International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.6, December 2013, pp. 405~418
ISSN: 2089-3337

Performance Evaluation of the CloudStack Private Cloud

Mumtaz M.Ali AL-Mukhtar *, Asraa Abdulrazak Ali Mardan **
* Departement of Internet Engineering, AL-Nahrain University
** Departement of Networks Engineering, AL-Nahrain University

Article Info

Article history:

Received Oct 15th, 2013
Revised Nov 20th, 2013
Accepted Dec 10th, 2013

Keyword:

Cloud Computing
IaaS
CloudStack
Benchmarks
Performance
Evaluation

ABSTRACT

The number of open source cloud platforms is increasing day-by-day. The features of these platforms vary significantly and this creates a difficulty for the cloud consumers to choose the platform based on their requirements. In this paper we build a private cloud using *Cloudstack***,** a popular open source platform used to build Infrastructure as a service (IaaS) cloud. We present its architecture and analyze performance of virtual machines initiated and managed by the CloudStack in terms of CPU usage, memory bandwidth, disk I/O speed and networking performance using suitable benchmarks. Different virtual machine management operations such as add, delete and live migration are also evaluated. The performance evaluation of CloudStack can help to determine its suability to be adopted as on premise cloud solution.

Corresponding Author:

Asraa Abulrazak Ali Mardan,
Departement of Networks Engineering,
AL-Nahrain University,
P.O.Box 64065, Al-Jadriya, Baghdad, Iraq.
Email: asraaiteng@gmail.com

1. INTRODUCTION

Cloud computing as a new Internet service concept has become popular to provide a variety of services to users. It is a combination of technologies that have been developed over the last several decades, which includes virtualization, dynamic provisioning, internet delivery of services, grid computing, cluster computing and utility computing [1][2]. According to NIST (National Institute of Standard and Technology), “Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [3]. There are three models by which Cloud computing services are delivered. **Software as a Service (SaaS)**: delivers applications to consumers using a multitenant architecture. **Platform as a Service (PaaS)**: delivers development environments to consumers, provide the required programming languages and tools. **Infrastructure as a Service (IaaS):** delivers hardware resources such as the processing, storage, networks, and other fundamental computing resources on which the consumer can rent and then run his operating system and applications. [4].

In addition to the above mentioned delivery models, there are three deployment models for Cloud computing: public, private, and hybrid. **Public Cloud** is a cloud computing infrastructure that is made available as ―pay-as-you-go and accessible to the general public. It is provided by an off-site third-party service provider which own and manage the physical infrastructure. **Private Cloud** refers to a cloud infrastructure that is internal to an organization and is usually not available to the general public. A private cloud’s data centers can be on premise and the physical infrastructure is owned and managed by the organization that owns it. **Hybrid cloud** deployment model is a composition of two or more cloud deployment models that are bound together by standardized or proprietary technology [5].

Due to security and privacy issues of public cloud and lack of service level agreement (SLA), many organizations hesitate to use public cloud in which computing resource are shared with other companies. Companies do not have any knowledge of where their applications are run and their data are stored or how access to them is controlled. Therefore, private cloud is a good alternative especially with popularity of open source cloud computing platforms nowadays. The performance evaluation of these open source solution is important as their number is increasing day-by-day and their features vary significantly which create difficulty for companies to choose one of them based on business and scientific requirements.

The rest of this paper is organized as follows: section 2 present related works. Section 3 introduces the architecture of CloudStack platform. Section 4 covers the performance evaluation of CloudStack VMs. Section 5 describes the test environment and methodology. Section 6 analyzes VMs startup and release time. Section 7 evaluates live migration of VMs. Finally conclusions are drawn in the last section.

2. RELATED WORKS

Many studies have been conducted to evaluate performance of open source cloud platforms such as Eucalyptus, Opennebula and Nimbus. Nevertheless a little work has been done yet to evaluate CloudStack due to the fact that it is a relatively new platform. However these researches haven't performed a complete performance analysis of the cloud platform.

de sousa et al. [1] evaluated eucalyptus VMs considered processing and disk I/O performance only while authors in [6, 7,8] brought out an overview of architecture of open source platforms architectures and presented their general features and Characteristics. Mao and Humphery [9] investigated the performance of VM startup and release time of public clouds. Authors in [10] evaluated performance and studied VM launch time of Eucalyptus and OpenStack, however they gave only a general view of VM performance. Folgar et al [11] evaluated performance of CloudStack primary storage only.

Differently from previous works, this research judges performance of CloudStack covering versatile parameters including performance of cloud platform management considering add, delete and live migration of VMs. Performance of VMs in term of CPU usage, memory bandwidth, disk I/O speed and networking performance is regarded as key point of our evaluation.

3. CLOUDSTACK ARCHITECURE

CloudStack is an open source software platform that pools computing resources to build public, private, and hybrid IaaS clouds. It manages the network, storage, and compute nodes that make up a cloud infrastructure. It is designed to deploy and manage large networks of virtual machines, as a highly available, scalable Cloud Computing platform. CloudStack currently supports the most popular hypervisors including VMware, Oracle VM, KVM, XenServer and Xen. CloudStack offers three ways to manage Cloud Computing environments: an easy-to-use web interface, command line and a full-featured RESTful API [12].

The general architecture of CloudStack is shown in figure 1 while the architecture of its availability zone is shown in figure 2. The architecture consists of the following components [13]:

3.1. Management Server

It manages cloud resources and provides the web user interface for the administrator to configure and manage cloud infrastructure and a reference user interface for end users. It controls allocation of virtual machines, manages the allocation of storage to guests as virtual disks, manages snapshots, templates, and ISO images; possibly replicating them across data centers and assigns IP and MAC addresses to the virtual machine instances. The Management Server can be deployed in a single node or multi-nodes for load balancing and high availability purposes.

