Evaluating persistent Memory, DRAM hybrid main memory Key-value-store server

WAI TING CHAN[1] MIDORI SUGAYA[1]

Abstract: Recently, IoT technology is becoming more popular and providing service via cloud server is becoming more common. As the use of cloud server by IoT devices is increased, the performance of server tends to decrease. To solve this problem, we focus on edge computing technology, fogcached, that is a KVS server system with expanding memcached using persistent memory. Since there are some unclear points of fogcached, we evaluated its performance. Our evaluation result shows that, with the same memory amount of DRAM and Optane DCPM, fogcached has better performance than memcached.

Keywords: Persistent memory, Key-Value store, Edge computing, Fogcached

1. Introduction

In the recent years, as IoT technologies is becoming more popular, cloud server with rich resource that provides services is becoming more common. However, cloud server is usually far away from the end user device, which lead to the problem of low responsiveness of server. In order to solve this problem, edge server, which is placed between IoT node and cloud server, has been proposed. Although memcached, which is one of the typical Key-Value-Store cached server system, can improve the performance in term of responsiveness [1], reliability has not been considered. To solve this problem, Kouki et al. [2] suggested fogcached, the DRAM/Persistent Memory hybrid memory server for edge computing which expands memcached algorithm with the Intel Optane persistent memory. It has larger capacity than DRAM and can store data in the memory even when powering off, which increases the reliability of the system [3].

Though the fogcached has been proposed, its Dual-LRU algorithm's evaluation is not sufficient. The data size moving from DRAM to persistent memory is not clear, and thus, the reliability cannot be guaranteed. In order to verify the reliability of fogcached, a new approach of evaluation is needed. In this research, we aim at clarifying the unclear point and improve the performance of fogcached. Firstly, we performed Kouki et al.'s evaluation system, and confirmed that fogcached has better performance than memcached. However, there are several issues related to main functions of fogcached that affect the performance but have not been evaluated such as how moving item between DRAM and persistent memory affects the performance. Therefore, we improved Kouki et al.'s evaluation system and perform the evaluation including the analysis of the data movement between two memory devices.

2. Proposal

2.1 Overview

Even though Kouki et al.'s evaluation system shows that fogcached's overall performance is better than the memcached, for further improvement, it is necessary to evaluate how algorithm affects the performance since the biggest different between fogcached and memcached is the algorithm. Also, to perform accurate evaluation, the detail of the system must be defined, such as network configuration, latency between client and server, etc.

2.2 Evaluation model

In order to analyze how data move between DRAM and persistent memory, we designed the evaluation model (Fig. 1).

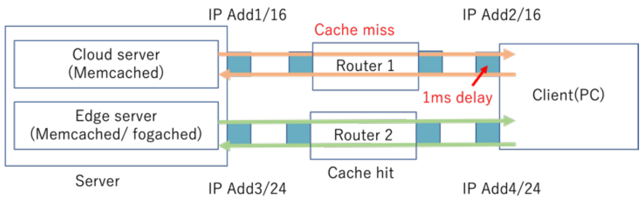

Fig.1 Evaluation model

In order to simulate the latency due to the physical distance of cloud server, we generated 1ms latency to the outgoing packet sent from client, which is not specified by the prior evaluation, by using a network traffic controller, Network Emulator (NetEm).

Before client starts accessing the server and measures the latency, the server first needs to warm up with the dataset to be used during the evaluation.

3. Evaluation

3.1 Evaluation system

We prepared the following components for the evaluation system:

- A server: 16GB×12 and 126GB×12 Intel Optane memory, ubuntu 18.04 OS
- A client computer: 2 network interface cards, 8GB×12 DRAM ubuntu 20.04 OS

For the network setting, we created two independent LAN with different netmasks instead of connecting to the internet to ensure the data flow with the correct route.

In the evaluation, we created both edge server and cloud server on the same machine, which possess 256MB DRAM+ 1GB/2GB DCPM and 100GB DRAM memory respectively. The total data

1 Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo 135-8548, Japan

size of the dataset is 50GB. All the data was stored in the cloud server and only part of the data as cache was stored in the edge server.

Client-side computer requested for items from the server for a million times, with the data size of 1MB per item. We performed the experiment for 5 times, in which only the memory configuration of the edge server was changed while the cloud server's setting was the same.

3.2 Latency

From the evaluation, we obtained the following result.

