Coconet-Procedia

by Agustinus Noertjahyana

Submission date: 09-Jun-2020 12:28PM (UTC+0700)
Submission ID: 1340544248
File name: Publikasi1_01036_6117.pdf (1.05M)
Word count: 4027
Character count: 21728

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Comparative Analysis of NFS and iSCSI Protocol Performance on OpenStack Cinder Technology

Agustinus Noertjahyana, Henry Novianus Palit, Reynaldo Chandra, Justinus Andjarwirawan, Lily Puspa Dewi *

Petra Christian University, Siwalankerto 121-131, Surabaya, 60236, Indonesia

Abstract

Cloud computing is a term used nowadays. Cloud computing usage is already spread all over the world and used by many companies. Because of its high usage, a need for cloud storage now emerges. Cloud storage not only uses resources more effectively but also makes it much easier to utilize a virtual machine. Several alternative protocols can be implemented for cloud storage, and each protocol has its advantages. This research aims to determine the most suitable alternative protocol to implement OpenStack cinder in utilizing cloud storage. Cloud computing implementation is done in a computer laboratory at Petra Christian University using the private cloud. A NAS Synology DS416J was used as the storage provider. The application built by using the OpenStack cloud framework that provides Infrastructure as a Service (IaaS). OpenStack cinder is one of the OpenStack projects that offer cloud storage with persistent storage. OpenStack Cinder itself can be implemented using fiber channel, NFS, and iSCSI protocols, but this research primarily focuses on two protocols, namely NFS and iSCSI. NFS and iSCSI have their respective advantages, so testing is needed to determine the most suitable protocol to be implemented on OpenStack cinder. After implementation, testing was carried out by measuring the performance of the NFS and iSCSI protocols when applied on OpenStack cinder using IO-zone. Based on the results, it can be seen where NFS has the advantage when writing files with a small record size, whereas iSCSI has the advantage when writing files with a large record size; however, in reading activity, there is no noticeable difference between the NFS and iSCSI protocols. By taking into account the results of testing and analysis of the systems that have been made, the conclusion is that the iSCSI protocol is better to be implemented on OpenStack cinder than NFS.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: author@institute.xxx

Keywords: Cloud computing, iSCSI, NFS, OpenStack, OpenStack cinder, Persistent Storage

1. Introduction

In this modern era, technology is developing rapidly. Many individuals or companies that are not making an effort to keep up with this technological development will risk becoming outdated. It is because current technology offers a lot of convenience and efficiency in many ways, both for work and everyday life. One of the techniques provided today is the use of cloud computing.

Cloud computing provides a set of computational resources consisting of applications, computing, storage, networking, development, deployment platforms, and business processes. Cloud computing transforms traditional computing assets into resources that can be used together to expedite a task via computer processes [1].

Cloud itself has different models based on its need. There are two main models in cloud implementation, namely public and private [2]. Public cloud is commonly used by companies or organizations that want to save time because they do not need to manage, maintain and update the data center used for the cloud, while private cloud is chosen because of its security and ease in moving and analyzing data.

To manage this data storage, OpenStack Block Storage (Cinder) is used. OpenStack Cinder is OpenStack's block storage project that can be easily obtained because it is the open-source [3]. It manages data storage by using volumes that are shared with VM instances. Cinder's advantages are the persistent function of block storage and volume snapshots that are provided so that if the VM (virtual machine) is removed, then the data that has been saved will not be lost [4].

Several protocols can be used for OpenStack Cinder, including the Internet Small Computer Systems Interface (iSCSI), Network File System (NFS), and Fiber Channel. The protocols, iSCSI, and NFS, are the focus of this research, to find the most suitable protocol to be implemented on OpenStack Cinder.

2. Theoretical Basis

OpenStack is an open-source cloud platform with an active community; it is highly supported by commercial cloud providers such as Rackspace, Canonical, DreamHost, and HP Cloud; OpenStack is also often used for research [5].

OpenStack involves an open-source tool (known as a project) that uses pools of resources to create and manage private and public clouds. These projects handle the core of cloud computing services, including computing, networking, storage, identity, and image service, that can be used with many optional projects to form a single cloud [6].