3.2. Availability Zone

The Management Server manages one or more zones containing host computers where guest virtual machines will run. The cloud infrastructure is organized as follows:

- Zone: Typically, a zone is equivalent to a single datacenter. A zone consists of one or more pods and secondary storage.
- Pod: A pod is usually one rack of hardware that includes a layer-2 switch and one or more clusters.
- Cluster: A cluster consists of one or more hosts with a common type of Hypervisor and primary storage.

- Host: A single compute node within a cluster. The hosts are where the actual cloud services run in the form of guest virtual machines.
- Primary storage is associated with a cluster, and it stores the disk volumes for all the VMs running on hosts in that cluster.
- Secondary storage is associated with a zone, and it stores templates, ISO images, and disk volume snapshots.

This architecture of Cloudstack provides flexibility in scaling the cloud horizontally by adding more than one zone, pod and cluster. These zones are placed at different geographical locations and connected together by the management server. CloudStack also provides High Availability & Disaster recovery solutions that are available in the commercial cloud software.

CloudStack uses several types of system virtual machines to perform management tasks in the cloud; these VMs are per zone and include:

- CPVM (Console Proxy VM): presents a console view of VM via the web user interface. It connects the user's browser to the VNC port made available via the hypervisor for the console of the user VM.
- SSVM (Secondary Storage VM): provides a background task that takes care of a variety of secondary storage activities as downloading a new template to a Zone, copying templates between Zones, and snapshot backups.
- VRouter (Virtual Router): provides NaaS (network as a service) module. It serves VMs of zone and provides rich network functions like DHCP and DNS.

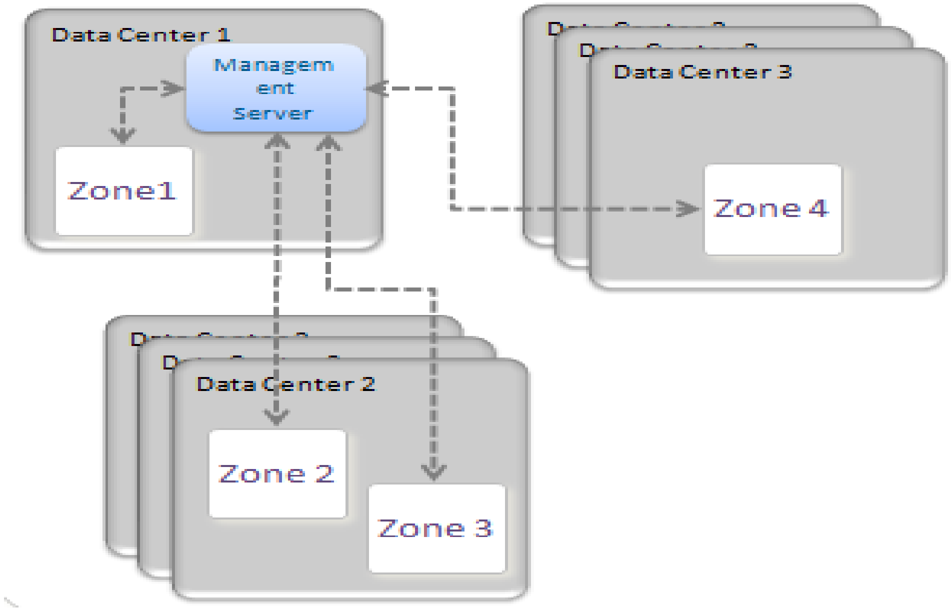

Figure 1. CloudStack Architecture [14]

CloudStack has a beautiful well designed web interface. Users access cloud services via web interface, located at an IP address. The prospective user requests an account that has to be given permission by the cloud administrator to grant the access. When a user requests a VM instance, the provisioning process include following steps:

1. The user logs in and selects the desired availability zone for their instance, and then selects the desired template.
2. Optional Network Services is provisioned according to user request.
3. Template is copied from the secondary storage of the zone to the primary storage of the cluster.
4. If the instance requires any data volumes, the data volumes are created on primary storage.
5. CloudStack then instructs the host to create and start the instance VM.

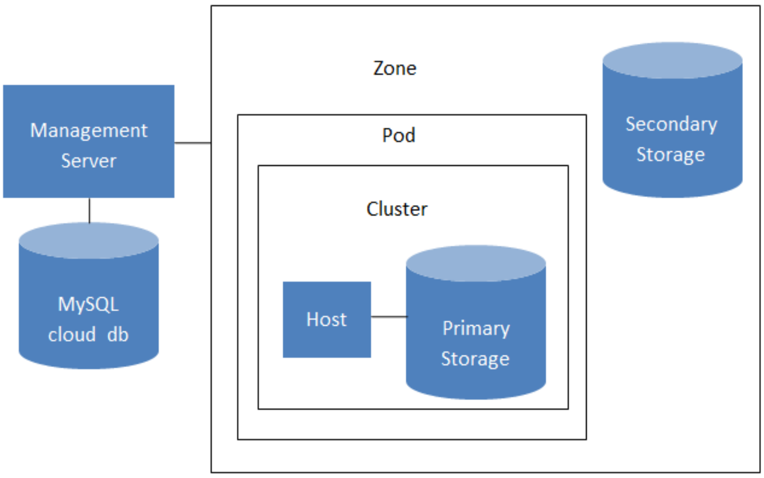

Figure 2. Availability Zone Nested Architecture [13]

4. TEST ENVIRONMEMT AND METHODOLOGY

CloudStack 4.1 with one zone, pod and cluster has been deployed using 3 identical physical servers. One server is used as a management server including primary and secondary storage and the other two servers are used as host machines. The servers are Intel[R] Core[TM] i5-2410M CPU 2.3GHz, 4GB RAM, 500GB SATA Hard Disk (except the management node where disk space is 1Terabyte) and 100MB Ethernet interface. Centos 6.3(final) is installed on each server as native OS. CloudStack with NFS configuration is deployed in servers and each host is configured with kernel-based virtual machines (KVM) as a hypervisor. Basic zone network configuration is used. The servers are connected together using 1GB Ethernet switch.

