Comparison of Various Virtual Machine Disk Images Performance on GlusterFS and Ceph Rados Block Devices

Johanes Joseph Johari, Mohammad Fairus Khalid, Mohd Nizam Mohamad Mydin, Nazaruddin Wijee,
Mimos Berhad,
Technology Park Malaysia, 57000 Kuala Lumpur ,Malaysia
{johanes.joseph, fairus.khalid, nizam.mydin, naza.wijee}@mimos.my

ABSTRACT

High availability, scalability and performance in addition to security are some of the important factors for Cloud Computing technology providers have to address. Open sourced solution enable new technology to be widely distributed, tested and improved. Currently we are experimenting open sourced distributed storage system namely Ceph and GlusterFS. Both are addressing the concern stated earlier. We setup two Cloud Computing environments which one is configured with Ceph and another with GlusterFS. We compare the performance of various virtual disk images such as QCOW2, RAW, QED on GlusterFS and Ceph RADOS Block Devices (RBD). We also present our analysis and findings in this paper.

KEYWORDS

virtual machine disk image, cloud computing, GlusterFs, Ceph RBD, performance.

1 INTRODUCTION

High availability, scalability and performance are few qualities attributes that need to be considered before adopting Cloud Computing services. One aspect of these quality attributes is the storage system. In our Cloud Computing architecture investigation we are using OpenNebula[1] as our cloud management software and KVM[11] as the hypervisor. In order to meet the high availability, scalability and performance requirements of the storage system we are evaluating 2 distributed storage solutions i.e. GlusterFS[3] and Ceph[2].

In this paper we are comparing the performance of various types of virtual machine disk images. The disk benchmark such as read, re-read, write and re-write of QED, RAW, QCOW2 and Ceph RBD of disk images are collected and analyzed.

This paper is organized as follows. Section 2 introduces the technological background. Section 3 talks about test methodology. Section 4 discusses the results and section 5 summarizes our work and future possibilities.

2 BACKGROUND

This section we will briefly introduce the Ceph and GlusterFS storage system; Cloud Computing management layer and Virtualization layer.

2.1 CEPH

Ceph was created by Sage Weil for his doctoral dissertation in 2007 Ceph: Reliable, Scalable, and High Performance Distributed Storage at the University of California, Santa Cruz[4]. After his graduation he continued to work on it and released the first major stable release called Argonaut in July 2012.

Ceph is an open source, object distributed file systems which uses the underlying RADOS, a reliable object storage service that can scales to many thousands of devices[5] and utilizes CRUSH algorithm[6], a scalable pseudo-random data distribution function designed for distributed function

designed for distributed object-based storage systems . It has 3 different implementations to interact with, which is the POSIX-compliant file-system, block device and object storage.

2.2 GlusterFS

GlusterFS is an open source, distributed file system which utilizes elastic hashing algorithm and is designed using a modular stackable architecture. GlusterFS does not separate metadata form data, and does not rely on any separate metadata server, whether centralized or distributed. It can be mounted by its own native protocol via the FUSE mechanism, using NFSv3 protocol, using a built in translator, or accessed via gfapi client library[7].

We used the FUSE client to store the images in OpenNebula datastore which accesses the volume of the GlusterFs filesystem.

GlusterFS uses the concept of storage bricks to store data. It has replication, striping and scales horizontally as well.

Data is stored on servers which are then tied to bricks or sub volumes. These bricks are then combined to form volumes. We created a volume and tied two bricks to it. Replication was also configured to two replicas. We installed the version of GlusterFS which is 3.4.3 and mounted the FUSE client for OpenNebula datastore to store the virtual machine disk images inside.

2.3 CLOUD COMPUTING

OpenNebula is an open source Cloud Computing toolkit for managing heterogeneous distributed data center infrastructure. OpenNebula orchestrates storage, network, virtualization and security technologies to deploy multi-tier services as virtual machines on distributed computing. For our setup we utilize the datastore to store images and run the same template to launch virtual machines with the correct and same settings regardless of the distributed storage/file-system in the backend. This will ensure that any virtual machines settings are consistent and correct during the testing.