OpenStack provides the module software needed to build a cloud platform. Even though OpenStack previously focused on delivering Infrastructure as a Service (IaaS) such as Amazon Web Services (AWS), several new projects have been introduced, which will enable capabilities of being more associated with Platform as a Service (PaaS) [7].

OpenStack project, Keystone, provides client authentication APIs, service discovery, and distributed multi-tenant authorization by implementing the OpenStack Identity API [8]. Keystone is needed as a means of authenticating all OpenStack services.

OpenStack project, Glance, provides a service where users can upload and find images that need to be used for virtual machines. Glance involves discovering, registering, and retrieving the virtual machine image. Glance further has a RESTful API that queries the virtual machine image metadata and first image retrieval [8].

OpenStack project, Nova, provides compute instances. Nova supports the creation of virtual machines, bare-metal servers (if using OpenStack project ironic), and has limited support for system containers. Nova runs as a set daemon on an existing Linux server to provide this service.

OpenStack project, Neutron, provides "network connectivity as a service" between device interfaces managed by other OpenStack services (for example, Nova). It is done by implementing the Neutron API. Neutron prepares a network that will be used by a virtual machine.

OpenStack project, Horizon, is an implementation for the OpenStack dashboard, which provides a web-based user interface for OpenStack services including Nova, Swift, Keystone, etc. [1,3,8]

OpenStack project, Cinder, offers persistent block storage to guest virtual machines (VMs). Block storage is often necessary for expandable file systems and applications that require access to raw block-level storage. Cinder manages the creation of volume, attachment to servers, and detachment from servers. Cinder application programming interface (API) also facilitates snapshot management, which can back up volumes of block storage [4].

Persistent storage is provided on OpenStack workload via a Cinder block storage component. The life cycle of Cinder volumes is maintained independently by computing instances, and volumes can be attached to or detached from compute instances to provide a backing store for file system based storage [7].

Some of the features provided by Cinder are volume management, snapshot management, attach or detach volumes from instances, cloning volumes, creating volumes from snapshots, as well as copying images to another volume and vice versa [9]. Other programs required for the proper functioning of all OpenStack services are NTP, MariaDB, RabbitMQ, and Memcached.

NTP stands for Network Time Protocol (NTP) and is an internet protocol that is used to synchronize the clock of a computer to a time reference. NTP is an internet standard protocol initially developed by Professor David L. Mills of the University of Delaware [10]. NTP is needed as a means of synchronizing time for server and client computers.

MariaDB Server is one of the most popular database servers in the world [11]. MariaDB is created by the MySQL developer and is guaranteed to remain open source. The well-known users are Wikipedia, WordPress.com, and Google. MariaDB is needed to provide a database for OpenStack services.

RabbitMQ is a messaging broker or an intermediary for messaging [12]. It gives the application a platform for sending and receiving messages and provides a place for the message until the message has been received. RabbitMQ is used by OpenStack services to communicate with each other.

Memcached is an open-source, high performance distributed memory object caching system, intended to speed up dynamic web applications by reducing database load. Memcached is an in-memory key-value that stores data (strings, objects), which is the result of database calls, API calls, or page rendering [13].

2.1. Network File System

Network file system (NFS) is network abstraction over a file system that allows a remote client to access it through a network using a method similar to a local file system. Although not a system that first appeared, NFS continues to evolve until it became widely used as a network file system on UNIX. NFS allows the sharing of a file system, which is common among many users and provides the advantage of data centralizing to minimize the storage needed [14].

2.2. iSCSI (Internet Small Computer System Interface)

Internet Small Computer Systems Interface (iSCSI) is a TCP / IP based protocol for sending SCSI commands over IP based networks. The iSCSI structure can continue to be developed over local LANs and used on WANs or even the internet. Internet SCSI treats external volume as an internal hard disk [15].

2.3. Previous Work

Previous research discussed the implementation of OpenStack Swift to optimize the use of Storage Resources [2]. The storage media used is the remaining storage in each client. However, there is a slight problem when there is a damaged client. The system will be a little disturbed, even though it will quickly recover. In the present study, OpenStack will be implemented but using NAS (Network Access Storage) as a centralized storage media, using OpenStack Cinder as a platform.