In order to estimate VMs performance, we have employed a number of benchmarks each for different evaluation purpose. Table 1 shows the selected benchmarking tools for testing CPU, storage, memory and network performance.

A customized CloudStack template has been created (image used to establish VM) in which all benchmarks are installed and configured to save time and ease of work. Each test is repeated five times consecutively and the average is taken into account. Different numbers and types of VM are considered in our performance evaluation. Table 2 shows types of VMs that are provided by our cloud.

Table 1. Benchmarks Deployed for VM Testing

Benchmarks	Testing Resource
Linpack	Processors
Bonnie++	Disk I/O
Stream	Memory
Iperf	Network

Table 2. VM Types

Type	RAM	CPU Core	Disk size
Small	512M	1	10G
Medium	1G	1	10G
Large	2G	2	20G
XLarge	4G	4	40G

5. PERFORMANCE EVALUATION OF VMs

VMs have been evaluated using selected benchmarks considering different resources.

a. CPU Performance

LINPACK [14] is a benchmark that measures a computer's floating-point rate of execution by solving a dense *n* by *n* system of linear equations in double precision. It allows defining the size of the system of linear equations in order to evaluate the performance of computing power. Gflop/s is a rate of execution; it refers to billions of floating point operations per second.

In this test two scenarios have been applied. First two types of VMs (small and large) are evaluated as VM computing power varies according to its type. The number of linear equations is set to n = 7000 in small VM and n= 10000 in large one. In the Second scenario, performance of VM is evaluated when there are different numbers of VMs under a high CPU load to test CPU isolation of VMs and check if there is any interference among them because of resource sharing. In this scenario a medium type VM with n=7000 has been used.

Figure 3 depicts the processor performance of VMs types. The floating point execution rate and time required to solve system of linear equation is considered very good with 5.5 Gflop/s and 13.8 Gflop/s for small and large VMs respectively as compared to values of performance of physical machines with similar hardware specifications [16]. This is due to fact that CloudStack VMs exert better utilization of resources. Figure 4 illustrates performance of VMs when there are other VMs running with 90% CPU utilization. Lookbusy [17] has been used to generate a high CPU load. The figure reveals that CloudStack provides a satisfactory CPU isolation as there is almost no affects from other VMs on the tested VM that run Linpack. Floating-point rate and time of execution are nearly the same as number of VMs with high utilization are increasing in each case.

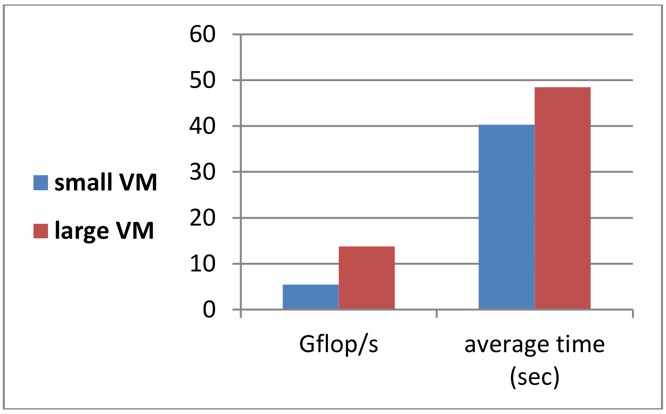

Figure 3. CPU Performance of VMs

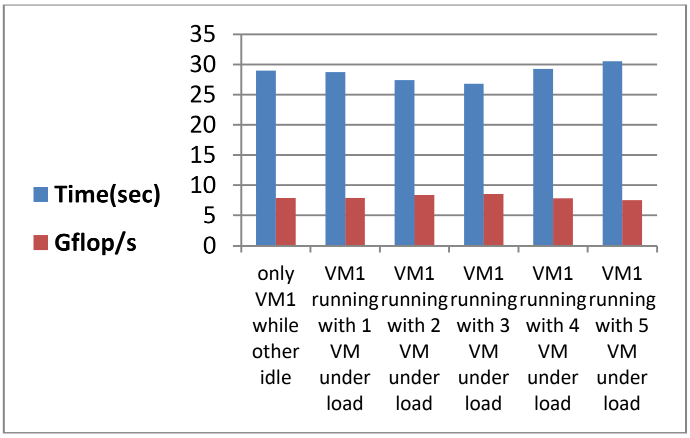

Figure 4. CPU Isolation

b. Disk 1/O Performance

Bonnie++ [18] is a well-known Disk I/O performance benchmark suite that uses a series of tests to reveal how file systems perform various tasks, how file systems are created, or how network file systems perform. Its tests include, data read and write speeds, maximum number of seeks per second, maximum number of file creation, deletion or gathering of file information per second.

Two scenarios are considered in this concern. First, Disk I/O of two types of VMs, small and large is evaluated. As recommend by bonnie++ documentation that file size should be double RAM size, files with 1GB and 4GB sizes for small and large VM are considered respectively. Second scenario, performance of the VM when there is another VM performing intensive disk I/O operation is inspected. This is carried out to test isolation between VMs and investigate if there is any interference due to CloudStak NFS configuration in which VMs share the primary storage disk. In this scenario medium type VM with 1GB file size is dealt with.

Figure 5 shows Disk I/O performance of VM types. Sequential Output shows the speed in KB/s in which the data has been written. Sequential Input is the speed the data has been read. Sequential and Random create is the number of files created per sec.

Figure 6 views performance when two VMs are carrying intensive read write file operations concurrently. It reveals that disk I/O performance of VM disk is affected by the other VM as its performance drops a little. This is expected in NFS configuring due to disk sharing and available network bandwidth of VM.