2.4 VIRTUAL MACHINE DISK IMAGE

The image formats selected are RAW, QED, QCOW2 normally used in a cloud setup apart from Ceph RBD which will be briefly explained.

QCOW2 (*Qemu Copy on Write version 2*) is an updated version of QCOW (Qemu copy on write) format. QCOW is a file image for disk image used by QEMU, a hosted virtual machine monitor[8]. It has features like multiple virtual snapshot, optional AES encryption, zlib-based compression, a flexible model for storing snapshots and most use it to have smaller images on file-systems that do not support holes. Most commonly used as it is the most versatile format.

QED stands for *QEMU enhanced disk format.* QED is an image format that supports backing files and sparse images[9]. It is similar to qcow2 and has new features like strong data integrity and fully asynchronous I/O path.

RAW is a plain binary image of disk image format and is very portable. Images in this format only use the space allocated by the data for it. It is normally used as the default image format. Its advantage is its simple and easily exported to all emulators.

Ceph RADOS Block Device is thin-provisioned, resizable and store striped data over multiple OSDs in a Ceph cluster. The block device can be rapidly resized, snapshotted, and cloned. It is integrated with libvirt/QEMU/KVM virtual clients and also can be replicated according to the policies that you have defined. RADOS preserves consistent data access and strong safety semantics while allowing nodes to act semi-autonomously to self-manage replication, failure detection, and failure recovery through the use of a small cluster map[10]. There are two ways to access RBD, through the RBD kernel module and

RBD QEMU client through librbd. In this setup we will utilize the QEMU RBD client.

2.5 VIRTUALIZATION

We are using KVM as our virtualization layer. KVM is a linux kernel module that allows a userspace program to utilize the hardware virtualization features of various processors [11]. This will be installed on nodes where the virtual machines will be launched.

QEMU is generic and open source machine emulator and virtualizer[12].A huge number of hardware architectures can be emulated and can run unmodified operating systems. The primary usage of QEMU is to run another operating system on existing one, such as Windows on Linux or Linux on Windows system and all its application in a virtual machine[13].QEMU can be run together with KVM kernel module and take advantage of KVM acceleration.

Virtio is a paravirtualized driver for kvm/Linux. It is the Input/Output virtualization in KVM[14].It is a feature for network and device drivers that will be used by the hypervisor to talk to the virtual machines as it is faster compared to default drivers. It is a series of efficient, well-maintained Linux drivers which can be adapted for various different hypervisors implementations using a shim layer[15].

3 TEST METHODOLOGY

3.1 TOOLS USED

We are using iozone[16] as a tool to get the measurement and throughput data on the storage. The iozone command used and run is iozone –a –s 1g –q 256 –I –i 0 –i 1 –i 2 -+u –R –b testresults.xls. The command is executed from inside the virtual machine. Table 1 describes the command in detail.

Table 1 The Description of Iozone command syntax.

Syntax	Description
-i	Which test to run 0=write/re-write, 1=read/re-read, 2=random-read/write.
-I	Use direct io. It bypass the buffer cache and go directly to disk
-s	The size of file in Kbytes to test.
-a	Covers all test from of 4k to 16M for record size
-b	the filename to store the output.
-q	Maximum record size (in Kbytes)

3.2 TEST ENVIRONMENT

Below are the specifications of servers and clients used in the setup.

Table 2 The specifications of Storage Servers.

Model	Intel®Xeon®CPU E5405@2.00GHz
Memory	8G
Ethernet	NetXtreme BCM5722 Gigabit Ethernet PCI Express
Hard disk	1x 80Gb sata
	1x 1Tb sata 7200 rpm
Cores	4

Table 3 The specification of client servers.