2.4. Similar Research

To understand what must be done when testing the NFS and iSCSI protocols, a review of the study in Hashimoto's research entitled Comparing Filesystem Performance in Virtual Machines was conducted [16]. Testing was done using

the IOzone benchmark filesystem. According to Hashimoto's research, NFS read performance is very good for files with small record sizes. Hashimoto suspected that NFS was doing a read ahead and using the cache to get the best performance. As for the reason the virtual machine can perform better than its physical host, Hashimoto suspects that the hypervisor has buffered it to read from the virtual machine. In the test results for write, it is concluded that the NFS performance was terrible due to network limitations.

3. Research Methodology

In this research, the steps taken are as follows:

Install and configure OpenStack on the controller node, Installing and configuring OpenStack on the compute node. Configuring Network Attached Storage (NAS) as part of the NFS and iSCSI protocol implementation and NFS and iSCSI protocol configuration on the controller node.

After the preparation of the controller node, compute node, and storage, the system is tested using the IOzone tool [17–19]. Then the results of the IOzone test will be compared to find the most optimal protocol performance.

3.1. System Design

The system design consists of a controller node and four computers as compute nodes. The controller node that acts as the OpenStack server uses the OpenStack projects Keystone (identity), Glance (image), Nova (compute), Neutron (network), Horizon (dashboard) and Cinder (block storage). On the compute node, the OpenStack project Nova is used.

To store the virtual machine (VM) data that is run on the compute node and use the data again, OpenStack project Cinder is needed because of persistent storage it provides. The data is stored on external storage, namely Network Attached Storage. For this research, the controller node is made so that it can only be accessed through the local network. The system and network design can be seen in Fig. 1.

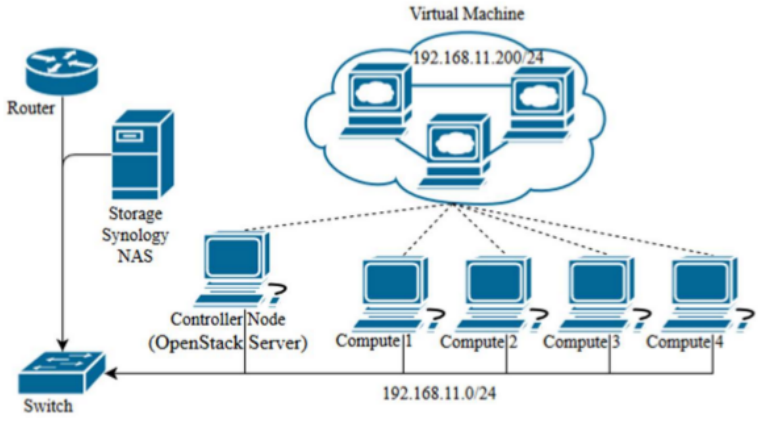

Fig. 1. System and Network Design

3.1.1. Controller Node

The first step is to install an operating system on the controller node. The operating system used is Ubuntu server 16.04 LTS 64 bit. Then the controller node is connected with the Pike version of OpenStack projects, namely the OpenStack project Keystone, Glance, Nova, Neutron, Horizon, and Cinder. The controller node is also installed with MariaDB, RabbitMQ, and Memcached as the pre-requirements of OpenStack version Pike. Besides, the Network Time Protocol (NTP) also needs to be installed as a means of synchronizing time. NTP needs to be installed on the

controller node. It will provide a time synchronization tool for all compute nodes that exist on the system so that all nodes in the network report at the same time.

It is mandatory to install a database to store data for all services that are run by OpenStack. The MariaDB installation needs to be done on the controller node. MariaDB itself is an open source-based relational database management system (RDBMS) that is recommended when using OpenStack.

RabbitMQ installation needs to be done on the controller node. RabbitMQ provides a means for all OpenStack service components to communicate with each other. All OpenStack services running on RabbitMQ will act as a guest. Memcached installation is done on the controller node. Memcached functions to speed up access to external data (e.g., databases and APIs) by using a distributed memory object caching system.