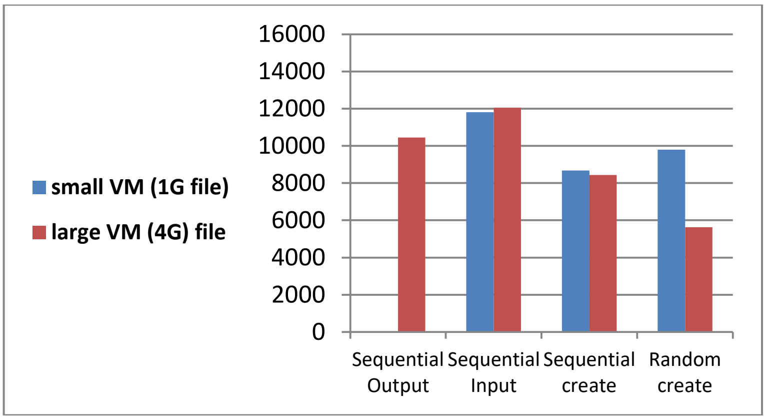

Figure 5. Disk Access Speed

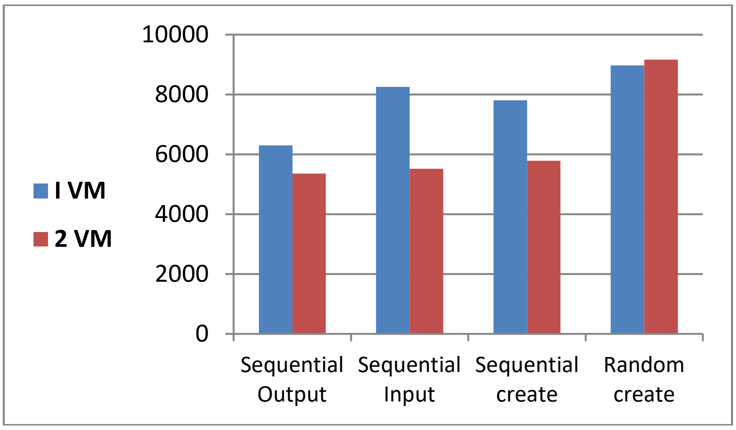

Figure 6. Disk Isolation

c. Memory Performance

The STREAM benchmark [19] is a simple synthetic benchmark tool that measures memory bandwidth (in MB/s). It is specifically designed to work with datasets much larger than the available cache on any given system, so that the results are more indicative of the performance of very large, vector style applications. The benchmark attempts to determine a sustainable “realistic” memory bandwidth, which is unlikely to be the same as the theoretical peak, using four vector-based operations: COPY a=b, SCALE a=q*b, SUM a=b+c and TRIAD a=b+q*c.

Figure 7 indicates the results of memory performance of small and large VMs in MB/s. The array size applied in the benchmarking is 10000000 elements for small VM and 70000000 elements for large VM. Figure 8 demonstrates that with only one VM provisioned, there is plenty of room for further utilization but as the number of VMs increases the bandwidth available to each drops. Hence it requires configuring a scheduler to avoid such effects.

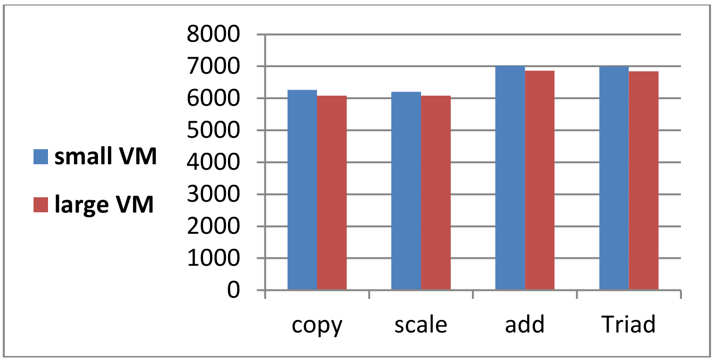

Figure 7. Memory Bandwidth

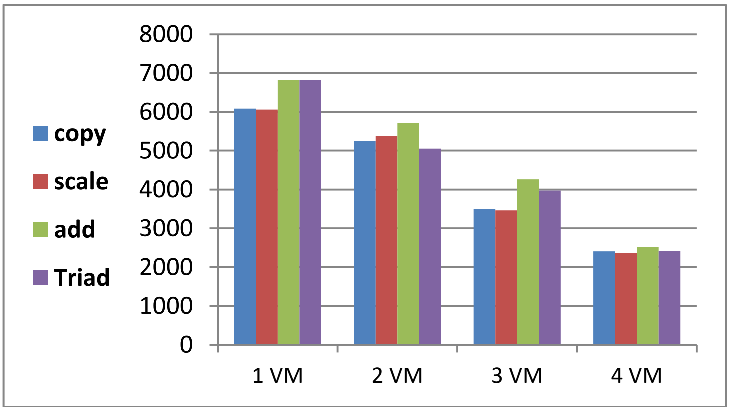

Figure 8. Memory Isolation

d. Network Performance

Iperf [20] is a network testing tool that allows the user to set various parameters that can be used for testing a network, or alternatively for optimizing or tuning a network. It applies a client/server scheme to measure network performance between two ends, by creating a TCP and UDP data streams and measuring the throughput of network that is carrying them.

To measure network functioning, three scenarios have been employed. First, bandwidth of VMs inside the cloud is measured by running two VMs one as client and other as a server and TCP bandwidth between them is measured. Thereafter, the test is repeated when there are others VMs using the network. Second, packet loss is calculated at different bandwidths using UDP mode with a different number of VMs. Third, jitter is determined using UDP mode when there is more than one VM using the network.

