own physical hardware have different uses and needs for it depending on internal groups and departments. Private clouds are a way for multiple groups or departments to share the same physical resources and scale up and down individual systems as needed. For instance during holiday season, the number of machines serving the online shop would grow, while the number of machines used for internal accounting could be lowered. However number of physical machines is still limited. The solution is given by hybrid clouds, which take advantage of both options, local trusted resources, for sensitive applications, and the promise of virtually infinite resources that could be offered by public clouds.

Next we present three software solutions for private and, more recently, hybrid cloud systems and describe our experiences with installing and managing these systems. The solutions we present are: OpenStack [2], OpenNebula [3] and CloudStack [4] - Section III. These are all popular, open source, private cloud management software solutions. In Section IV we present experimental results and a direct comparison between these solutions. Finally we finish with our conclusions and recommendations as well as suggestions for future work – Section V.

II. Related Work

With the recent interest and demand of Cloud systems there are numerous works and projects that try to analyze the capabilities of clouds. For instance [5] makes a detailed analysis on the capabilities of Amazon EC2 to run scientific computing software. These tests are also run on private cloud systems, like the Illinois University Cloud [6]. Other works concentrate on performance and power management [7].

Most research in cloud system is concentrated on improvements in performance from the perspective of the cloud owner. These improvements are given by better ways of transferring data [8], selecting [9] or allocating [10] services.

However not all works clarify what they are testing. For instance, in [11] the authors test the Hadoop map reduce engine on the Eucalyptus [12] cloud system. This kind of test offers no information about how the management system, Eucalyptus, works but shows how the physical infrastructure and the virtualization system work. This type of measurements make more sense when one is trying to determine the performance of the virtualization system, such as in [13] where an in depth analysis and comparison between the popular virtualization systems Xen [14] and KVM [15] is made.

The problem of measuring the virtualization system instead of the cloud management system is present in many works that claim to compare different private cloud software systems. For instance, in [16] the authors claim to compare Eucalyptus and OpenStack but use a lot of benchmarks that measure the performance of the virtual machines themselves. In [17] claim to make a performance evaluation of Eucalyptus and CloudStack but put together metrics that compare, very thoroughly, the behavior of the virtualization systems with metrics such as start time which actually give information on the Cloud software. The other metrics measure things such as CPU/RAM/Network for the isolation of different virtual machines.

It is vital to differentiate between the metrics that indicate the performance of the different components of a cloud framework. This is especially important considering these components can be interchangeable. For instance, most cloud system support both Xen and KVM.

Works that make comparisons of cloud systems concentrate on a variety of things, for instance [18] studies the problem of security in clouds and [19] concentrates on security issues of open source, private clouds.

Other works only compare the features of various cloud software systems like [20] which compares multiple IaaS solutions and [21] which compares not only IaaS but also PaaS solutions. Similarly [22] tries to identify the best open source cloud software based on the user needs looking only at the features. The features of public and private cloud systems are analyzed in [23]. A more through study conducted in the same manner is presented in [24].

A proposed benchmark for analyzing cloud management software is presented in [25], here the authors test their solution on Eucalyptus and OpenStack. However the author's methodology only takes into account the number of virtual machines being started but it doesn't take into account the size of each machine. A similar solution is presented in [26] where the solution is used to supplement a comparison between OpenNebula and OpenStack. The authors of [27] present a similar comparison, but in the context of cosmological simulation use case, they do not investigate varying the size of the virtual machines because they concentrate on the application that runs on them. In the following chapters we show how the size of the machine can affect the start time.

The work of [28] tries to determine the effects of different configurations when deploying virtual machines. They show how different RAM or CPU settings don't effect the results very much, but the size of the virtual hard drive does.

We are not aware of any work that compares OpenStack, CloudStack and OpenNebula in a comprehensive matter, looking both at the future and the elasticity performance of these three private cloud systems.