A keystone installation is performed on the controller node. Keystone serves to store user, service, and endpoint data of all other OpenStack projects and also as an authenticator for all OpenStack services. Besides, Keystone acts as a user with administrative access rights and also has an administrator role. All other OpenStack services must have a user, service, and endpoint registered with Keystone so that the service can be used.

Glance installation is performed on the controller node. Glance functions to provide a particular directory for storing images that are used for the virtual machine. Glance also requires a database in MariaDB along with an account with admin rights on that database.

Nova installation is done on the controller node. Nova functions to provide a compute instance. Nova also needs a database in MariaDB along with an account with admin rights on the database. Neutron installation is performed on the controller node. Neutrons function to prepare virtual networks for virtual machines that will be run later.

Horizon installation is done on the controller node. Horizon serves to provide a web-based user interface for OpenStack services, especially Nova, including selecting flavors and running virtual machines. Cinder installation is done on the controller node. Cinder serves to provide volume services to end-users, namely virtual machines, through OpenStack Nova.

3.1.2. Compute Node

The first step is to install an operating system on the compute node. The operating system used is Ubuntu server 16.04 LTS 64 bit. Compute nodes use four computers in a multimedia laboratory, namely compute1, compute2, compute3, and compute4. Next, the compute node is installed with the Pike version of OpenStack, which is needed to do computing, namely the OpenStack project Nova. Network Time Protocol (NTP) is also established as a means of synchronizing time. NTP needs to be installed on the compute node. It will synchronize the compute node time with the controller node time, so there is no problem regarding timestamp difference. Nova installation is performed on the compute node. Nova on the compute node functions so that users can launch virtual machines using the compute service provided.

3.2. Network Design

For the preparation of this research, a router is used and has a network IP address of 192.168.11.0, subnet mask 255.255.255.0, default gateway 192.168.11.1, DNS server 203.189.120.4; it does not require a proxy to access the internet. The plan is for the network to be used for server and client computers in the multimedia laboratory on 192.168.11.0/24 and the network will be used for VM instances using the IP range 192.168.11.200/24 - 192.168.11.254/24 through a multimedia laboratory router so that the network made for this study will not collide with the existing network.

The computers are arranged using static IP with the first computer starting at number 6 using 192.168.11.6. The second computer is number 7, using 192.168.11.7 and so on. The network design that has been planned to be used by the controller node compute node, NAS Synology virtual machine and storage can be seen in Fig. 1.

3.3. Testing Design

Test using the IOzone tool to find out what protocols are more optimal for use on OpenStack Cinder. The parameters to be tested are read performance (bandwidth) and write performance (bandwidth).

The selection of RAID that will be used based on the results of reading and write performance tests that have been done will be selected based on the best performance results so that the analyzed data flow is not affected by other factors.

4. Result and Discussion

This test is done to compare the performance of the NFS and iSCSI protocols. All testing of the NFS and iSCSI protocols is done on the virtual machine. In Fig. 2 and 3, there is a graph of testing results from NFS, namely, write and read performance using the IOzone benchmark.

In Fig. 2, it can be seen where the throughput speed drops dramatically when the file size is 524288 Kbyte and continues to decrease, this may be caused by the limitations of the ext4 filesystem and constraints on the processor used by the virtual machine.

