Block Level Storage Support for Open Source IaaS Clouds

Sandor Acs, Mark Gergely, Peter Kacsuk
Computer and Automation Research Institute
MTA SZTAKI
H-1518 Budapest, P.O. Box 63, Hungary
{*acs.sandor, gergely.mark, kacsuk.peter*}*@sztaki.mta.hu*

Miklos Kozlovszky
University of Obuda
OE NIK
H-1300 Budapest, PO BOX 267.
kozlovszky.miklos@nik.uni-obuda.hu

***Abstract*—Cloud computing is the dominating paradigm in distributed computing. The most popular open source cloud solutions support different type of storage subsystems, because of the different needs of the deployed services (in terms of performance, flexibility, cost-effectiveness). In this paper, we investigate the supported standard and open source storage types and create a classification. We point out that the Internet Small Computer System Interface (iSCSI) based block level storage can be used for I/O intensive services currently. However, the ATA-over-Ethernet (AoE) protocol uses fewer layers and operates on lower level which makes it more lightweight and faster than iSCSI. Therefore, we proposed an architecture for AoE based storage support in OpenNebula cloud. The novel storage solution was implemented and the performance evaluation shows that the I/O throughput of the AoE based storage is better (32.5-61.5%) compared to the prior iSCSI based storage and the new storage solution needs less CPU time (41.37%) to provide the same services.**

***Keywords*-Cloud Computing; Storage Area Network; ATA-over-Ethernet; iSCSI;**

I. Introduction

Cloud computing, based on the results of virtualization technologies and gathered from experiences of designing grid and cluster computing, became a successful paradigm of the distributed IT infrastructures [1]. The consumers of the clouds reach the resources through three major service models (Software/Platform/Infrastructure-as-a-Service) [2]. The Infrastructure-as-a-Service (IaaS) layer is the lowest level service model and it serves fundamental IT resources (e.g. CPU, storage, networking). Open source technologies and converged networks [3] [4] are cost-efficient IaaS system building blocks, therefore we focus on open source IaaS solutions and Gigabit Ethernet (GE) [5] networks in this paper.

Related works already disclosed that the performance of communication intensive services is degraded by the virtualized I/O subsystem [6] [7]. Hence, we investigate the storage subsystem of open source IaaS clouds (i.e. Eucalyptus [8], OpenStack [9], Nimbus [10], CloudStack [11] and OpenNebula [12]) with respect to I/O throughput performance. All of the IaaS clouds support distributed and block level storages as well. The advantage of the distributed storages is that the total capacity of the cloud storage can be increased easily by adding new storage components into the system and the redundancy can be handled in the software layer without the need of any special hardware or Redundant Array of Independent Disks (RAID) technique. However, this kind of storage has lower I/O throughput and higher latency [13] compared to block level storages. Accordingly, block level storages are preferred for running I/O intensive applications and open source IaaS clouds use iSCSI [14] for block level storage support. However, the ATA-over-Ethernet (AoE) based storages could provide higher I/O throughput [15] and lower latency [16] in small and middle-scale cloud infrastructures, where the components are located on the same (data link layer [17]) Local Area Network (LAN) segment. Hence we designed and implemented an AoE based storage prototype for OpenNebula. The implementation was tested on the same physical architecture as the prior iSCSI based storage and the results show that the I/O performance of the new AoE based storage subsystem is significantly better (32.5-61.5%) than the iSCSI based storage subsystem and at the same time the AoE server service uses less CPU time than the iSCSI server service. Accordingly, it can be declared that the AoE based storages are more advantageous than existing iSCSI based storage solutions for I/O intensive services in IaaS clouds where the components of the system are located on the same LAN segment.

This paper is organized as follows: first, we introduce the storage solutions of the current open source IaaS clouds in Section II. In Section III, we propose the classification of the IaaS storage solutions. In Section IV, we describe the difference between the iSCSI and AoE protocol and present the novel AoE based storage prototype for OpenNebula. The performance evaluation results of the prototype are introduced in Section V. Finally, we conclude our research in Section VI.

II. Related work

We have investigated the storage subsystems of the most popular open source IaaS clouds.