Figure 9 proves that when one VM is communicating, it utilizes all available network bandwidth but when there are others VM using the network, the bandwidth is fairly divided among them. As depicted in figure 10, the packet loss is persisting around zero when each VM is communicating at a small bandwidth but as the bandwidth increases the packet loss increases considerably. However it doesn't arrive to a critical loss value.

Figure 11 expresses jitter when the VM is utilizing 100Mbit/s bandwidth. As the number of VMs concurrently using network increases, the jitter value continues nearly the same. This is due to that bandwidth is fairly divided among VMs.

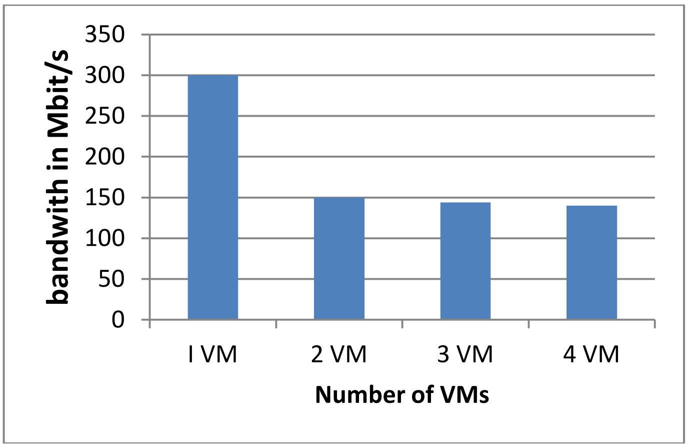

Figure 9. Network Performance Inside Cloud

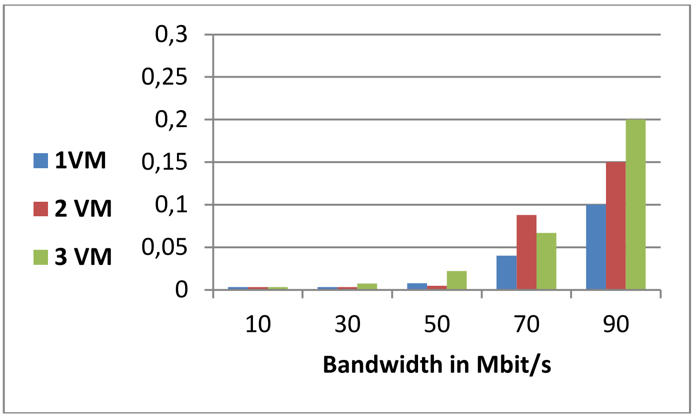

Figure 10. Packets Loss

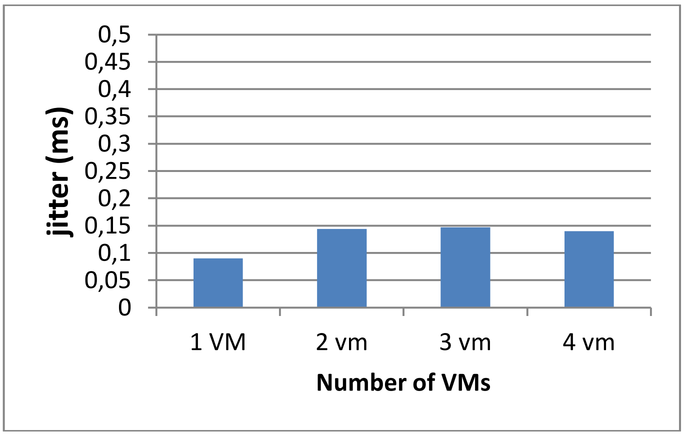

Figure 11**.** Jitter

6. VM PROVISIONING AND REALEASE TIME

One of many advantages of the cloud is its elasticity that is the ability to dynamically acquire or release computing resources in response to demand. However, this elasticity is only meaningful to the cloud users when the acquired VMs can be provisioned in time and be ready to use within the user expectation. The long unexpected VM startup time could result in resource under-provisioning, which will inevitably hurt the application performance [9]; hence it is vital to evaluate the VM startup and release time to help cloud users to plan ahead and make in-time resource provisioning and releasing decisions.

A systematic study of VM provisioning and releasing times has been done considering different factors as follows:

6.1. Number of VMs

The average startup time of VMs in our cloud is considered 16 seconds. When the number of VMs requested increase the time increases accordingly. This is due to that CloudStack handles each VM requested as if it is being launch individually. This is depicted in figure 12. The provisioning time of 2 VMs request is 31 seconds which equals the sum of two VMs startup time requested alone, and the same applies for VMs release time as shown in figure 13. The average VM destroy time is 27 seconds and for 5 VMs is 142 seconds which equals the sum of a 5 VMs release time.

