Analysis of object-based storage implementation on storage data center

by Bekti Susanto

Analysis of object-based storage implementation on storage data center

Agus Hariyanto, Bekti Maryuni Susanto

Departement of Computer Engineering, State Polytechnic of Jember

agus_hariyanto@polije.ac.id, bekti@polije.ac.id

Abstract. The data center is the backbone of various services offered by internet giant companies like Google, Amazon, and others. Increasingly complex, varied, and the rapid growth of services offered, the data center will continue to grow and develop which makes data center management and planning more challenging. One of the most important parts of data centers that require good planning is network storage. This network storage determines how data is stored in the data center so that it can be accessed at any time by service users effectively. Network storage is a term used to refer to a network-based storage. Based on the results of a survey conducted in Australia, human error was the biggest cause of sudden data center malfunction with a percentage of 28%. The next biggest causes are electrical problems, hardware and software which each have a percentage of 25%, 21% and 11%. While natural disasters such as earthquakes only have a percentage of 8%. This study applies object based storage to data center storage. The storage object used is openstack object storage. Measurement of network performance using COSbench software. Measurement of network performance is carried out under normal conditions which include read, write and delete activities. The file size operated when measuring network performance is divided into four clusters, tiny, small, medium and large.

1. Introduction

The data center is the backbone of various services offered by internet giant companies like Google, Amazon, and others. Increasingly complex, varied, and the rapid growth of services offered, the data center will continue to grow and develop which makes data center management and planning more challenging. One of the most important parts of data centers that require good planning is network storage. This network storage determines how data is stored in the data center so that it can be accessed at any time by service users effectively. Network storage makes it easy to store data with certain methods so that users can use it in a network. Network storage is a term used to refer to a network-based storage. There are many technologies that can be used in building network-based storage such as Network Attached Storage (NAS), Direct Attach Storage (DAS), and Storage Area Networks (SAN) [1].

Stored data is an important asset for each user so data must be available when needed by anyone who wants to access it and can be accessed every second every day. If the data center stops operating because certain things will have an impact on the availability of data until the data is damaged. This will have fatal consequences on data that are sensitive and must be available 24 hours such as bank data, finance, government, insurance, hospitals, and others.

Based on the results of a survey conducted in Australia, human error is the biggest cause of sudden data center malfunction with a percentage of 28%. The next biggest causes are electrical problems, hardware and software which each have a percentage of 25%, 21% and 11%. While natural disasters such as earthquakes only have a percentage of 8%. This shows that data center malfunctions caused by natural disasters are very unlikely. In most cases, to be able to return to operations after a disaster, a data center requires an average time of between 6-20 hours.

OpenStack Object Storage or also known as Swift is one of the services on openstack with a scalability level and a very high level of storage redundancy in hardware. With openstack object storage, users can store many objects of infinite size as limited by the capacity of the storage media [2].

To accommodate the number of objects stored by the administrator can also increase the storage capacity according to the desired without any restrictions. The high level of redundancy of openstack object storage is suitable for data archiving purposes such as log data and also for storing virtual mechine templates used by openstack compute.

Swift can be used in a variety of needs, including can be used for mobile or web applications, backup, and archiving. The storage system provided by Swift can be accessed via the HTTP protocol, text-based applications (cli), filesystem interface, or can also be used easily by applications to store and synchronize data with its desktops, tablets and mobile devices.

Swift is an object-based storage system, this is certainly different from storage media with SAN systems that are "block" based and NAS based on "files". But with this object-based storage system that makes Swift has a high level of data availability, redundancy, throughput, and capacity. The level of writing large and concurrent amounts of data can occur quickly, as well as the level of reading. This shows that swift can be used to build centralized storage media with a very high level of data traffic and the number of connections that occur simultaneously.

This study applies object based storage to data center storage. The storage object used is openstack object storage. Measurement of network performance using COSbench software. Measurement of network performance is carried out under normal conditions which include read, write and delete activities. The file size operated when measuring network performance is divided into four clusters, tiny, small, medium and large.

2. Method

Swift data (account, container, and object) is a resource that is stored in the physical hardware at the end. Node is a machine that runs the process of swift. Swift clusters are a collection of nodes that run the whole of the swift processes and services needed to act as an object-based distributed storage system. To ensure durability and prevent the occurrence of storage system failures, the accumulation of nodes must be arranged in such a way into a cluster consisting of Region and Zone [3].