Eucalyptus and **OpenStack** are interface compatible with the Amazon Elastic Compute Cloud (EC2) and Simple Storage Service (S3), accordingly they have two different type of storage subsystems. The S3 like component (Walrus and Swift) can store consumers' data, organized as objects and buckets on a distributed storage infrastructure, which can be even reached from outside of the cloud. The second component is responsible for providing similar functionality than Amazon Elastic Block Store (EBS) and it can interface with various storage systems (e.g. Network File System (NFS) [18], iSCSI). Block based volumes can not be shared between virtual machines (VM) and could not be reached from outside of the cloud. Eucalyptus supported AoE based block volumes. However it has been removed because the AoE protocol could not guarantee security by itself and the iSCSI is more flexible from the networking point of view.

Nimbus has an S3 interface compatible storage system as well, called Cumulus. It focuses on middle-scale infrastructures and it integrates existing storage solutions (e.g. NFS, General Parallel File System (GPFS) [18], Parallel Virtual File System (PVFS) [19], Hadoop Distributed File System (HDFS) [20]).

CloudStack has primary and secondary storages. Every cluster has at least one primary storage for storing the disk volumes that are used by the VMs of the cluster. The primary storage uses NFS or iSCSI protocol. The secondary storages are associated with zones. These elements can be reached from more than one clusters that makes them suitable for storing templates (OS images), ISO images (data or bootable media) and disk volume snapshots. CloudStack implements the Amazon S3 APIs and supports the Swift of the OpenStack.

OpenNebula has its own concepts, called datastore (DS) and transfer manager (TM), for storage operations in the cloud. A DS can be any storage medium used to store disk images for VMs. A DS is backed by storage area network (SAN) or by network attached storage (NAS) servers typically. The TM drivers contain low-level storage operations, which allows the disk images to be transferred to hosts machines. OpenNebula does not have any integrated distributed storage platform. However, the system architecture is highly flexible, hence already existing distributed storage solutions (e.g. Ceph File System [21], Distributed Replicated Block Device (DRBD) [22]) can be easily adopted.

The well-known open source IaaS systems use and support different types of storage systems because of the different needs of the users and their services.

III. Classification of Storage Systems for IaaS clouds

There are numerous kinds of storage systems that are used in IaaS architectures, as we already introduced in Section II. The following section describes some of the different features of storage systems in order to be able to categorize the available solutions.

- **Local/Remote**: If a VM uses the local resources (e.g. hard drive, solid-state drive, random-access memory) of the host to store virtual disk images, it is called local

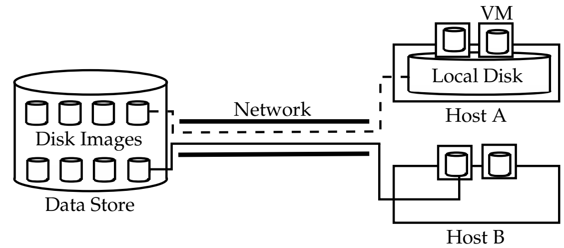

Figure 1. Local and remote storage

storage. In another case, the VM uses remote resources (e.g. shared file system, exported block device) that is called remote storage solution. Figure 1 shows a storage (called datastore in OpenNebula) and two hosts are presented. Host ”A” uses its local storage and Host ”B” uses only remote storages. The advantages of the local storages can be that the data and its processing are located on the same physical machine so the latency is low (if the local resources' load is low enough). Some services prefer to use local storages (e.g. HDFS of Hadoop). Another benefit is that the I/O load of central storages is decreased by using distributed resources [23]. However, remote storages are required in many cases because the centralized image management is essential for live migration, which is the procedure when a VM instance is moved from one host to the other without any outage (that can be detected by end-users).
- **Block/File level**: Virtual images can be stored directly on raw block storage devices (e.g. Physical disk, Logical Volumes [24]) or in file format, which requires file-system (FS) on the block storage device. Storages based on raw block devices support pure I/O operations (e.g. reading, writing) in fixed-sized blocks, sectors, or clusters. These kind of storages have less overhead than file based images [25], however files are easier to operate and use (e.g. copy, move, delete).
- **Shared/Non-shared**: A storage is a shared storage, if it can be reached by more than one host at the same time, as Figure 2 shows. Shared storages are always remote storages from the host machines' point of view. However, non-shared storages are not always locals. For example, a disk image connected by the remote iSCSI is not shared because it can be used exclusively only by one host at the same time.
- **Distributed/Non-distributed**: A storage system is distributed if the stored data is located on more than one storage machine, however it can be perceived as only one source from the usage point of view (Figure 3). The fault tolerance can be achieved by design in a distributed storage system. Storage redundancy can be built with using RAID techniques as well, however it can guarantee fault tolerance only inside each physical