III. Cloud Management Software

Cloud management software are complex stacks that deal with everything from provisioning to monitoring inside a cloud environment. Because they are so complex, they are usually built using interchangeable modules. Common modules, found in most solutions, are: a GUI dashboard, a way to control the infrastructure; networking, controlling things such as firewalls and connections inside the cloud and allocation of IP addresses; identity service, to control who has access to what parts of the cloud; monitoring services; storage services, that are used to store the virtual machines and the different flavors of virtual machines the cloud can use; and most importantly a hypervisor (a physical node that runs a hypervisor is also known as a compute node), the module that is in charge of running a virtual machine on each individual physical host.

OpenStack is a free open source cloud management software stack. It started as a joined NASA and Rackspace project and now it is supported by a large number of corporations. Installing OpenStack showed us that it is the software which is most complex, with most features. This means all the details of how the cloud is organized can be controlled, but it takes a rather steep learning curve.

OpenNebula is also an open source project. It is mostly popular in Europe where many different research projects and grants are invested in its development. It is not as full of features as OpenStack but it is very close. During our experiments we have found no feature that we needed and was not available. The software itself was probably the simplest to install and manage and it offered enough variety to please most cloud administrators.

Finally, Apache Foundation answer to a Cloud Management engine, **CloudStack**. This solution claims to be highly available and highly scalable. It supports multiple management server nodes and experimental results show that it can theoretically manage 30.000 machines with only 4 management server nodes. This solution has intensive CPU and RAM usage.

To run our experiments, we used three identical physical machines with the following configuration: Intel Core 2 Quad CPU Q9400 at 2.66 GHz with virtualization enabled and 4GB of RAM.

IV. Comparison of Cloud Systems

In this chapter we compare three of the most popular cloud management software solutions: OpenStack, CloudStack and OpenNebula. Each of them are widely used for the management of private clouds and are slowly extending with the appearance of hybrid clouds.

Because the fact that these are open-source systems, the more popular they are and the more attention they receive the more programmers and engineers contribute to the code base. With more people contributing, more features are available and security issues are solved faster. To understand the popularity of these systems we looked at google trends, a method of finding out how many people used a search term in a certain period.

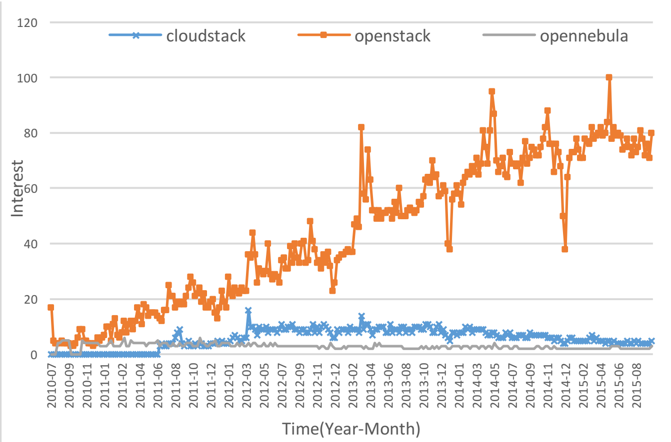

Fig. 1 Popularity of Cloud Management Systems over time

The results shown in Fig. *1* shows how, based on google trends, OpenStack is, by far, the most popular cloud software system. In 2010 the idea of private clouds appeared and in the next years it became popular. OpenStack was also the first to be launched, the last of the three being CloudStack.

Furthermore OpenStack is popular worldwide, with a large interest in Asia, USA and Europe. In contrast CloudStack is mostly popular in Asia and USA while OpenNebula is most popular in Germany, followed by USA. The popularity of OpenNebula in Europe, especially Germany is also motivated by the large number of research grants that promote the extension of OpenNebula. On the other hand OpenStack was started by NASA and Rackspace, and is now supported by a very large number of companies. Finally CloudStack is an Apache Software Foundation project.