2.1. Region

By using swift, a collection of several nodes can be made into a cluster which is then separated by region. Region can be defined as the boundary or coverage of a region or geography. For example, a region might be a number of racks consisting of many nodes placed in the highest latency or locations that are far apart from other nodes. A cluster must have at least one region, making it possible to have multiple clusters with the same region as all nodes belonging to the same region. A cluster can use two or more clusters called multi-region clusters (MRC). If a cluster has more than one region (MRC), when there is a read / write process, it will take precedence over the closest region by choosing the lowest latency. In the write process, the data will be stored in the nearest region first and then proceed with the replication process. This will speed up the data write process carried out by service users.

2.2. Zone

In Region, Swift has a zone that has been configured to prevent storage failures. A zone must have hardware that is different from the other zones. In large scale applications, a zone must be separated by data center, as separated by firewalls, and different service providers. If you only have one data center, zones can be separated by rack.

2.3. Node

Nodes are physical servers that run one or more Swift server services [4]. Swift's main service is proxy, account, container, and object. A node that runs an account or container service will store data and metadata from the account and container. A set of nodes that run all the services needed by swift to become distributed storage systems are called clusters. Openstack object storage implementation show in Illustration 1.

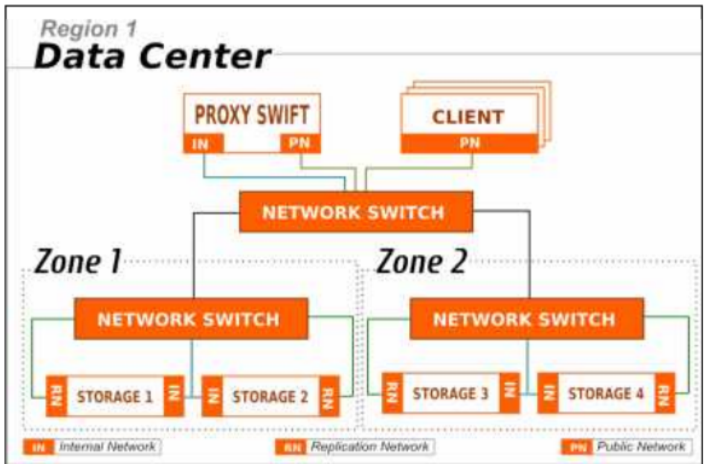
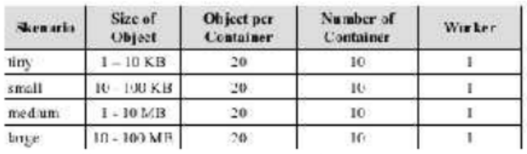


Illustration 1: Openstack Object Storage Implementation

Benchmarking will be divided into several scenarios, namely tiny, small, medium, and large. Any details of each scenario are shown in Illustration 2.



Skenario	Size of Object	Object per Container	Number of Container	Worker
tiny	1 – 10 KB	20	10	1
small	10 - 100 KB	20	10	1
medium	1 - 10 MB	20	10	1
large	10 - 100 MB	20	10	1

Illustration 2: Bencmarking Scenario

3. Result and Discussion

Based on the network topology built, there are three networks with different subnets, namely public networks, internal / data networks, and replication networks. The public network will have a network address of 192.168.0.0/24 which is a network that connects the client to a proxy server. The internal / data network will have a network address of 10.10.0.0/24 which is a communication network between the proxy server and the storage node in retrieving and entering data (GET / PUT). While the replication network will be in 10.10.1.0/24 which is a replication network between storage nodes.

Swift-Ui is a web-based interface that can be used in accessing storage. The previous staticweb module must be activated first so that the container can be accessed as previously mentioned in the proxy-server.conf configuration. The config.js configuration file on swift-ui must be adjusted first before swift-ui is downloaded to the container. Things that need to be adjusted are username, key, and endpoint. Set index.html as the web index container so that when the client opens the web-client container it will be directed to index.html. Use the browser to access storage with a web interface.

Benchmarking will be carried out with several parameters and scenarios. The tool that will be done in benchmarking is Cloud Object Storage Benchmarking (COSBench) developed by Intel [5]. This tool can be downloaded for free (Open Source) from gihub. The performance of swift storage will be measured based on the performance metrics that are available in this tool. This tool will be installed on the PC client.