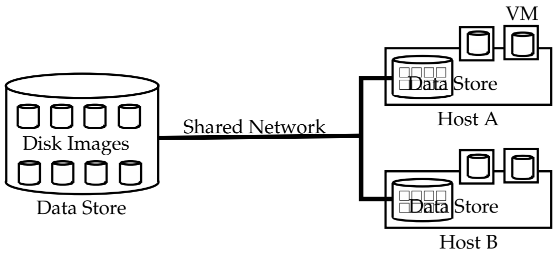

Figure 2. Shared storage

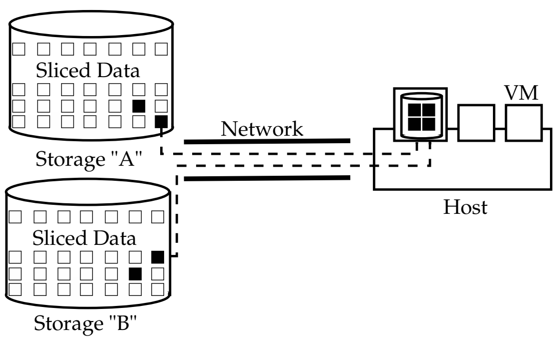

Figure 3. Distributed storages

storage server separately and it costs storage capacity overhead.

	Remote	Block	File	Shared	Distributed
NFS	Y	N	Y	Y	N
DRBD	Y	Y	N	Y	Y
CEPH	Y	N	Y	Y	Y
GlusterFS	Y	N	Y	Y	Y
Walrus	Y	N	Y	Y	Y
Swift	Y	N	Y	Y	Y
GPFS	Y	N	Y	Y	Y
PVFS	Y	N	Y	Y	Y
HDFS	Y	N	Y	Y	Y
AoE	**Y**	**Y**	**N**	**N**	**N**
iSCSI	**Y**	**Y**	**N**	**N**	**N**

Table I
COMPARISON OF THE STORAGES ACCORDING TO THE CLASSIFICATION

Table I shows the supported types of storages, categorized by the classification. It indicates that the different types of services prefer or require different types of storage systems.

If performance and reliability are the most important criteria for a service running on IaaS, the underlying storage system should be:

- **Remote** for the reliability, because VM instances should be able to live migrate from one host to another.
- **Block** storages can have better performance than the file based ones because they have less overhead.
- **Non-distributed** from performance point of view, because they can have less latency and higher write throughput than distributed storages.

In open source IaaS clouds, iSCSI storage solutions are sufficient for the criteria of running reliable services with high I/O requirements. However, there is another protocol as Table I shows, the AoE which has the same features like iSCSI and we have not found any publication about the AoE integration in open source cloud systems yet and the current open source IaaS systems does not support AoE based storage solutions.

IV. NOVEL STORAGE SUPPORT IN OPENNEBULA

The following sections describe the iSCSI and AoE protocols and they also focus on the new storage design in OpenNebula.

A. The iSCSI protocol

Internet Small Computer System Interface (iSCSI) is a standard Storage Area Network (SAN) protocol for connecting storage facilities over Internet Protocol (IP)-based networks. It is able to manage storage devices in LAN and from long distances as well because of the routable IP. The clients (called initiators) and storage devices (targets) use standard SCSI commands. The iSCSI protocol can use different authentication methods between the initiators and targets.

B. The ATA-Over-Ethernet protocol

The ATA-over-Ethernet (AoE) is a SAN protocol. It is designed to use standard technologies in low levels. AoE sends Advanced Technology Attachment (ATA) commands through Ethernet (data link layer) networks. AoE based storages cannot be reached over the Internet because the Ethernet protocol is not routable—unlike the IP. It cannot be secured by authentication methods, only with media access control (MAC) address.