As we showed in the Related Work section, a lot of papers that claim to compare cloud management systems measure instead just the capabilities of their respective hypervisors. With a lot of interest in these technologies the components each of them supports has continued to grow.

Table *1* shows the supported hypervisors for each of the three systems. It is clear that all of them have interchangeable hypervisors and all of them support the most popular ones: Xen, KVM and VmWare. Because of this it is important that any measurement on the performance of the virtual machines run by these systems should take into account the hypervisor being used.

Table 1 Supported Hypervisors

	OpenStack	CloudStack	OpenNebula
Xen	Yes	Yes	Yes
KVM	Yes	Yes	Yes
LXC	Yes	No	No
VmWare	Yes	Yes	Yes
Hyper-V	Yes	Yes	No
Oracle	No	Yes	No
BareMetal	Yes	Yes	No

OpenNebula is the solution that has the fewest options for choosing a hypervisor. This is because the way OpenNebula developers concentrate on free and open source software.

We make note that bare metal is not technically a hypervisor. Bare metal represents installing machines, directly on physical hardware, without any virtualization. This technique is called Metal as a Service, because using this method gives the client access to the actual physical machine, the metal. Metal as a Service usually requires more complex hardware that can provide remote power management for the machines. This technique could be extremely beneficial for users that need maximum efficiency or that need features that aren't perfectly virtualized. The use of Metal as a Service could also translate into an increase in security as there is no worry for a different machine, of a different client, running on the same system.

Table 2 Supported Configuration Methods

	OpenStack	CloudStack	OpenNebula
GUI	Yes	Yes	Yes
SDK	Yes	No	No
CLI	Yes	Yes	Yes
HTTP API	Yes	Yes	Yes

All the cloud systems presented in this paper offer a variety of ways to configure them. This is shown in Table *2*. OpenStack is the one with the highest variety, supporting a large, and increasing, number of languages from which configurations can be made. Offering a variety of options to configure the cloud is important because it raises the ease to integrate it into an existing system or to implement new applications for it. These applications need to use APIs or the CLI to be able to better take advantage of the elasticity feature. However a large variety is not always the best option, because of the need for extensive documentation and the possible incompatibility in features offered by different options.

To evaluate the efficiency of these systems we look at elasticity. And in order to measure elasticity we measure the start time of virtual machines. This indicate how fast a system can take advantage of elasticity to respond to an increase in the number of clients.