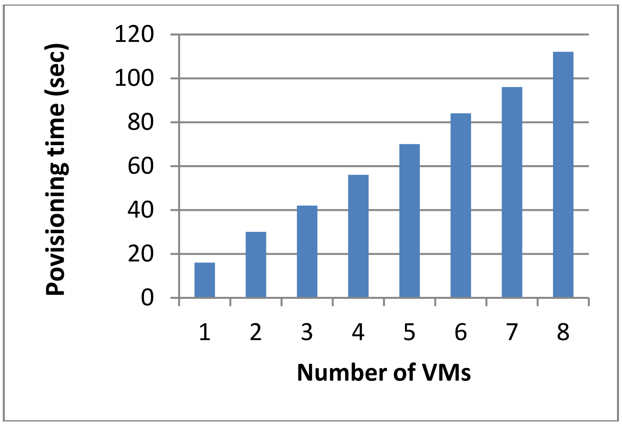

Figure 12. VMs Startup Time vs. Number of VMs

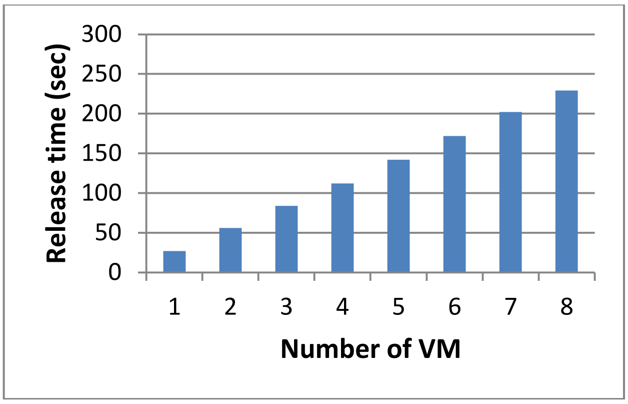

Figure 13. VMs Release Time vs. Number of VMs

6.2 Type of VMs

The VM provisioning and release time is not affected by its type as illustrated in figures 14 and 15. VMs with different types have almost the same startup and release time around 16 and 28 seconds respectively. This reveals the satisfactory and quick VM resource allocation schedulers of CloudStack .

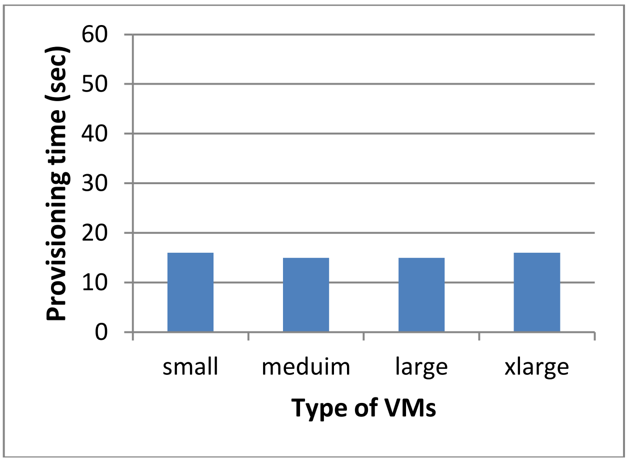

Figure 14. VMs Startup Time vs. Type of VMs

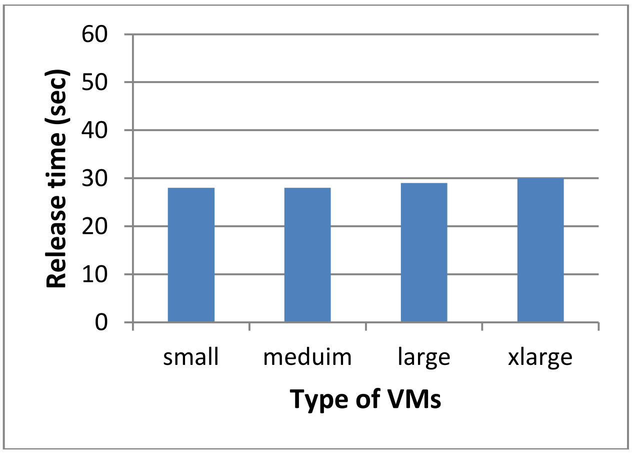

Figure 15. VMs Release Time vs. Type of VMs

6.3 Image Size

The VM provisioning time is not affected by size of the image or template used to initiate it as depicted in figure 16. VMs with different image sizes have almost the same startup time around 16 seconds. This is due to using primary storage as shared disk for VMs in CloudStak access via NFS instead of local disk configuration. So there is no need to copy templates (of different size) from secondary storage to hosts which results in reducing the time of VM startup regardless of template size.

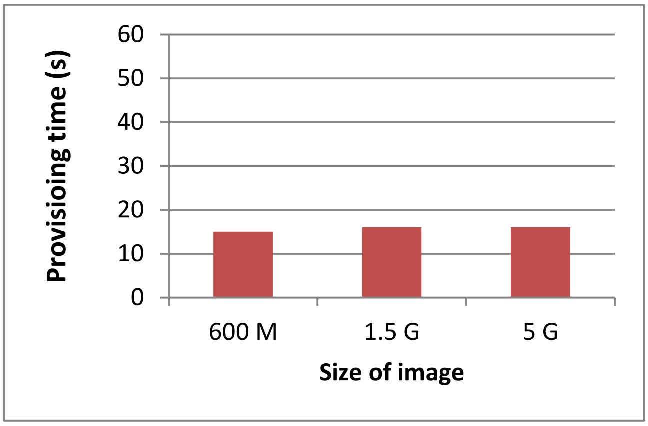

Figure 16. VMs Startup Time with Different Image Sizes

6.4 Additional Disk Space

The VM provisioning and release time is not affected by adding additional disk volumes during VM creation as being requested by user. The VM startup and release time remains almost the same, when adding different disk size to VM at 16 and 29 seconds respectively as shown in figures 17 and 18. This is probably due to that CloudStack uses the primary storage to provide disks to VMs with a quick resource allocation.