C. Comparison of the protocols

The most significant difference between the protocols is that the AoE uses fewer protocol layers than the iSCSI which makes the:

- **AoE** is more lightweight, because it needs to operate in lower level;
- **AoE** is faster in Ethernet based networks; (However iSCSI is not restricted to Ethernet networks, it can be used over the Internet because of the routable IP base.)
- **iSCSI** is more secure, because of the supported authentication methods. (However, AoE based storages are used in dedicated LANs—in this manner, the MAC filtering could be sufficient.)

The iSCSI based storages are appropriate for general purposes IaaS systems because they can be secured and extended easily. However, AoE based storages are more suitable if the key aspect is performance.

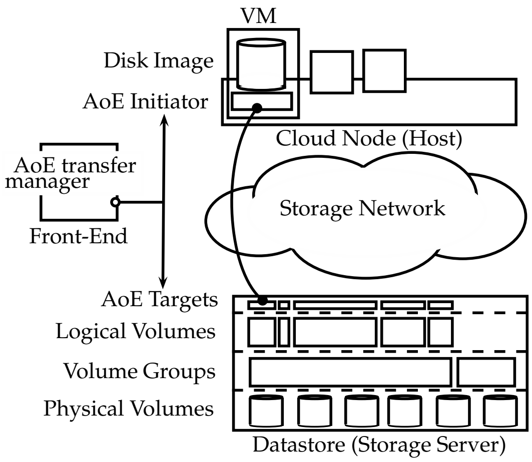

Figure 4. Host with AoE based storage

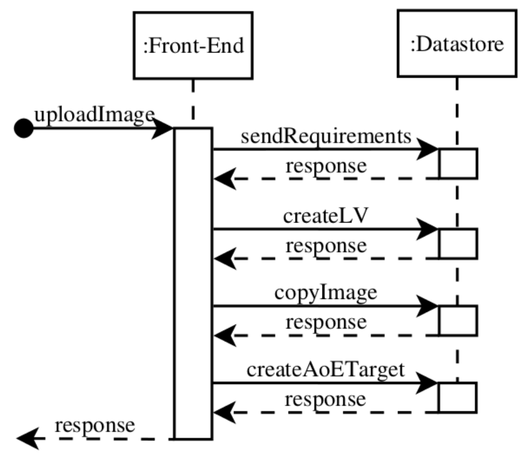

Figure 5. Upload disk image

D. AoE based storage design in OpenNebula

In order to use AoE based datastore, we designed an AoE transfer manager (TM) driver in OpenNebula. For creating the TM driver, the vblade [26] AoE protocol implementation and bash shell scripting were used. Our implementation does not harm other functionality of the original software. Even it can be used next to the prior datastore TMs on the same storage and cloud node servers at the same time. Our contribution, the AoE TM driver and the wrapper script can be found on the official development site [27] of OpenNebula.

Figure 4 presents a standard Linux based storage server, an Ethernet based storage network, a cloud node (host) with running VMs and the front-end machine with the AoE TM driver. The storage has physical volumes (e.g. HD, SSD) and the logical volume management (LVM) software tool. LVM bounds the physical devices into volume groups. The disk images of the VMs are stored in logical volumes (LVs), created on a volume group. LVs are block devices, therefore disk images can only be stored in raw format. For binding the operating system with the block devices, the AoE kernel module creates virtual devices on the storage server (AoE target) and on the cloud node (AoE initiator) as well.

Figure 5 describes the object interactions when a user uploads a new disk image into the cloud. The user can upload a disk image with an OpenNebula compatible interface (e.g. Sunstone WEB based front-end). The front-end checks the available free space of the datastore. If the available free space is more than the size of the image file, it orders the datastore to create a new LV with the size of the image. If the previous command has finished successfully then the front-end copies the image with the secure copy (SCP) [28] tool into the new LV of the datastore. After the copy, the datastore creates an AoE target virtual device and exports the LV through the new AoE target.