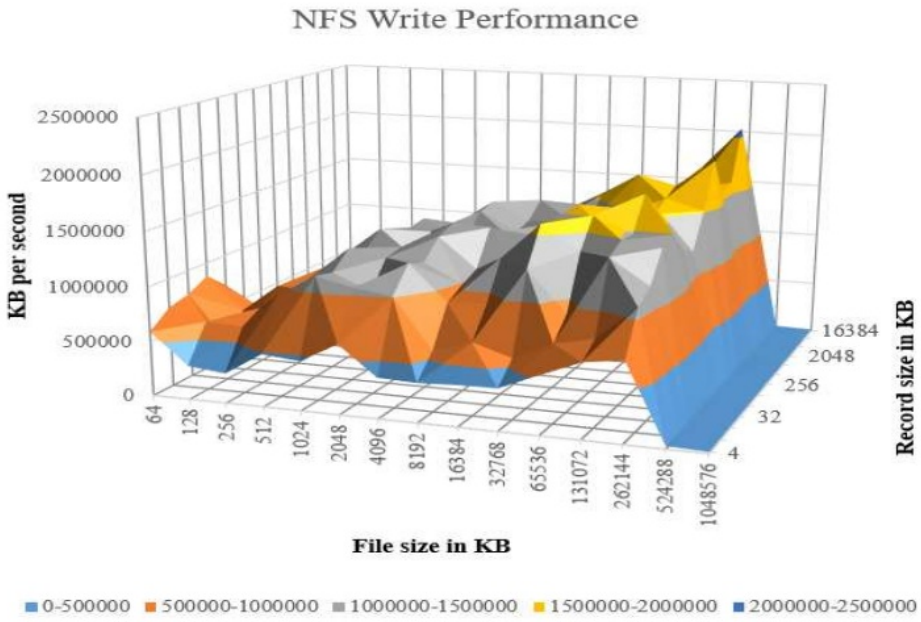

Fig. 2. NFS Write Performance

In Fig. 3, it can be seen where the read throughput in the virtual machine is higher than that done directly on the physical computer. It might occur because of the buffering activity carried out by the hypervisor.

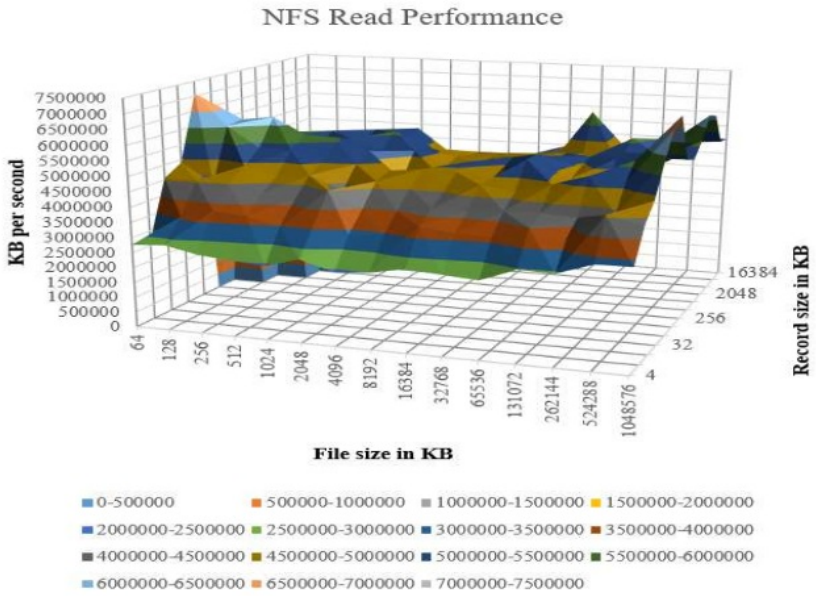

Fig. 3. NDS Read Performance

Furthermore, in Fig. 4 and 5, there is a graph of testing results from iSCSI, namely, write and read performance using the IOzone benchmark.

In Fig. 4. it can be seen where the throughput speed drops dramatically when the file size is 524288 Kbyte and continues to decrease; this may be due to the limitations of the ext4 filesystem and constraints on the processor used by the virtual machine.

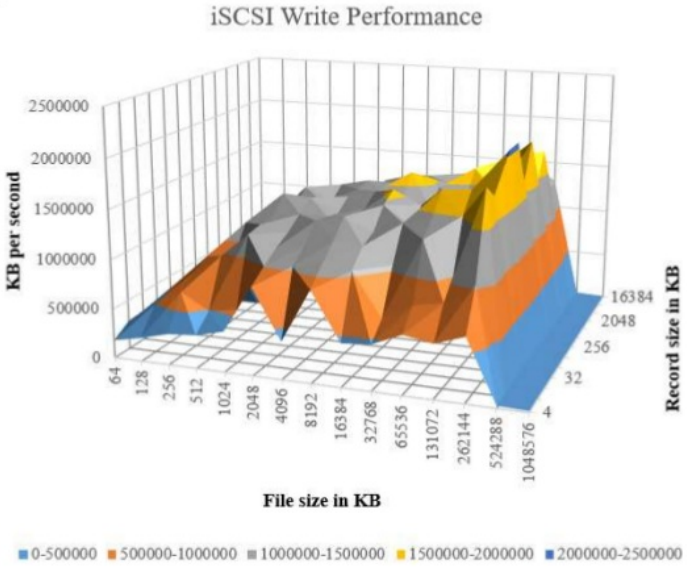

Fig. 4. iSCSI Write Performance

In Fig. 5, it can be seen where the read throughput in the virtual machine is higher than that done directly on the physical computer. It might occur because of the buffering activity carried out by the hypervisor.