Model	Intel®Xeon®CPU E5606 @2.13 GHz
Memory	8G/6G
Ethernet	3c940 10/100/1000Base-T Marvell
Hard disk	1x 700Gb sata 7200 rpm
Cores	4

Ubuntu 12.04 LTS 64 bit on all servers and upgraded to the kernel version 3.8.0. Ceph firefly version 0.80.1 and GlusterFS 3.4.3 were used. All clients were installed with libvirt 0.9.8 and qemu-kvm.1.0. We used only two nodes for

performance testing. As for replication between storage servers, 1TB of volume is replicated over two nodes. The secondary hard-disk is formatted using XFS filesystem for the storage. Although Btrfs is better in term of performance compare to XFS, we opted with XFS due to its maturity. The storage disk is on a separate secondary hard-disk on the servers as we did not want to mix the operating systems files with Ceph storage files.

Figure 1 and 2 shows the setup diagram of Ceph and GlusterFS.

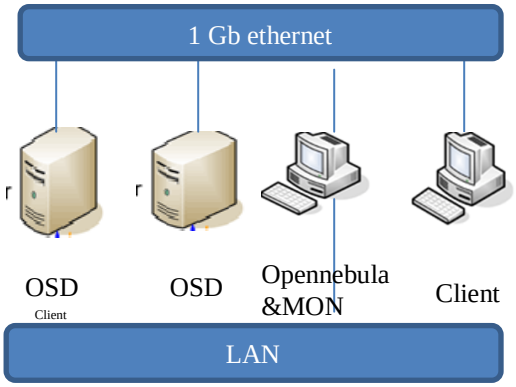

Figure 1 Ceph RBD setup diagram

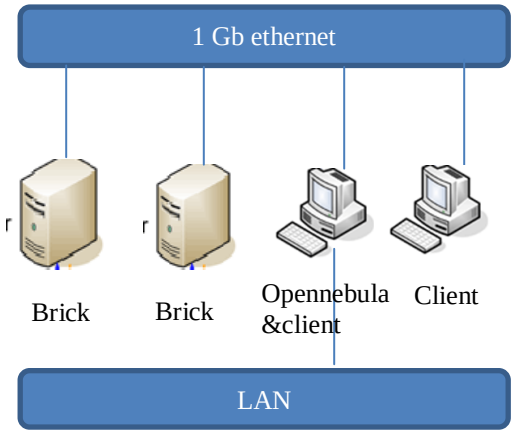

Figure 2 GlusterFS setup diagram

3.2 GLUSTERFS SETUP

The GlusterFS is deployed with two storage servers or two bricks and two clients to access the volumes. One of the clients is installed with OpenNebula frontend. We mounted GlusterFS using FUSE client to a volume which was created from a sub volume then configured as a OpenNebula frontend datastore as a shared storage where all the virtual disk images are stored.

3.3 CEPH SETUP

The current Ceph setup, there is one Ceph-monitor (MON) and two object storage devices (OSD). Since we weren't going to test the file-system and only test block device, we just deploy the ceph-mon without metadata server, ceph-mds. OpenNebula frontend is deployed on the same node as ceph-mon. We created a pool and changed the default replication size to two from three. The secondary hard-disk on both ceph-osd nodes were formatted with XFS and dedicated as storage nodes, as we did not want to mix the operating system files in the first hard disk. No additional tuning was done. We created a pool called 'one' with the command "ceph osd pool create 'one' 128 128". This pool would be configured to create the Ceph RBD datastore in OpenNebula.

3.4 NETWORK SETUP

A dedicated 3com Baseline 2816 switch with a network throughput of 1GbE is used in the backend. All servers are configured with a 1G Ethernet and no bonding was configured. Iperf a commonly used network testing tool[18], was used to see the throughput from client to server node .We measured a maximum throughput value of 944 Mbits/sec and minimum throughput of 469 Mbits/sec between client and servers.

3.5 VIRTUAL MACHINES

For Ceph setup, we created Ceph RBD datastore in OpenNebula and for ClusterFS setup we mounted a volume to store the images. The Virtual Machine was created with an operating system using Ubuntu 12.04 LTS with 4GB of memory.