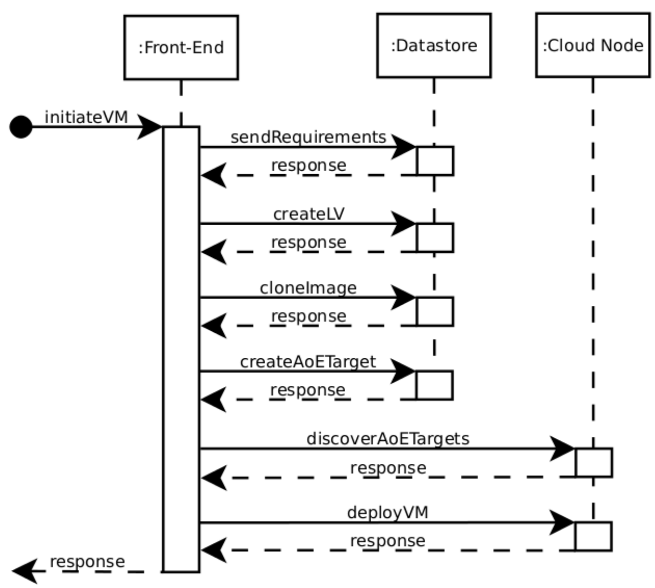

Figure 6. Inititate VM instance

In a cloud environment, it is common to upload a disk image once and use the same image for many VM instances. Therefore, Figure 6 presents the initiation of a new VM instance. The method of the VM initiation begins in a similar way as in case of uploading a new disk image. However, the image is already stored in the datastore so the corresponding LV is cloned instead of copied. After the new LV becomes available, the front-end sends an AoE discover command to the cloud node in order to refresh the cloud node's list of the available AoE devices. Then, the front-end starts to deploy the VM and the cloud node creates an AoE initiator and attaches the block device that is exported by the datastore. The attached block device can be understood as normal storage device from the VM instance's point of view.

A VM can have persistent or non-persistent storage. If an

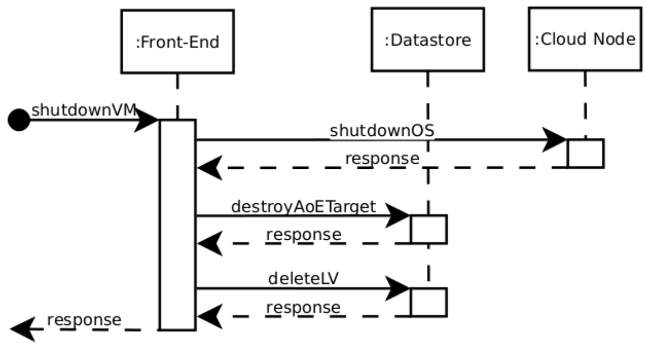

Figure 7. Shut down VM instance

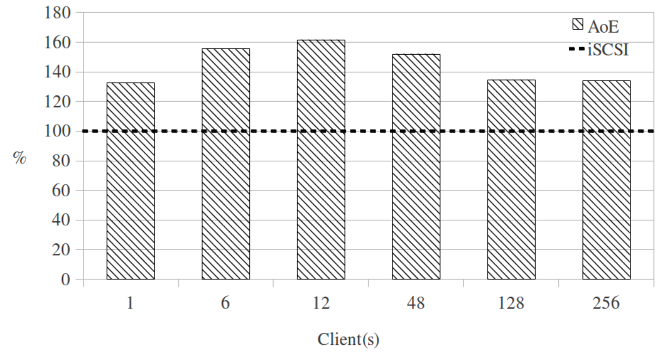

Figure 8. AoE I/O throughput performance compared to iSCSI

instance has persistent storage that means the changes (e.g. new files) will be kept after a shut down. However, if the VM uses non-persistent storage then the virtual disk (LV) will be deleted after a shut down. Figure 7 presents a non-persistent instance shut down from the AoE TM drive's point of view. The user sends the ”terminate VM instance” command to the front-end. The front-end forwards the request to the cloud node and it shuts down the Operating System (OS) of the VM. Afterwards, the datastore destroys and removes the corresponding AoE target and deletes the LV of the image.

V. Evaluation

We set up a test environment in order to evaluate the iSCSI and the new AoE based storage solutions. A storage server and a host machine were connected directly through a dedicated Gigabit Ethernet network. The performance tests were executed inside a KVM [29] based VM instance, that contained the I/O benchmark software. The host machine ran the VM instance and its image was stored on the Storage Server and connected to the host machine with the corresponding protocol. Table II shows the details of the used physical and virtual hardware.