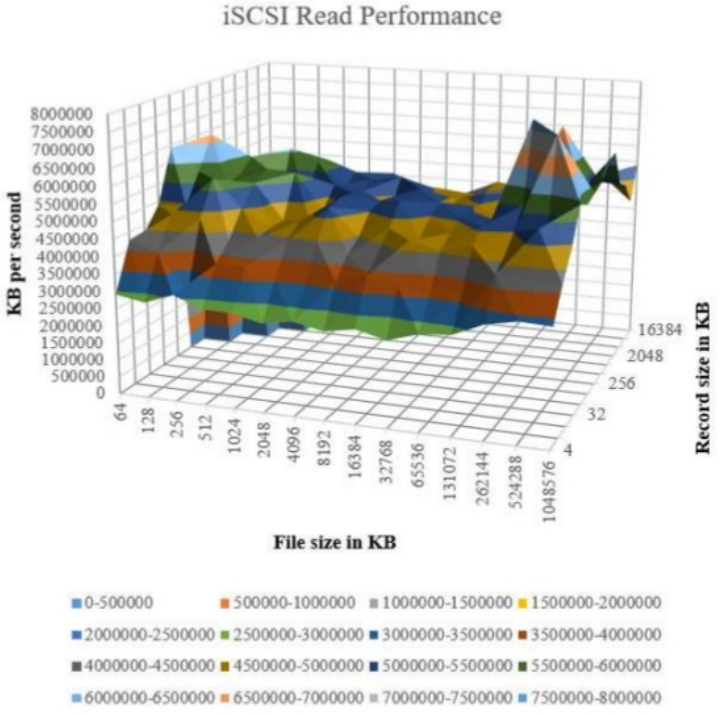

Fig. 5. iSCSI Read Performance

Based on the throughput generated by the NFS and iSCSI protocols, the iSCSI protocol has better throughput than the NFS protocol, both from performance when doing write and read activities. Therefore, it is concluded that iSCSI is a better protocol to implement on Open-Stack cinder than NFS because the iSCSI protocol is faster than NFS.

5. Conclusion

From the research, outcomes can be as follows:

NFS has the advantage when writing files that have a small record size, while iSCSI has the power when writing files that have a significant record size. In reading, there is no noticeable difference between NFS and iSCSI. Overall, the process of both write and read iSCSI has a small advantage in its throughput, so that iSCSI is a better protocol to implement on OpenStack Cinder than NFS.

The best configuration to implement is to use a configuration file that has been provided by OpenStack. NFS and iSCSI protocol configurations on the client-side can be done directly on the Cinder configuration file, so there is no need to configure the computer manually. The setting of the NFS and iSCSI protocols on the server-side is done directly on Synology NAS storage because the Synology NAS system can accept NFS configuration and make iSCSI LUN and iSCSI its targets.

Acknowledgments

This research project was sponsored by the Directorate General of Higher Education Indonesia on the scheme "Penelitian Terapan Unggulan Perguruan Tinggi" in 2019. Many thanks to the Center of Research, Petra Christian University, Surabaya, Indonesia, for the great excellent piece.