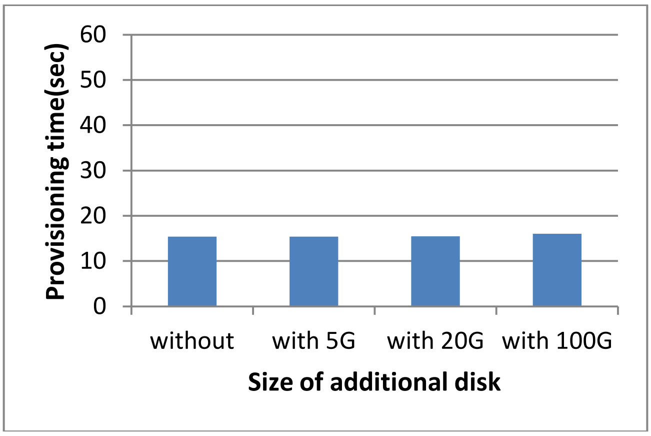

Figure 17. VMs Startup Time vs. Additional Disk

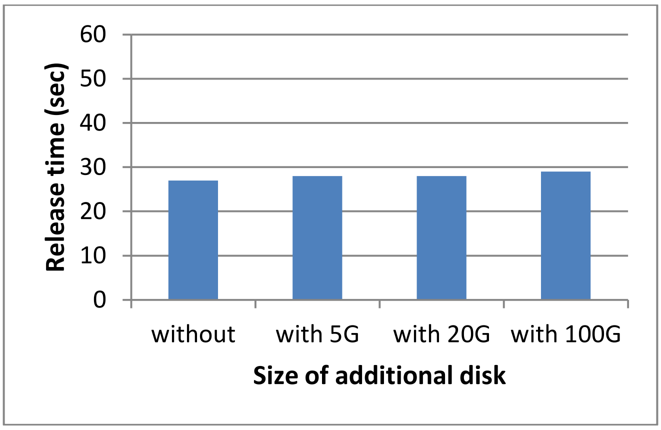

Figure 18. VMs Release Time vs. Additional Disk

7. PERFORMANCE EVALUATION of LIVE MIGRATION

Live migration is the movement of a virtual machine from one physical host to another while continuously powered-up and running, taking in consideration that the migration is not disrupting any active network connections. When properly carried out, this process takes place without any noticeable effect from the point of view of the end user [21]. Live migration is an important feature for host failure recovery and load balancing between hosts. Time duration of VM live migration has been expressed considering different factors as pursued.

7.1 Image Size

Duration of VM migration is affected by image or template size used to initiate it as illustrates in figure 19. There is no difference when using 1GB and 5GB image size for VMs using shared disk. There is no need to move data disk from source host to destination host. That is the size should not affect migration time. However when 600MB image has been deployed, it takes a shorter time than 1GB and 5GB. This is due to that these are GUI OS image while 600M is non. This means that its applications consume less CPU and memory; the context switch compromising CPU status and memory pages copied from source to the destination host, is of small size thus it migrates faster.

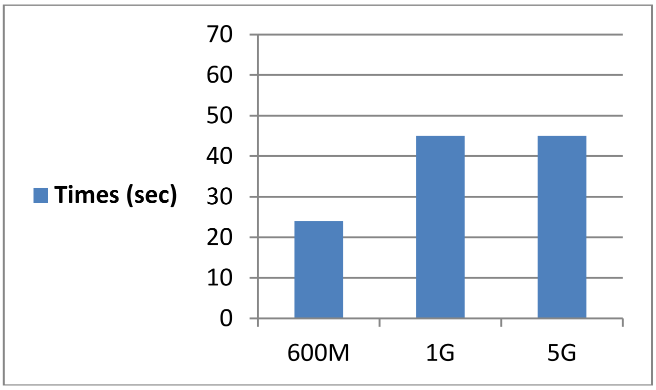

Figure 19. Live Migration with Different Image Size

7.2. Type of VM

The migration time of different types of VMs running normal application has been estimated. The type of VM can largely affect the duration of migration as depicted in figure 20. This is due to increasing memory size assigned to VM in each type so the duration of live migration increases linearly with it.

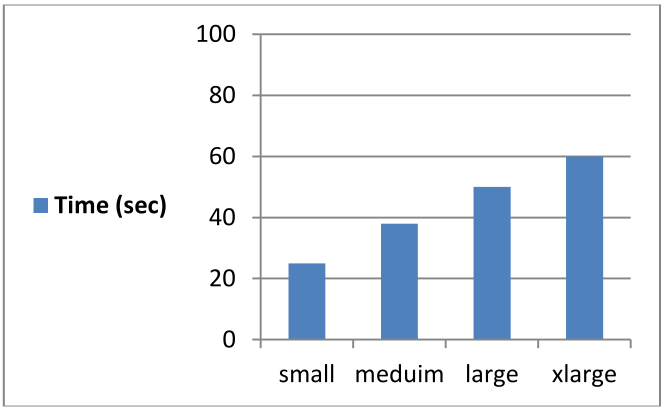

Figure 20. Live Migration with Different VM types

7.3 Number of VMs

The average time of live migration of VM in our cloud is 40 seconds; when the number of VMs that was migrating at the same time increase this time increases accordingly, as it depicts in figure 21.

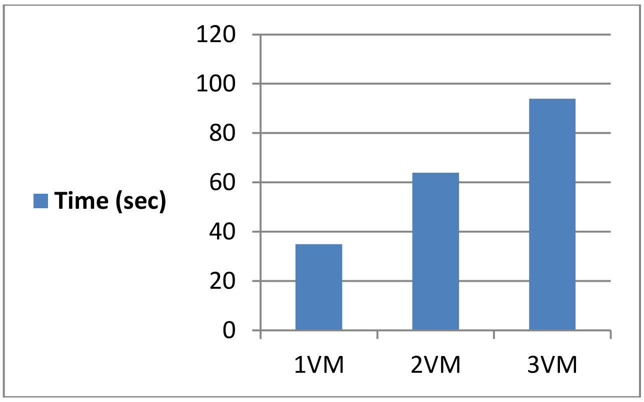

Figure 21. Live Migration with Different Number of VMs

7.4 CPU Load

The migration time of VM when the CPU is running an intensive application has been measured to assess its affect on migration as a relating factor. We have tested two types of VMs, medium and large and have used Lookbusy tool to generate a 90% CPU utilization. We have found that the CPU load can have an impact on the duration of migration as shown in figure 22. We can conclude that live migration depends on CPU utilization and applications running on the VM.