	Type	CPU	RAM	Store
Storage	Dell R510	2x Xeon E5620	48G	12x 3TB
Host	Dell R815	4x Opteron 6262 HE	256G	6x 1TB
VM	KVM virt.	2x	2G	16GB

Table II
Test hardware

Client(s)	1	6	12	48	128	256
iSCSI [MB/s]	25.75	48.03	51.02	59.56	60.78	57.36
AoE [MB/s]	34.14	74.72	82.43	90.57	81.83	76.92

Table III
DBENCH Results

A SAN device is usually used by many clients at the same time, therefore we evaluated and compared the iSCSI and the AoE based storage solutions with high I/O load, generated by different number of clients. For this purpose, the DBENCH [30] tool was chosen, because it was designed to generate and measure I/O workloads either to a file-system or to a remote storage. DBENCH uses a concept called ”loadfile” which is basically a sequence of I/O operations in order to emulate various repeatable workloads. These workloads can be processed in parallel, simulating multiple clients, which is similar to the case with a remote storage device serving multiple virtual machines. Table III shows the bandwidth value of the performance tests that are the average transfer rates of the workloads.

In order to handle the deviation of the performance results, a benchmark framework called Phoronix Test Suite (PTS) [31] was used. PTS executes the benchmark at least 3 times, while monitoring the standard deviation. It executes new benchmark runs while the standard deviation is more than 3.5%. The results are the calculated average of the benchmark runs.

Figure 8 describes the differences between the benchmark values. The X axis represent the number of the emulated clients and the Y axis shows the AoE I/O throughput performance compared to iSCSI where the iSCSI values were the base (100%). Test results indicate the advantage in favour of the AoE protocol. The utmost performance gain was 61.5% at 12 clients, the least difference was 32.5% with one client. A typical Gigabit Ethernet iSCSI SAN is used by 10 to 100 clients [32], which coincides with the interval in the performance benchmark where the benefit of the AoE is higher than the average (49.5%).

During the DBENCH tests (described above), the CPU time consumption of the AoE and the iSCSI server services were measured on the Storage Server. The CPU time (in seconds) consumed by a process can be measured by using the top tool in Linux distributions. We used and compared the aggregated CPU times while the processes were executed in system (kernel) level because in that case they do not contain the overhead of either the waiting time for outstanding disk requests (I/O wait) or the idle times. The iSCSI server service consumed 3471 seconds in system level for completing the DBENCH test scenarios while the AoE needs 1436 seconds for completing exactly the same amount of tasks. The result shows that the AoE based storage solution needs less than half of the amount of CPU time (only 41.37%) for providing the same service.

VI. Conclusion and future work

In this paper, we have presented and investigated the storage solutions of the open source IaaS clouds and created a classification. We pointed out that the iSCSI based storages can be used for I/O intensive services currently, however the AoE protocol has better I/O throughput performance. Then, the differences between the two protocols have been discussed. We introduced a novel AoE based storage support for OpenNebula. Finally, we evaluated our contribution from performance point of view.

In the future, we will test the AoE based storage in different environments. For example, we want to add more resources (cloud nodes) to the test bed and change the Gigabit Ethernet to 10 Gigabit Ethernet connection, because the new storage solution should be evaluated in middle-scale cloud environments as well. We have already shared our contribution with the OpenNebula community through the official development site. However, we plan to support and keep our software contribution up-to-date.

Acknowledgment

The work described in this paper was partially supported by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 283770 (agINFRA), by TAMOP-4.2.1.B-11/2/KMR-2011-0001 "Kritikus infrastruktura vedelmi kutatasok", and by the Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI) under grant agreement SZTAKI Cloud project.

References

[1] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616, June 2009.

[2] P. Mell and T. Grance. The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[3] CM Gauger, M. Kohn, S. Gunreben, D. Sass, and S.G. Perez. Modeling and performance evaluation of iscsi storage area networks over tcp/ip-based man and wan networks. In *Proceedings Broadnets*, 2005.

[4] J. Jiang and C. DeSanti. The role of fcoe in i/o consolidation. In *Proceedings of the 2008 International Conference on Advanced Infocomm Technology*, page 87. ACM, 2008.