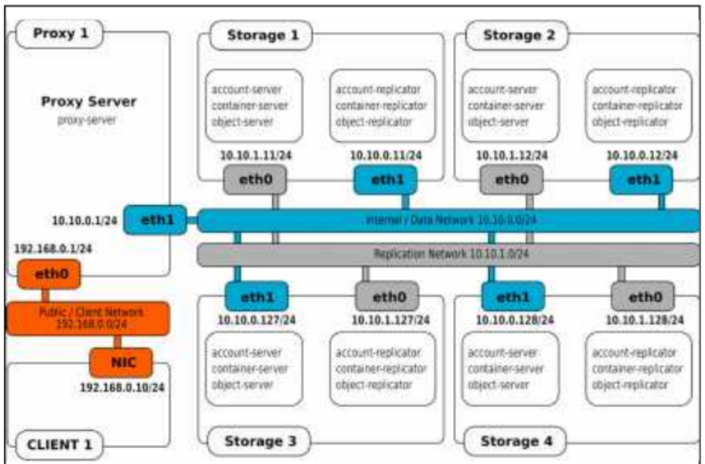


Illustration 3: Network Topology

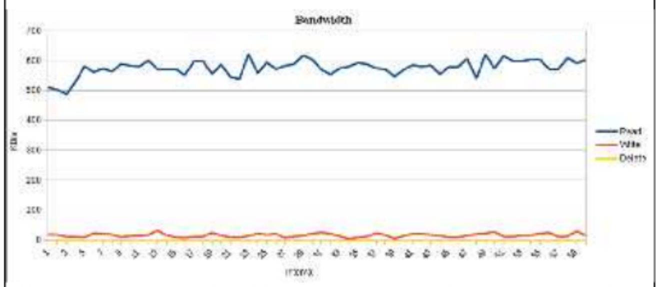


Illustration 4: Bandwidth Measurement of Tiny Bencmarking

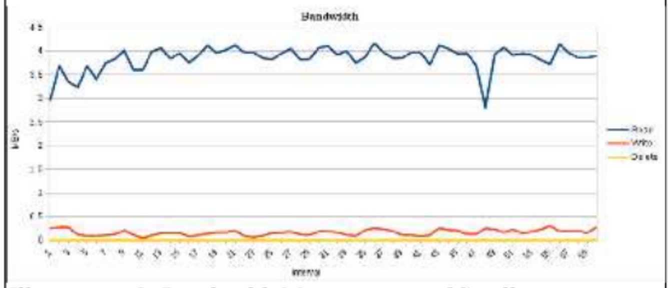


Illustration 5: Bandwidth Measurement of Small Benchmarking

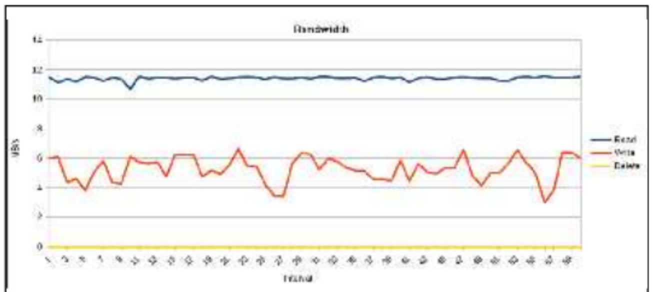


Illustration 6: Bandwidth Measurement of Medium Bencmarking

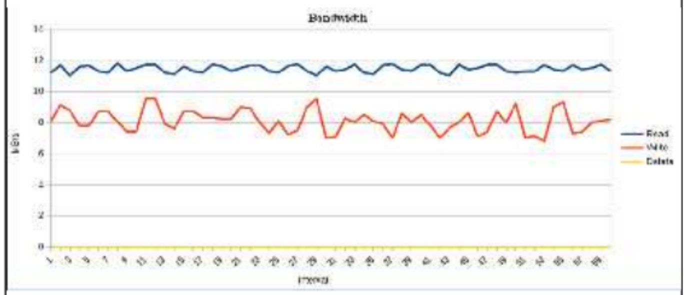


Illustration 7: Bandwidth Measurement of Large Bencmarking

4. Conclussion

Based on the experimental results, object storage is able to maintain data replication in the data center. Test results showed tiny, small, medium and large file operations were successful overall with different bandwidths. The larger the file being operated requires greater bandwidth. Data replication in the datacenter functions as disaster recovery when an error occurs in the system that will be discussed in subsequent studies.

Acknowledgement

Our thanks goes to Departement of Information Technology, State Polytechnic of Jember who has helped support for this research.

Analysis of object-based storage implementation on storage data center

ORIGINALITY REPORT

4%	%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Lambung Mangkurat University Student Paper	2%
2	Submitted to British Institute of Technology and E-commerce Student Paper	1%
3	Jannatun Noor, Saiful Islam Salim, A.B.M. Alim Al Islam. "Strategizing secured image storing and efficient image retrieval through a new cloud framework", Journal of Network and Computer Applications, 2021 Publication	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		