Read/write Performance Optimization based on Ceph Heterogeneous Storage

Pengcheng Yao[1,2, a]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

[2]Chongqing Engineering Research Center of Mobile Internet Data Application, Chongqing, 400065, China

[a]2668632054@qq.com

Keywords: Ceph, CRUSH, Heterogeneous storage, Weak consistency, Multi-node read.

Abstract: The arrival of the era of big data has brought great challenges to the storage and management of massive data. Distributed storage is an extremely important solution to storage capacity pressure and cost in the era of big data. As an emerging open source distributed system, Ceph can provide three storage functions: object storage, block storage and file storage. The pseudo-random data mapping function CRUSH is used to complete the mapping of data to storage nodes, eliminating the traditional centralized metadata nodes, while system scalability has no theoretical upper limit. However, Ceph adopts a strong consistency strategy and only reads the primary node when reading objects, which results in low read-write efficiency. At the same time, Ceph does not perceive heterogeneous storage and does not make full use of SSD storage characteristics. Therefore, a combined storage strategy based on weak consistency model and multi-node read optimization is proposed to optimize Ceph storage. The system can effectively reduce the read-write delay of Ceph and improve its storage efficiency. Experiments show that compared with the original mechanism of Ceph, the proposed optimization mechanism can improve the writing efficiency by more than ten times and the reading efficiency by more than six times.

1. Introduction

Enterprises and users have an explosive demand for storage [1], and single node storage can no longer meet the demand. Distributed storage is an extremely important solution to storage capacity pressure and cost [2]. As an emerging open source distributed system, Ceph can provide three storage functions: object storage, block storage and file storage [3]. Ceph is a reliable, automatic re-equalization, automatic recovery distributed storage system [5], which can provide object storage, block storage and file storage. Ceph cancels the traditional metadata node, and complete the mapping of data to storage nodes through the pseudo-random data mapping function CRUSH (Controlled Replication Under Scalable Hashing) [6]. It only needs a small amount of local metadata and simple calculations to implement data addressing. At the same time, there is no theoretical limit on system scalability, so that Ceph is widely used. However, Ceph's strong consistency strategy and primary node reading strategy greatly limit its read-write efficiency and increase the read-write delay. And Ceph does not perceive heterogeneous storage media. So, this paper proposes a Ceph read-write optimization strategy and heterogeneous storage strategy to solve the problems of Ceph in read-write efficiency and heterogeneous storage.

In terms of read and write efficiency, during the write operation, since the client sends the data to the primary node, the data is sent to the secondary node via the primary node. When all the nodes are persistent successfully, the primary node will reply to the client. During this period, the client cannot perform other operations, which results in higher write delay in the cluster. In the read operation, only the primary node is read [4], and the I/O performance of the secondary node is not utilized. As a result, the reading efficiency of the whole cluster is low. To solve this problem, Liu [21] first proposed a dynamic replica consistency strategy based on the ratio of read and write. Through the read and write operations of the mobile phone system, the system is divided into four states:

read-write loose, read intensive and write loose, read loose and write intensive, read-write intensive. The number of copies written synchronously is obtained according to the ratio of reading and writing, which solved the problem of high write latency and improved the writing efficiency. Liu [22] proposed an LRC erasure code algorithm by improving the RS erasure code algorithm, which solved the problem of low efficiency of RS erasure code data and improved the efficiency of reading and writing. Based on two methods of command line and library librados, Zhan, et al. [7] implement the files reads/writes (also called download/upload) algorithms. For the library librados method, Zhan, et al apply two different multi-threaded algorithms to optimize the files reads/writes. Different multi-threaded algorithms are utilized for small files and large files. For small files, multiple threads read and write files simultaneously; For large files, one producer thread and one consumer thread model are designed to promote the reads/writes capability. Wu [8] proposed a hybrid distributed file system based on Ceph and HDFS. According to the characteristics of Ceph and HDFS, we proposed two file placement mechanisms based on memory resources. One is to store files with limited memory resource. In this mechanism, a K-Nearest Neighbors algorithm (KNN) is used to decide where to store a file (Ceph or HDFS) based on the input file size. The other is to store files with enough memory resource. In this mechanism, use Ramdisk with caching and parallel programming technique to improve write throughputs in the proposed hybrid distributed file system. The above schemes are either too complicated to implement, or improve writing efficiency at the expense of security, and there is no effective focus on reading efficiency.

