Performance research and prospect exploration of distributed object storage of meteorological data in domestic platform

Kexin Cheng

1 School of Software Engineering, Chengdu University of Information Technology, Chengdu, Sichuan, 610225, China

*Corresponding author’s e-mail: 3180702002@cuit.edu.cn

Abstract. Meteorological data are important information data in China, and have made tremendous contributions to disaster prevention and disaster reduction in China. However, the current meteorological system has a long-term dependence on foreign hardware and software platforms, showing obvious security problems. It is an urgent task to complete the domestication and performance improvement of the meteorological system. At the same time, as the black box architecture of flash memory devices greatly hinders the co-optimization of software and hardware, open-channel flash memory has begun to attract attention as a new type of flash memory architecture. In this article, we have adopted new excellent domestic products including Phytium chips, Starblaze open-channel flash memory devices, and Kylin operating systems as components of a new domestic software and hardware platform. The object storage system of meteorological data has been implemented in a real environment by Ceph. We provides a built solution and optimizes the parameters. Based on the comparison with the commercial platform storage system, the problems and challenges faced by the meteorological data distributed object storage system in autonomy are discussed. Finally, this article looks forward to the distributed storage of meteorological data under domestic platforms.

1. Introduction
At present, the Meteorological Center has used a distributed system to complete data storage on a large scale. This allows data limited by traditional stand-alone storage to be stored across hosts and regions, and performance and security have been greatly improved. However, as the data volume of the Meteorological Center increases geometrically, the reading and writing of data still face huge challenges[14]. Object storage is a new type of distributed storage solution. The idea is to abstract each file into one or more objects and store them in the storage node responsible for data management. The storage node manages the data operations such as space allocation, atomic update, read-write response. Compared with file storage, the flat data structure of object storage makes it have good performance and expansion prospect when solving massive data and concurrent access to massive data, and provides a solution for high-performance meteorological data access.

Although the currently used flash memory devices exhibit superior performance compared to mechanical hard drives, the black box architecture used by flash memory devices shields the perception between storage software and flash memory devices, and loses the space for software and hardware plus collaborative optimization[4][9]. In recent years, an entirely new flash memory architecture, the Open-Channel SSD (oc-ssd) architecture, has received widespread attention in the industry. It removes the FTL in the flash memory device, exposes the internal information and control
interface to the host, breaks the semantic isolation and perception shielding of software and hardware, and provides opportunities for software-defined storage[2].

Meteorological data, as important safety information in China, has long-term dependence on foreign hardware and software platforms. Most of the business runs on foreign host systems, causing uncontrollable system and data problems, and information security is facing a serious threat[15]. The fundamental solution to the problem of information security in my country lies in achieving the independence of core technologies and products. In recent years, the successful research and development of domestic Phytium processors and the emergence of domestic Kylin operating systems and Starblaze oc-ssd products are a milestone in the development of China's localized software and hardware. It not only broke China's high-end processors and the operating system market has been monopolized by foreign products for a long time, and it indicates that China's computer hardware and software design technology has reached the world's advanced level. The performance of CPU developed by Phytium has been improved rapidly in recent years, and it has shown a trend of catching up or even overtaking in comparison with multiple high-end CPUs of Intel. Since the domestically-made Phytium CPU adopts the ARM architecture, it has lower power consumption than the CPU of the X86 architecture[3][6]. This means that the deployment of the Phytium processor in the object storage system is expected to provide high-performance services while reducing a large amount of power consumption.

We found that at present, there is no research and analysis on the implementation and performance of distributed object storage under the new domestic platform, and the realization and application of the distributed object storage system under national production is of great significance to promote autonomy. Therefore, publishing a paper that introduces the performance of distributed object storage under the national production platform can not only deeply understand the development of big data performance in my country, but also verify the a priori conditions for achieving data security in my country.

Based on the above situation, the main contributions and work of this article are as follows:

1. Built a distributed object storage system Ceph on a domestic platform in a real environment which combin a Phytium chip, Starblaze oc-ssd, and a Kylin operating system, and solutions to the problems encountered during the environment construction process were organized;

2. Carried out functional and performance tests on the system, and optimized configuration of key parameters;

3. Compared with the distributed object storage system under the existing commercial platform, the reasons for the gap are analyzed, and the prospect of the localization of distributed storage is prospected.