ISBN: 978-1-941968-00-0 ©2014 SDIWC

4 OBSERVATIONS

The result is presented in graphical form as below and they are arrange into read, write, re-read and re-write. The x-axis represents the file block sizes and y-axis represents the bytes copied/written. Each column on x –axis is separated into Ceph RBD, image types on GlusterFS performance.

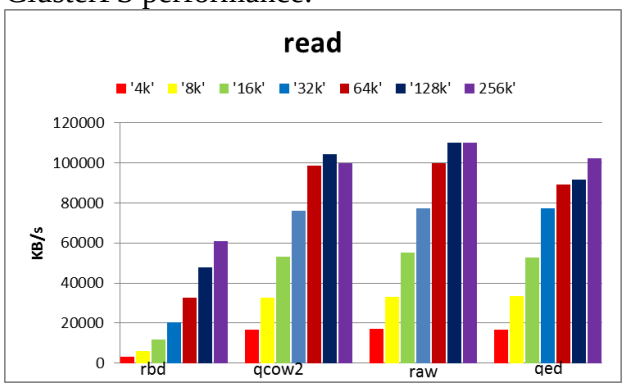

Figure 3 Read throughput

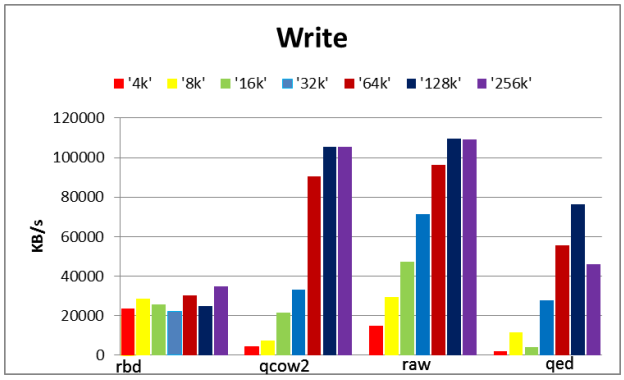

Figure 4 Write throughput

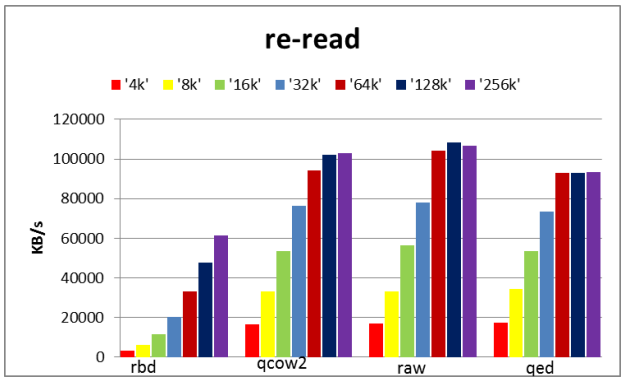

Figure 5 Re-read throughput

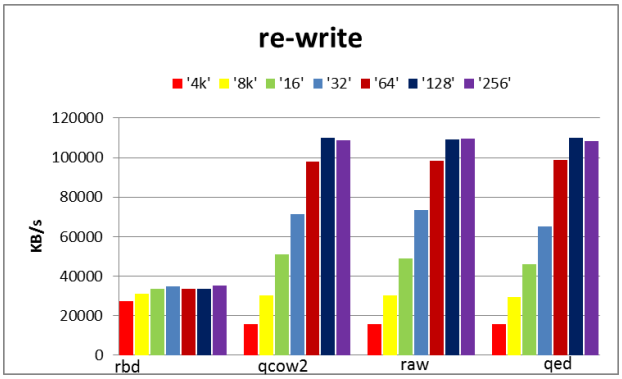

Figure 6 Re-write throughput

4.1 IO TEST RESULTS

In Figure 3, read raw disk image also perform better than qcow2 disk image and only slightly better results when block sizes are 128k and 256 k.

In Figure 4, write with small sizes of block size of 4k and 8k Ceph RBD performs better than other formats but when the block size increases the performance drops.

In Figure 5, re-read raw images perform better than qcow2 images and only slightly better when the block size of file is 128k and 256k.