In the aspect of heterogeneous storage, Ceph did not consider the performance of heterogeneous storage devices that cannot be effectively implemented in SSD and HDD hybrid storage environments at the beginning of its design. In order to solve this problem, Zhan [23] proposed a set of memory management module based on MRAM on the basis of Ceph distributed file system, and modified the disk-disk storage mode of Ceph system (primary replica and secondary replica are stored on disk) to MRAM-disk (primary replica is stored on MRAM, secondary replicas are stored on disk). Heterogeneous features are utilized to improve cluster performance. Stefan Meyer et al. [9] proposed to create heterogeneous pools with different characteristics and behaviors to provide different storage services. By partitioning clusters according to different disk types and allocating pools to these separate parts, heterogeneous problems can be solved. Veda Shankar et al. [10] proposed to create a multi-level cache and automatically determine the optimal cache level of active data for the application, using Intel Cache Acceleration Software and high-performance solid-state drive (SSD). It can make applications run faster, taking advantage of the characteristics of heterogeneous devices. Although the above solution solves the problem that Ceph does not perceive heterogeneity to a certain extent, the scheme is not flexible enough, and even some schemes can reduce the efficiency of reading and writing to some extent.

This paper proposes a combined storage strategy based on weak consistency model and multi-node read optimization to optimize the Ceph storage mechanism. This paper mainly completes the following aspects:

1. A weak consistency strategy of data replica based on Ceph is proposed to reduce the write latency of the cluster and improve the write performance of the whole cluster.

2. Using the Ceph cluster replica mechanism, an optimized read model is proposed to improve the read performance of the whole cluster.

3. For Ceph cluster does not perceive heterogeneity, the combinatorial optimization storage strategy is studied, to play the heterogeneous characteristics of heterogeneous replicas.

The structure of this paper is as follows: The first chapter mainly introduces related work. The second chapter mainly introduces the reading and writing optimization mechanism and the heterogeneous combination optimization strategy. The third chapter introduces experiments and evaluations. The fourth chapter is the conclusion and outlook.

2. Related Work

2.1 Ceph

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file. The first version of Ceph was released in June 2012. It is a reliable, automatic re-equalization, automatic recovery distributed storage system [11], which can provide three storage functions: object storage, block storage and file system storage. The advantage of Ceph over other storage is that it is not just storage, but also makes full use of the computing power on the storage node [12]. When storing each data, it calculates the location of the data storage and tries to distribute the data as much as possible. At the same time, due to the good design of Ceph, the CRUSH algorithm is adopted, so that it does not have the problem of traditional single point of failure, and the performance will not be affected as the scale is expanded.

The logical structure of Ceph includes four levels from the bottom up: Basic storage system RADOS, LIBRADOS, application interface, APP layer. RADOS (Reliable, Autonomic Distributed Object Store) [18] is one of the cores of Ceph and RADOS is mainly composed of two nodes: OSD and monitor. OSD is mainly responsible for data storage and maintenance functions [19]; Monitor is mainly responsible for completing system status detection and maintenance [20]. As shown in Fig. 1.

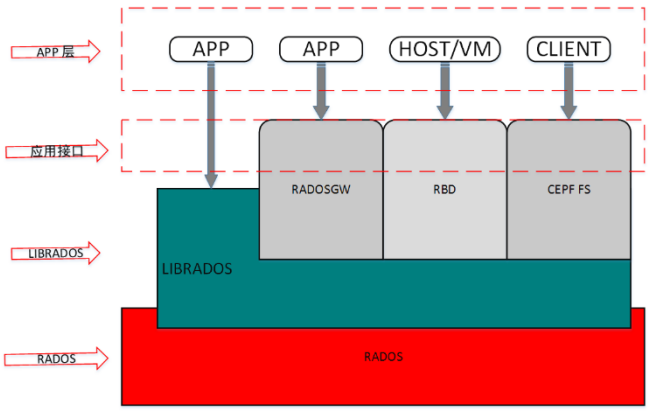

Fig. 1. Ceph data storage structure.

The bottom layer of Ceph is the object system, so a file will be divided into multiple objects, each object will be mapped to a PG (Plancement Group) [13], each PG will be mapped to a set of OSD (Object Storage Device) [14]. The first OSD is the primary node and the rest are secondary nodes. In the Ceph storage system, the data storage process is divided into three mapping processes [15]. First, file is divided into object according to the size of object. Then objects are mapped to PG, and after file is mapped to one or more objects, each object needs to be mapped to a PG independently. The third mapping is that PG, as the logical organizing unit of objects are mapped to the actual storage unit OSD through CRUSH algorithm. As shown in Fig. 2.