[5] M. Ko, D. Eisenhauer, and R. Recio. A case for convergence enhanced ethernet: Requirements and applications. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 5702–5707. IEEE, 2008.

[6] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. A performance analysis of ec2 cloud computing services for scientific computing. *Cloud Computing*, pages 115–131, 2010.

[7] D. Ghoshal, R.S. Canon, and L. Ramakrishnan. I/o performance of virtualized cloud environments. In *Proceedings of the second international workshop on Data intensive computing in the clouds*, pages 71–80. ACM, 2011.

[8] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on*, pages 124–131. IEEE, 2009.

[9] A. Beloglazov, S.F. Piraghaj, M. Alrokayan, and R. Buyya. A step-by-step guide to installing openstack on centos using the kvm hypervisor and glusterfs distributed file system. Technical report, Technical Report CLOUDS-TR-2012-3, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, 2012.

[10] J. Bresnahan, K. Keahey, D. LaBissoniere, and T. Freeman. Cumulus: an open source storage cloud for science. In *Proceedings of the 2nd international workshop on Scientific cloud computing*, pages 25–32. ACM, 2011.

[11] Fernando Gomez-Folgar, Antonio J. García-Loureiro, Tomás F. Pena, and Raúl Valín Ferreiro. Performance of the cloudstack kvm pod primary storage under nfs version 3. In *ISPA*, pages 845–846. IEEE, 2012.

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

[13] Shin gyu Kim, Hyuck Han, Hyeonsang Eom, and H.Y. Yeom. Toward a cost-effective cloud storage service. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 1, pages 99 –102, feb. 2010.

[14] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, and E. Zeidner. Iscsi. Technical report, RFC 3720, April, 2004.

[15] A.P. Gerdelan, M.J. Johnson, and C.H. Messom. Performance analysis of virtualised head nodes utilising cost-effective network attached storage. *Australian Partnership for Advanced Computing, number ISBN*, pages 978–0, 2007.

[16] Chuang He and Wenbi Rao. Modeling and performance evaluation of the aoe protocol. In *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on*, volume 1, pages 609 –612, nov. 2009.

[17] A.S. Tanenbaum. Distributed operating systems. *CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTS-CERN*, pages 101–102, 1996.

[18] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation of the sun network filesystem. In *Proceedings of the Summer USENIX conference*, pages 119–130, 1985.

[19] P.H. Carns, W.B. Ligon III, R.B. Ross, and R. Thakur. Pvfs: A parallel file system for linux clusters. In *Proceedings of the 4th annual Linux Showcase & Conference-Volume 4*, pages 28–28. USENIX Association, 2000.

[20] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. Ieee, 2010.

[21] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 307–320. USENIX Association, 2006.

[22] P. Reisner. Distributed replicated block device. In *Int. Linux System Tech. Conf*, 2002.

[23] S. Acs, P. Kacsuk, and M. Kozlovszky. Semi-shared storage subsystem for opennebula. In *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 51–56, 2012.

[24] D. Teigland and H. Mauelshagen. Volume managers in linux. In *Proc. of USENIX 2001 Technical Conference*, 2001.

[25] Michail D. Flouris, Renaud Lachaize, Konstantinos Chasapis, and Angelos Bilas. Extensible block-level storage virtualization in cluster-based systems. *Journal of Parallel and Distributed Computing*, 70(8):800 – 824, 2010.

[26] E.L. Cashin. Kernel korner: Ata over ethernet: putting hard drives on the lan. *Linux Journal*, 2005(134):10, 2005.

[27] The ata-over-ethernet stoarge driver implementation. http://dev.opennebula.org/issues/1648, November 2012.

[28] D.J. Barrett, R.E. Silverman, and R.G. Byrnes. *SSH, the secure shell: the definitive guide*. O'Reilly Media, 2005.

[29] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the linux virtual machine monitor. In *Proceedings of the Linux Symposium*, volume 1, pages 225–230, 2007.

[30] A. Tridgell. Dbench filesystem benchmark, 2010.

[31] Phoronix test suite. http://www.phoronix-test-suite.com, August 2012.

[32] G. Schulz. *Cloud and Virtual Data Storage Networking*. CRC Press, 2011.