Object Storage Performance Analyzing System Based on Packet Transfers and Method Calls Visualization

Shunpei Hayakawa†
Electrical Eng. and Electronics
Kogakuin Univ. Graduate School
Tokyo, Japan
cm18030@ns.kogakuin.ac.jp

Saneyasu Yamaguchi
Dept. of Info and Comm Eng.
Kogakuin Univeristy
Tokyo, Japan
sane@cc.kogakuin.ac.jp

1 INTRODUCTION

An object storage system is a newly emerging and highly scalable storage system that manages data as an object. Since it is a complex disturbed system composed of several server computers, to identify the cause of performance degradation is usually difficult. An object storage system handles data access using HTTP. In this paper, we propose a system for analyzing the performance of OpenStack Swift [1], which is one of the most popular object storage systems, by visualizing packet transmissions and method calls. We then show the result of analyzing and its effectiveness.

2 ANALYZING METHOD

Here, we propose a system for monitoring the behavior of Swift object storage. The system records the outputs and inputs of all the packet at all the machines in the object storage system. The inputs and outputs of packets can be captured in the Ethernet layer in the usual operating systems with the administrator authority. The bottleneck process, in which a long time is consumed, is clearly highlighted by visualizing these inputs and outputs.

After identifying the bottleneck machine that consumes a long time, the proposed system monitors the begins and ends of the function/method calls in the bottleneck machine. This visualization identifies the method that consumes a long time.

3 VISUALIZATION

In this section, we practically analyze a Swift object storage system. It is implemented in Python language. Thus, the proposed system monitors the method calls of Python language.

The experimental object storage system was composed of four physical machines: Client, Proxy Node, Auth Node, and Storage Node. The SwiftClient process is invoked on Client. The ProxyServer service is running on Proxy Node. Identity service is running on Auth Node. The Account, Container, and Object services are running on Storage Node. We repeated to upload an object the size which is 1MB and then identified the bottleneck method by the proposed analyzing system.

Figure 1 shows the visualized system behavior of one upload. The horizontal axis represents time (unit: sec). The vertical axis represents the processes in servers. The vertical lines represent packet transfers. From the figure, we can see that processes on the Identify Server took a long time in an upload.

We investigated insight into the methods invoked in Identity Server. The gray line segments in the figure show calls of methods in Python on the server. This figure shows that the most time-consuming calls are for `authenticate` and `initialize_application`. The former method is for authentication using passwords. The latter is for the initialization of the application. This latter method was called in the first upload. This was not called in the following uploads.

From this visualized figure, we can conclude that the proposed system can identify and highlight the time-consuming methods.

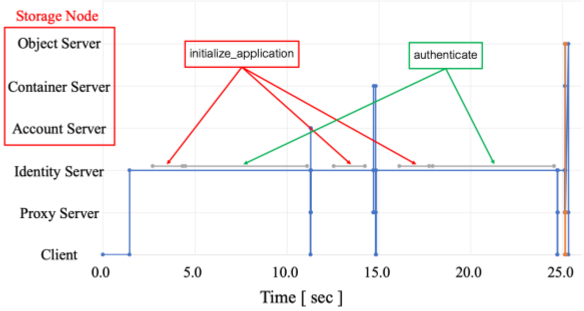

Figure 1: Visualized object storage behavior

3 CONCLUSION

In this paper, we proposed an object storage analyzing system based on visualization of packet transmissions and method calls. We presented a visualized behavior and showed that the proposed system can identify the bottleneck methods.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers 17K00109, 18K11277. This work was supported by JST CREST Grant Number JPMJCR1503, Japan.

REFERENCES

[1] Sridevi Bonthu, Y S S R Murthy, M. Srilakshmi “Building an Object Cloud Storage Service System using OpenStack Swift”, International Journal of