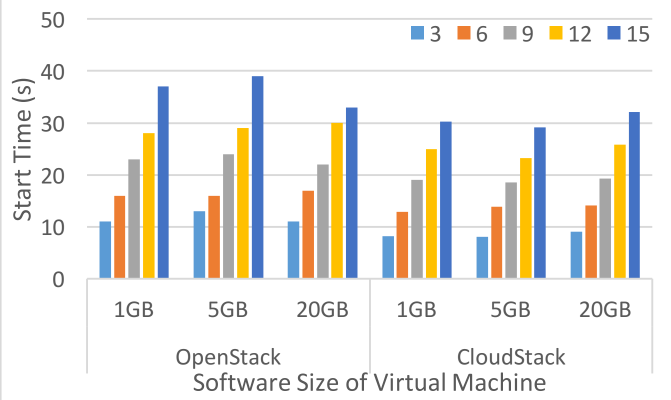

Fig. 2 Start Time depending on number of VM and their size

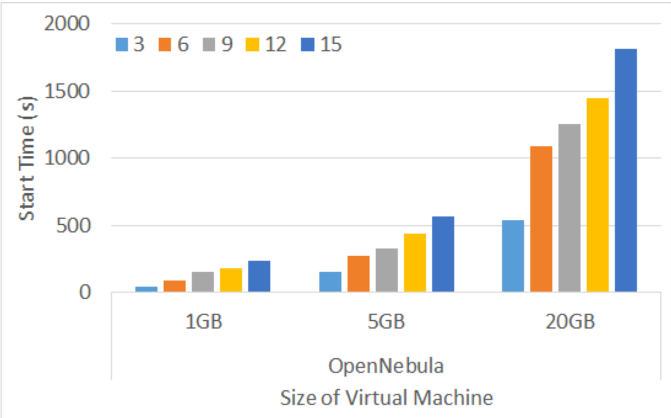

Fig. 3 Start Time depending on number of VM and their size

We varied both the number of virtual machines run simultaneously and the size of these machines and measured the Start Time. We define the start time as the time period beginning at the moment the command to start them was given, to the moment when the last machine was started. The virtual machines all run the same software, have allocated one virtual CPU core each and 128MB of RAM, the only thing that varies is the virtual hard drive space they have allocated. There is no information in the drive itself, other than the installed operated system. In all cases this was Cirros-0.3.3-x86_64.

In these experiments we only measured the start time, as this is the most indicative of elasticity. In other works we have identified experiments that also measure the termination time for a machine, but this is relevant only for the cost, not the efficiency of elasticity. In other words, it may cost more for the client if a machine takes a lot of time to shut down, but as soon as the termination process starts, the machine is no longer useable.

As we can see in *Fig. 2* and in Fig. *3* OpenStack and CloudStack are not affected by the size of the allocated hard drive space. OpenStack was configured to deploy all machines simultaneously. This is not the default method, but it was a simple configuration. However this size has dramatic effects for the OpenNebula system. We plotted the results for OpenNebula in a separate graph because they vary so much from the results of the other two systems. As the results show, Cloud Stack is probably the most efficient when the goal is to minimize the start time of virtual machines and to achieve high elasticity.

V. Conclusion

We showed how three of the most popular open source cloud management solutions compare with each other in terms of features, popularity and elasticity.

Considering our analysis we consider OpenStack to be the solution that has most features and is the most popular. However the large number of features translate in a difficult to use and install. Given our experience we appreciate the simplicity in dealing with OpenNebula and CloudStack.

As future work we envision extending this comparison to other cloud management solutions as well as introducing a larger number of measures with which to benchmark the cloud management software, and not their individual components such as the interchangeable hypervisors they use. For instance it would be interesting to test the scalability of each solution.

Acknowledgment

The research presented in this paper is supported by projects: *MobiWay*, Mobility beyond Individualism: an Integrated Platform for Intelligent Transportation Systems of Tomorrow - PN-II-PT-PCCA-2013-4-0321; *DataWay*, Real-time Data Processing Platform for Smart Cities: Making sense of Big Data - PN-II-RU-TE-2014-4-2731. We would like to thank the reviewers for their time and expertise, constructive comments and valuable insight.

References

[1] A. Michael, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," *Communications of the ACM* , vol. 53, no. 4, pp. 50-58, 2010.

[2] J. Kevin, OpenStack cloud computing cookbook, Packt Publishing Ltd, 2012.

[3] M. Dejan, I. M. Llorente and R. S. Montero., "Opennebula: A cloud management tool," *IEEE Internet Computing,* vol. 2, pp. 11-14, 2011.

[4] "Apache CloudStack," Apache, 01 10 2015. [Online]. Available: https://cloudstack.apache.org/. [Accessed 01 10 2015].

[5] O. Simon, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," *Cloud computing* , pp. 115-131.

[6] K. Ahmed, A. Al-Nayeem and I. Gupta, "Performance evaluation of the illinois cloud computing testbed," 2009.

[7] H. V. Nguyen, F. D. Tran and J.-M. Menaud, "Performance and power management for cloud infrastructures," in *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 329-336, 2010 .

[8] M.-C. Nita, C. Chilipirea, C. Dobre and F. Pop, "A SLA-based method for big-data transfers with multi-criteria optimization constraints for IaaS," in *11th Roedunet International Conference (RoEduNet)*, Sinaia, 2013.