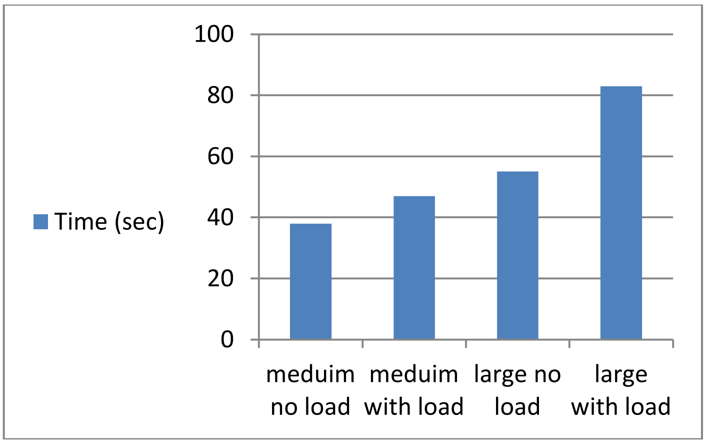

Figure 22. Live Migration vs. CPU load

8. CONCLUSION

This paper has analyzed the performance of CloudStack private cloud thoroughly to assess its suitability to be adopted as an open source cloud solution for different business and scientific purposes. We have considered performance of VM in terms of CPU, disk I/O, Memory, Network and VM management operations as add, delete and live migration. CloudStack appeared to have a well defined internal architecture, as concluded from its stability and performance of VM. It provides a good CPU and network isolation between VMs but there is disk I/O interference between VMs. The major lessons learned related to the performance evaluation of VM management operation are: (1) the duration for the live migration changes with the CPU load; (2) the duration for the live migration increases linearly as the memory assigned to the VM increases; (3) the startup and release time have not been impaired by the VM type or the image size; neither this is affected by adding additional disk volumes; (5) memory performance decreases as number of VMs increases.

As a future work we intend to analyze security aspects of CloudStack by evaluating the compliance of it with security standards related to cloud security.

REFERENCES

[1] E. T. de Sousa, et al, " Evaluating Eucalyptus Virtual Machine Instance Types: A Study Considering Distinct Workload Demand ", Third International Conference on Cloud Computing, GRIDs, and Virtualization, pp. 130-135, July 2012.
[2] H. Tianfield, "Cloud Computing Architectures", Proceeding of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1394-1399, Oct 2011.
[3] P. Mell and T. Grance, “The NIST Definition of Cloud Computing” National Institute of Standards and Technology Special Publication 800-145, September 2011.
[4] NEC Company, Ltd.and Information and Privacy Commissioner, Ontario, Canada, " Modeling Cloud Computing Architecture Without Compromising Privacy: A *Privacy by Design* Approach ", May 2010.
[5] Z. Pantic and M. A. Babar, "Guidelines for Building a Private Cloud Infrastructure ", Technical Report TR-2012-153, IT University of Copenhagen, Denmark, 2012.
[6] C. El Amrani, et al, "A Compartive Study of Cloud Computing Middleware", Proceeding of the12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 690-693, 2012.
[7] I. Voras, et al, "Evaluating Open-Source Cloud Computing Solutions", Proceedings of the 34th International Convention MIPRO, pp. 209-214. May 2011.
[8] G. V. Laszewski, et al, "Comparison of Multiple Cloud Frameworks", proceeding of the 5th IEEE International Conference on Cloud Computing (CLOUD 2012), pp.734-741, June 2012.
[9] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud ", Proceeding of the 5th IEEE International Conference on Cloud Computing (CLOUD 2012), pp.423-430, June 2012.
[10] D. Steinmetz, et al, "Cloud Computing Performance Benchmarking and Virtual Machine Launch Time", Proceedings of the 13th annual conference on Information technology education, ACM New York, pp. 89-90, 2012.

[11] F. Gomez-Folgar, et al, " Performance of the CloudStack KVM Pod Primary Storage under NFS Version 3", Proceeding of 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, pp. 845-846, 2012.
[12] L. Huang and Y. Yang, "Facilitating Education using Cloud Computing Infrastrucure", Proceeding of the 6th CCSC south region conference, Journal of Circuits, Systems and Computers, volume 28, number 4, pp.19-25, April 2013.
[13] Apache CloudStack 4.0.1-incubating, CloudStack Administrator's Guide, Apache CloudStack, 2012.
[14] A. Huang, Apache CloudStack (Incubating) An Introduction, http://www.citrix.co.jp/event/pdf/1-introduction.pdf, retrieved August 2013.
[15] http://software.intel.com/en-us/articles/intel-math-kernel-library-linpack-download, retrieved August 2013.
[16] J. J. Dongarra, "Performance of Various Computers using Standard Linear Equations Software", Report, University of Manchester, February 5, 2013.
[17] http://www.devin.com/lookbusy , retrieved August 2013.
[18] http://www.coker.com.au/bonnie++, retrieved August 2013.
[19] http://www.cs.virginia.edu/stream , retrieved August 2013.
[20] http://iperf.fr/, retrieved August 2013.
[21] A. Agarwal and R. Shangruff, "Live Migration of Virtual Machines in Cloud", International Journal of Scientific and Research Publications, Volume 2, Issue 6, pp.1-5, June 2012.

BIOGRAPHY OF AUTHORS

Mumtaz AL-Mukhtar is associate professor at Information Engineering College in AL-Nahrain University, Iraq. He received M.Sc. degree in Technical Cybernetics from Czech Technical University, Prague in 1979. He received M.Sc. in computer engineering in 1989 and in 2001 he earned his Ph.D. in the same field from the University of Technology, Iraq. He advised tens of Ph.D dissertations and Master theses. He has published widely in journals and conferences. His research interests include distributed systems, cloud computing, wireless sensor networks, social networks, pervasive computing, and mobile learning. He also holds an Adjunct position of electrical& computer engineering at Michigan State University, USA.

Asraa Abdulrazak Ali Mardan is received the Bachelor of Information Engineering degree from AL-Narain University, Iraq in 2010. Currently she is an MS.c student in Al-Nahrain University, College of Information Engineering, Department of Networks Engineering. Her research interests are Cloud Computing and Virtualization Technology.