2. Background
At present, the design ideas based on object storage are widely used in the design of distributed storage systems, such as Ceph[8], HDFS[7], COSS[12], etc. The doctor developed the crush algorithm to implement a decentralized pseudo-random data placement mechanism for object storage. Object storage consists of the OSD responsible for managing data and the Monitor that monitors the state of the system. Architecture, each OSD manages a real physical storage device or a hard disk logical partition, and information such as the system's mapping relationship, OSD parameter configuration, and system hierarchy is stored in Monitor. When the client reads and writes from the storage system, it first sends a request to the Monitor, and the Monitor returns the system-related mapping relationships and other information to the client. By calculating the system mapping information, it can be obtained under which OSD the data to be accessed is managed, and then the client directly establishes communication with the OSD and reads and writes the data. This completely decentralized data access method abandons the traditional distributed storage architecture centered on the metadata server that requires metadata search through the metadata server, avoiding a single point of failure of the metadata server and performance bottlenecks.
Solid State Disk (SSD) with a significantly improved performance compared to mechanical hard drives. Unlike disks, SSDs have the characteristics of erasing before writing, limited life, and asymmetric read and write. Flash memory manufacturers have implemented a Flash Translation Layer (FTL) on the device side. The storage, mapping, transfer, and deletion of data in SSDs are processed through FTL and provide transparent use to users. This design pattern provides convenience while shielding the characteristics of flash memory devices, hindering developers’ optimization path. oc-ssd is a new type of flash memory architecture. The device removes the FTL on the device side, provides an interface for the host side to obtain information about the device, and provides software-defined storage opportunities for the host side software. The Linux kernel also began to support oc-ssd[1].

3. Material and methods

3.1 System structure
We first built a model for the collaborative use of domestic platforms and distributed object storage systems. The system uses a Phytium chip, uses an oc-ssd driver and Kylin system. The oc-ssd device is initialized and mounted through Lightnvm module in the kernel, and finally used as a data storage device in the distributed object storage system.

We chose Ceph as a distributed object storage system, strip the file into objects of the same size and give each object a unique ID. The placement of objects is divided into two steps. In the first step, each object is selected and placed in the specified logical storage pool pool, and a placement group is calculated through the hash algorithm. The second step is to calculate the corresponding osd group based on the placement group id, cluster mapping relationship, and placement strategy.

3.2 Construction process
We purchased a server equipped with Phytium FT-2000+/64 CPU and Starblaze open-channel SSD. The operating system is Kylin Galaxy 14. 0. Using the memory core driver and the kernel lightnvm module to abstract the flash open-channel SSD as a block device, the minic 13. 0 version of the Ceph source code was downloaded. After compiling and installing the source code, the monitor and osd services of the distributed object storage system are built on a single server. The storage device managed by the osd service is set as the open-channel flash memory card of Starblaze. Use Cephfs to mount the cluster to the client for file access.

3.3 Problems encountered and solutions
Because the Kylin operating system is not included in the list of script identification for automatic installation, building the Ceph in the Kylin operating system needs to solve the dependency problem in the server. We manually solved the problem of Ceph’s dependency package in the Kylin operating system. We use software to calculate the crc32 value to solve the source code crc calculation error problem.

3.4 Parameter optimization
We selected four parameters in Table 1 that have a significant impact on cluster performance from a large number of Ceph parameter configurations[5][13]. Using the read and write IO performance of the cluster as an indicator, find the configuration of the optimal performance of each parameter by Grid search. Among them, osd max write size specifies the maximum value that the OSD can write at one time, and filestore op threads specifies the number of IO threads.

| parameter name             | Type    | Defaults | Optimization |
|----------------------------|---------|----------|--------------|
| osd max write size         | Integer | 90       | 1090         |
| osd map cache size         | Integer | 50       | 1024         |
| filestore op threads       | Integer | 2        | 32           |
| Osd client message size cap| Integer | 524288000| 2147483648   |
4. Results
We compare the domestic platform system designed in this article with the platform currently used by the Meteorological Center, the platform hardware comparison is shown below:

|                | FT-1500A/4       | Intel Xeon E5-2603 v4 |
|----------------|------------------|-----------------------|
| Core           | 4                | 6                     |
| Cache          | 2MB Level 2 cache*2 8MB Level3cache | 15MB Level 3 cache |
| Frequency      | 1.5Ghz           | 1.7Ghz                |
| Power          | 15W              | 85W                   |

We use the fio (3.7) testing tool to test read and write scenarios of small meteorological data. We tested the throughput performance of the system before and after parameter optimization. As shown in Figure 3, Figure 4, we use the line chart test data for visualization. In order to better show the difference of the results, we compressed the coordinates of the line chart. After parameter configuration, the read and write performance of small files is significantly improved, up to 133%, because we make the size of the osd data processing is consistent with the size of the object. The performance data tested on a single machine show that although there is still a gap between the domestic platform and the current commercial platform, the gap is not obvious.