[9] V. Serbanescu, F. Pop, V. Cristea and O.-M. Achim, "Web Services Allocation Guided by Reputation in Distributed SOA-Based Environments," in *11th International Symposium on Parallel and Distributed Computing (ISPDC)*, Munich/Garching, Bavaria, 2012.

[10] O.-M. Achim, F. Pop and V. Cristea, "Reputation Based Selection for Services in Cloud Environments," in *14th International Conference on Network-Based Information Systems (NBiS)*, Tirana, 2011.

[11] J. J. P. and A. Basu, "Performance Analysis of Hadoop Map Reduce on Eucalyptus Private Cloud," *International Journal of Computer Applications ,* vol. 79, no. 17, pp. 10-13, 2013.

[12] N. Daniel, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," *IEEE/ACM International Symposium on Cluster Computing and the Grid,* pp. 124-131, 2009.

[13] D. Todd, Z. Shepherd, J. Matthews, M. Ben-Yehuda, A. Shah and B. Rao., "Quantitative comparison of Xen and KVM," in *Xen Summit*, Boston, MA, USA , 2008.

[14] B. Paul, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review ,* vol. 37, no. 5, pp. 164-177, 2003.

[15] K. Avi, Y. Kamay, D. Laor, U. Lublin and A. Liguori, "kvm: the Linux virtual machine monitor," in *Linux Symposium*, 2007.

[16] S. S and A. Basu, "Performance of Eucalyptus and OpenStack Clouds on FutureGrid," *International Journal of Computer Applications ,* vol. 80, no. 13 , pp. 31-37, 2013.

[17] M. M. AL-Mukhtar and A. A. A. Mardan, "Performance Evaluation of Private Clouds Eucalyptus versus CloudStack," *International Journal of Advanced Computer Science and Applications,* vol. 5, no. 5, pp. 108-117, 2014.

[18] A. Mohamed, J. Grundy and I. Müller, "An analysis of the cloud computing security problem," in *APSEC Cloud Workshop*, Sydney, Australia, 2010.

[19] R. Sasko and M. Gusev, "Security evaluation of open source clouds," in *IEEE EUROCON*, 2013.

[20] S. Yadav, "Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, Openstack and Opennebula," *International Journal Of Engineering And Science,* vol. 3, no. 10, pp. 51-54, 2013.

[21] G. v. Laszewski, J. Diaz, F. Wang and G. C. Fox, "Comparison of Multiple Cloud Frameworks," in *IEEE International Conference on Cloud Computing (CLOUD)*, 2012.

[22] S. Peter and D. Thain, "A comparison and critique of eucalyptus, opennebula and nimbus," in *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010 .

[23] K. Nawsher, A. Noraziah, E. I. Ismail, M. M. Deris and T. Herawan, "Cloud computing: Analysis of various platforms," *International Journal of E-Entrepreneurship and Innovation ,* vol. 3, no. 2, pp. 51-59, 2012.

[24] V. Ivan, M. Orlić and B. Mihaljević, "An early comparison of commercial and open-source cloud platforms for scientific environments," *Technologies and Applications In Agent and Multi-Agent Systems,* pp. 164-173, 2012.

[25] S. Dylan, B. W. Perrault, R. Nordeen, J. Wilson and X. Wang, "Cloud computing performance benchmarking and virtual machine launch time," in *ACM 13th annual conference on Information technology education*, 2012.

[26] K. Konstantinos, A. Kapsalis, D. Kyriazis, M. Themistocleous and P. R. d. Cunha., "Open-Source Iaas Fit For Purpose: A Comparison Between Opennebula and Openstack," *International Journal of Electronic Business ,* vol. 11, no. 3, pp. 191-201, 2013.

[27] C. Eddy, L. Toch and J. Rouzaud-Cornabas, "Comparison on OpenStack and OpenNebula performance to improve multi-Cloud architecture on cosmological simulation use case," 2013.

[28] P. Aaron, L. Liu and B. Yuan, "Benchmarking the Performance of OpenStack and CloudStack," *IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC),* pp. 405-412, 2014.