In Figure 6, re-write for sizes of block size of 4k and 8k on Ceph RBD performs better than other virtual image disk but when the block size increases the size the performance drops and qcow2 images performance better after block size increase to 16k.

5 CONCLUSIONS

The results from this study have been obtained with a small setup consisting of four servers. Two are storage nodes and another two are clients. Considering the results presented here, it should not be compared to a larger deployment even if the use case and scenario

ISBN: 978-1-941968-00-0 ©2014 SDIWC

are similar. The results here showed that GlusterFS for applications which do read and writes with block sizes ranging between 16k-256k is better compared to Ceph RBD. QCOW2 virtual image disk format on top of GlusterFS is recommended compared to raw images even though raw performance is slightly better in some cases due to its features which are more advanced. Interestingly we would like to point out that in the case of writing, or rewriting small sizes of files between 4k-8k, Ceph RBD would seem a better choice for some applications.

For future work we would like to explore some parameters of Ceph and GlusterFS, for example stripe size. Ceph block stripe size is 4Mb while GlusterFS uses 128kb to a similar size. Since Ceph and GlusterFS itself has some parameters to tune it would interesting to see the results when the same test is done with all the parameters utilized.

As for the network portion we tested on a dedicated local area network in the backend and would be interesting to test instead with a Software Defined Switch (SDN). This would be interesting due to the fact we could test concepts like Quality of Service (QoS) for example and see if it has any substantial effect on performance.

Another portion was to look at the storage devices itself. Since we used SATA hard disk for the storage, it would be interesting to see if we could use normal SSD storage and at least double the amount of storage devices and clients to the current setup. Nevertheless preliminary testing has enabled us to get a slight peek in the performance. With more storage and more client nodes it would be possible to gauge a larger datacenter environment performance and scalability.

6 ACKNOWLEDGEMENTS

I would to thank those who have assisted us especially to our colleagues in Advanced Computing Lab, MIMOS for their input, advice and assistance.

7 REFERENCES

[1] OpenNebula http://opennebula.org

[2] Ceph Community http://ceph.com

[3] Gluster, Gluster Community https://www.gluster.org.

[4] S.A.Weil , S. A. Brandt, E. L.Miller,D. D.E.Long,C.Maltzahn "Ceph:A Scalable,High-Performance Distributed File System",Ph.D. thesis , University of California, Santa Cruz, December 2007

[5] S.A.Weil, A.W.Leung, S.A.Brandt, C.Maltzahn "RADOS: A Scalable Storage Service for Petabyte-scale".PDSW '07 Proceedings of the 2nd international workshop on Petascyle data storage:held in conjunction with Supercomputing '07, pp. 35-44.

[6] S.A.Weil, S.A.Brandt, E.L.Miller, C.Maltzahn, "CRUSH:Controlled, Scalable,Decentralized Placement of Replicated Data". SC '06 Proceedings of 2006 ACM/IEEE conference on Supercomputing Article No.122.

[7] http://en.wikipedia.org/wiki/GlusterFS

[8] "QEMU Emulator User Documentation". Wiki.qemu.org.

[9] http://wiki.qemu.org/Features/QED

[10] http://www.linux-kvm.com/content/qed-qemu-enhanced-disk-format

[11] http://wiki.qemu.org/KVM

[12] http://wiki.qemu.org/Main_Page

[13] https://www.usenix.org/legacy/event/usenix05/tech/freenix/full_papers/bellard/bellard_html/

[14] http://www.linux-kvm.org/page/Virtio

[15] R.Russell "virtio:Towards a De-Facto Standard For Virtual I/O Devices.

[16] Iozone Filesystem Benchmark http://www.iozone.org/

[17] F.Wang, M.Nelson, S.Oral,S.Atchley, S.Weil, B.W.Settlemayer, B.Caldwell, J.Hill, "Performance and Scalability Evaluation of Ceph Parallel File System".

[18] Iperf http://en.wikipedia.org/wiki/Iperf