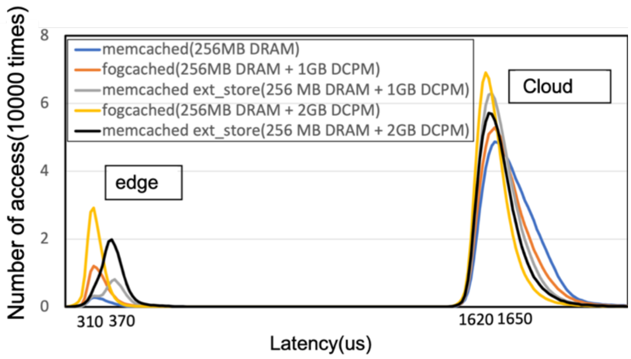

Fig.2 Edge-Cloud latency

As shown in Fig. 2, we noticed two points from the graph. Firstly, on the edge side, when the memory configuration is the same, fogcached has lower average latency than memcached. Secondly, the increase in memory size of Intel Optane memory (DCPM) did not only lead to the increase in hit rate on edge side, but also the decrease in average latency on the cloud side.

3.3 Data flow between DRAM and DCPM

The data shows that after a certain period, all DRAM's data moved to DCPM and DCPM's data did not move to DRAM.

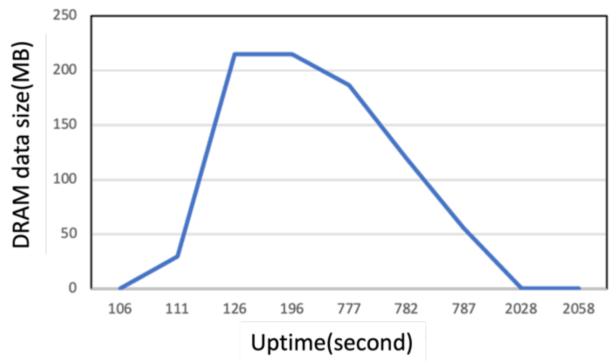

Fig.3 Fogcached edge server's DRAM data usage

The whole evaluation lasted for around 2000 seconds. From Fig. 3, data size of DRAM reached the peak of around 210MB at the uptime of 126 seconds and started to decrease from the uptime of 777 seconds. Finally, after 2028 seconds, the data size of DRAM became 0, which indicates that there was no data loaded on DRAM. By investigating the movement of item, we found that the data size of DCPM increased at the same time when DRAM data size decreased. However, we expected that the data size of DRAM does not become 0MB once the data is set on DRAM. According to the algorithm of fogcached, the items discarded by DRAM will be moved to DCPM. Therefore, our assumption is that the issue is related to Time to Live (TTL) of all items. Based on the TTL principle, when the designated time past, the data will be discarded. As certain time has passed, TTL count of all items becomes 0, which means that the timeout is reached and all items are discarded by the DRAM and moved to DCPM.

3.4 Usage of DRAM, DCPM

Fig. 4 shows the hit rate of DRAM and DCPM. According to the graph, the total number of cache hit is equal to the number of caches hit of DCPM and no DRAM data get hit, indicating that the DRAM was not used during the evaluation.

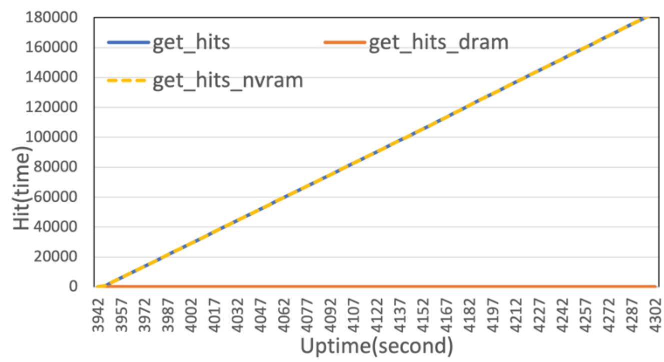

Fig.4 Cache hit on fogcached server

4. Conclusion

The hybrid memory cache server system, fogcached, has a better performance than existing cache server system, memcached using DCPM. However, in our evaluation system, fogcached did not use DRAM, but only used DCPM to execute the evaluation program because all DRAM data was discarded due to the TTL mechanism, which means that there are large chance that fogcached can obtain a better result.

Our research goal is to examine how data movement between DRAM and DCPM affect the performance of fogcached. However, in our evaluation system, DCPM's data did not move to DRAM, indicating that our evaluation system did not reach the goal. The reason is that none of the item reached the condition of promotion. The same items on DCPM was accessed by client side 7 times in a short period and transfererd the items to DRAM. Although the result did not meet our expectation, it reveals certain suggestion to improve the evaluation such as extending the TTL duration of all items. By doing so, we are likely to avoid the dataset on DRAM being moved to DCPM due to timeout. Our future work will evaluate fogcached by implementing it to application.

References

[1]Amazon Web Service, Inc."Memcached" https://aws.amazon.com/jp/memcached

[2]Kouki et al., "fogcached: DRAM-NVM Hybrid Memory-Based KVS Server for Edge Computing", Edge Computing –EDGE 2020 –4th International Conference, Lecture Notes in Computer Science 12407, Springer, 2020, pp.50-62

[3]Intel Optane DC Persistent Memory https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html