![Figure 1. Throughput in systems.](image1)

![Figure 2. IOPS in systems.](image2)

We compare the difference between the domestic platform and the current commercial platform under different loads for one hour under different loads (the system is equipped with 500G oc-ssd). For the measurement of arm platform, we use s-tui to record the power consumption. As can be seen from Table 3, in multiple load scenarios, domestic The platform has huge advantages in terms of energy saving.

|                | Domestic platform | Business platform |
|----------------|-------------------|-------------------|
| Standby        | 2.59              | 17.1              |
| Run Ceph       | 2.89              | 20.3              |
| High concurrent| 4.45              | 24.7              |

5. Discussion
Because the Phytium chip uses an arm architecture and the commercial platform uses an x86 architecture chip, the difference in architecture is also an important reason why the system shows insufficient performance in the face of complex instructions and excellent performance in power consumption. In an environment where computing and storage are becoming increasingly popular, the bottleneck of the storage server is mainly the performance of the hard disk itself, and the excellent power consumption of the domestic server can save a lot of costs for the data center.

The performance improvement of the platform after using oc-ssd is not obvious. The reason is that in this article, we simply use it but did not make special optimizations for it.
6. Conclusions
This paper combines the domestic hardware and software platform with a distributed object storage system to verify the feasibility of the system, and tests the system's function and performance for the high concurrent reading characteristics of meteorological data. The experimental results show that through parameter optimization, system performance can be increased by up to 33%, and power consumption can be reduced by up to 81% compared to existing platforms. Although there is still a certain performance gap with commercial platforms. However, domestic products continue to maintain rapid development, and the difference in performance is gradually narrowing. The low power consumption it exhibits is very attractive for a distributed architecture with separate storage and computing.

This article is only a preliminary exploration of the nationalized distributed object storage of the Meteorological Center. In the future, we will use the information exposed by the open-channel flash memory to implement software and hardware collaborative optimization to improve the data concurrent performance and data storage efficiency, and monitor the service through the monitor Migrate to a server with better computing power to achieve separation of storage and computing to further improve data computing performance.

It is believed that with the continuous development and growth of domestic production and the continuous improvement of the architecture of open-channel flash memory, the national distributed object storage system of distributed object storage will have a stronger competitiveness in the world.

References
[1] Bjørling, M., Gonzalez, J., & Bonnet, P. (2017). LightNVM: The Linux Open-Channel SSD Subsystem. Paper presented at the File and Storage Technologies (FAST ’17).
[2] Bjørling, M. J. V., Mar. (2015). Open-Channel Solid State Drives, 12, 22.
[3] Gudu, D., & Hardt, M. (2016). ARM Cluster for Performant and Energy-Efficient Storage. In Computational Sustainability (pp. 265-276): Springer.
[4] Jung, M., & Kandemir, M. (2013). Challenges in getting flash drives closer to CPU. Paper presented at the Usenix Conference on Hot Topics in Storage & File Systems.
[5] Lyu, W., Lu, Y., Shu, J., & Zhao, W. (2020). Sapphire: Automatic Configuration Recommendation for Distributed Storage Systems. arXiv preprint arXiv:2007.03220.
[6] Ou, Z., Pang, B., Deng, Y., Nurminen, J. K., Yla-Jaaski, A., & Hui, P. (2012). Energy-and-cost-efficiency analysis of arm-based clusters. Paper presented at the Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (cgrid 2012).
[7] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The hadoop distributed file system. Paper presented at the MSST.
[8] Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D., & Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system. Paper presented at the Proceedings of the 7th symposium on Operating systems design and implementation.
[9] Yang, J., Plasson, N., Gillis, G., Talagala, N., & Sundararaman, S. (2014). Don’t stack your log on my log. Paper presented at the 2nd Workshop on Interactions of NVM/Flash with Operating Systems and Workloads ({INFLOW} ’14).
[10] Zhang, J., Lu, Y., Shu, J., & Qin, X. (2017). FlashKV: Accelerating KV Performance with Open-Channel SSDs. Acm Transactions on Embedded Computing Systems, 16(5s), 1-19.
[11] Zhang, J., Shu, J., & Lu, Y. (2016). ParaFS: a log-structured file system to exploit the internal parallelism of flash devices.
[12] Chen, X.,Zhu J.,& He, X. (2017). A high performance computing oriented distributed object storage system% J Computer Engineering, 43(08), 69-73.
[13] Chen, Y.,Mao, J.(2017). Automatic optimization of Ceph parameters based on random forest and genetic algorithm, 40(2), 347-351.
[14] Yang, M., Chen, H., Chen, J., Xiao, Y., Gao, Z., & Zeng, Y. (2017). Cloud data storage technology and application of meteorological data, (06), 1017-1021.

[15] Zhou, W., & Zhang, C. (2015). Design and implementation of object data access architecture based on domestic platform. 35(08), 94-98+115.