References

1. Li Z, Li H, Wang X, Li K. A generic cloud platform for engineering optimization based on OpenStack. Adv Eng Softw [Internet]. 2014;75:42–57. Available from: http://www.sciencedirect.com/science/article/pii/S0965997814000775
2. Noertjahyana A, Reno J, Palit HN, Andjarwirawan J. Private cloud storage implementation using OpenStack Swift. TELKOMNIKA (Telecommunication Comput Electron Control. 2019;17(1):218.
3. OpenStack.com. OpenStack Operations Guide. OpenStack. 2014;
4. Couto RS, Sadok H, Cruz P, da Silva FF, Sciammarella T, Campista MEM, et al. Building an IaaS cloud with droplets: a collaborative experience with OpenStack. J Netw Comput Appl [Internet]. 2018;117:59–71. Available from: https://doi.org/10.1016/j.jnca.2018.05.016
5. Pacevič R, Kačeniauskas A. The development of VisLT visualization service in OpenStack cloud infrastructure. Adv Eng Softw. 2017;103:46–56.
6. Costache S, Dib D, Parlavantzas N, Morin C. Resource management in cloud platform as service systems: Analysis and opportunities. J Syst Softw [Internet]. 2017;132:98–118. Available from: http://dx.doi.org/10.1016/j.jss.2017.05.035
7. Silverman B, Solberg M. OpenStack for Architects. Packt Publishing Ltd; 2017.
8. Litvinski O, Gherbi A. Experimental Evaluation of OpenStack Compute Scheduler. Procedia Comput Sci [Internet]. 2013;19:116–23. Available from: http://www.sciencedirect.com/science/article/pii/S1877050913006285
9. Khedher O. Mastering OpenStack : discover your complete guide to designing, deploying, and managing OpenStack-based clouds in mid-to-large IT infrastructures with best practices, expert understanding, and more [Internet]. [cited 2019 Jun 20]. Available from: https://www.packtpub.com/virtualization-and-cloud/mastering-openstack-second-edition
10. Huang L, Sezaki K. Adjustment on end-to-end delay to remove the distortion caused by NTP clock adjustment. 2010;1075–86.
11. About MariaDB - MariaDB.org [Internet]. [cited 2019 Jun 20]. Available from: https://mariadb.org/about/
12. What can RabbitMQ do for you? — RabbitMQ [Internet]. [cited 2019 Jun 20]. Available from: https://www.rabbitmq.com/features.html
13. Berezecki M, Frachtenberg E, Paleczny M, Steele K. Power and performance evaluation of Memcached on the TILEPro64 architecture. Sustain Comput Informatics Syst [Internet]. 2012;2(2):81–90. Available from: http://dx.doi.org/10.1016/j.suscom.2012.01.006
14. Ou Z, Song M, Hwang ZH, Ylä-Jääski A, Wang R, Cui Y, et al. Is cloud storage ready? Performance comparison of representative IP-based storage systems. J Syst Softw. 2018;138:206–21.
15. Entezari-Maleki R, Sousa L, Movaghar A. Performance and power modeling and evaluation of virtualized servers in IaaS clouds. Inf Sci (Ny) [Internet]. 2017;394:106–22. Available from: http://www.sciencedirect.com/science/article/pii/S0020025517305145
16. Comparing Filesystem Performance in Virtual Machines [Internet]. [cited 2019 Jun 20]. Available from: http://mitchellh.com/comparing-filesystem-performance-in-virtual-machines
17. Zhang Z, Feng D, Tan Z, Yang LT, Zheng J. A light-weight log-based hybrid storage system. J Parallel Distrib Comput [Internet]. 2018 Aug 1 [cited 2019 Jun 20];118:307–15. Available from: https://www.sciencedirect.com/science/article/pii/S0743731517303404
18. Medel V, Tolosana-Calasanz R, Bañares JÁ, Arronategui U, Rana OF. Characterizing resource management performance in Kubernetes. Comput Electr Eng [Internet]. 2018 May 1 [cited 2019 Jun 20];68:286–97. Available from: https://www.sciencedirect.com/science/article/pii/S0045790617315240
19. . He X, Zhang M, (Ken) Yang Q. STICS: SCSI-to-IP cache for storage area networks. J Parallel Distrib Comput [Internet]. 2004 Sep 1 [cited 2019 Jun 20];64(9):1069–85. Available from: https://www.sciencedirect.com/science/article/pii/S0743731504000711

Coconet-Procedia

ORIGINALITY REPORT

17%	12%	12%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

7%

★ Submitted to Amrita Vishwa Vidyapeetham

Student Paper

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		