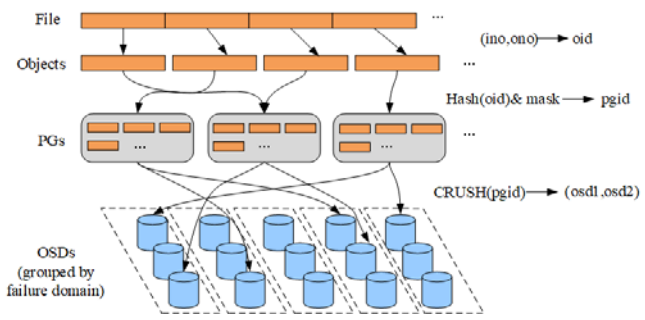

Fig. 2. Ceph data storage structure.

CRUSH implements a pseudo-random function that takes an object id or object group id and returns a set of storage devices (used to hold a copy of object OSD). The CRUSH algorithm calculates the distribution of data objects through the weight of each device. Object distribution is determined by cluster map and data distribution policy. The cluster map describes the available storage resources and hierarchies (such as how many racks there are, how many servers there are on each rack, and how many disks there are on each server). Data distribution policy consists of placement rules. Rule determines how many copies of each data object there are, and the constraints on how those copies can be stored (for example, three copies in a different rack).

2.2 Ceph Strong Consistency Strategy Model and Read Strategy

To ensure data security, Ceph adopts a strong consistency model of object writing, that is, when the object is written, the client sends the object to the primary node, and the primary node sends the object to each secondary node. The primary node receives all the replies from the completion of the write to the secondary nodes, and the primary node also writes successfully, then the reply is written to the client successfully, thereby ensuring write consistency of all replicas. But such a strong consistency strategy will result in a longer write latency [16], as shown in Fig. 3. At the same time, when Ceph reads an object, it only reads the object from the primary node, which causes the I/O pressure of the primary node to be relatively large, thus affecting the read performance. At the same time, it does not exert the I/O performance from the replica node, as shown in Fig. 4.

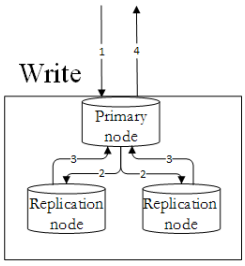

Fig. 3. Ceph write strategy.

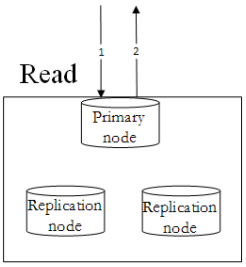

Fig. 4. Ceph read strategy.

2.3 Ceph Heterogeneous

In Fig. 2, when PG is mapped to the actual storage unit OSD of data by CRUSH algorithm, it needs to be completed by CRUSH map, CRUSH rules and CRUSH algorithm. CRUSH map is the logical location of the hard disk distribution, similar to the tree's multi-layer results, the child nodes are devices that actually store data, each device has ID and capacity-based weight, and the intermediate node is bucket. CRUSH rules determine how many replicas of each data object, and what are the restrictions on the distribution of the data, such as the ability that the same data cannot put in the same cabinet. CRUSH algorithm is a pseudo-random algorithm, which determines the storage of data by weight based on capacity. At the time of design, Ceph used capacity as the basis for selecting storage nodes, and did not consider the heterogeneity of clusters, which seriously restricts the performance of clusters in heterogeneous environments.

2.4 Motivation and Goals

Since Ceph adopts a strong consistency strategy, after the client sends the data to the primary node, the data is sent to the secondary node via the primary node. When all the nodes are persistent successfully, the primary node will reply to the client. During this period, the client cannot perform other operations, which greatly limits the writing efficiency of Ceph. And Ceph only reads the primary node during the read operation, which greatly wastes the I/O performance from the secondary node, and ultimately results in low read efficiency of the entire cluster. More importantly, Ceph does not consider the heterogeneity of storage nodes. In other words, Ceph is not perceptually heterogeneous [17]. This will make Ceph's data storage strategy not effectively exploit the performance of heterogeneous storage devices. This paper fully considers Ceph's high read-write

latency and non-perceived heterogeneity, and proposes a read-write optimization model under heterogeneous storage conditions.

3. Ceph Heterogeneous Storage Optimization Mechanism

This paper proposes an optimization mechanism to solve the problem of Ceph's high read-write latency and non-perceived heterogeneity. When reading objects, this mechanism uses a multi-node read optimization strategy to improve read performance, and further proposes a replica combination optimization storage strategy to improve cluster performance.

3.1 Replica Weak Consistency Model

This paper uses a replica weak consistency model, as shown in Fig. 5. The client sends the object to the primary node. When the object is written on the primary node, it immediately returns a write success message to the client, and then updates to the replica, thereby improving the writing efficiency. The primary node maintains the object writing process of the secondary node, and after the secondary node is written, a write success message is returned to the primary node. If the primary node does not receive the write success message from the secondary node within the set time, the object data is sent from the primary node to the secondary node, and the write process is repeated.

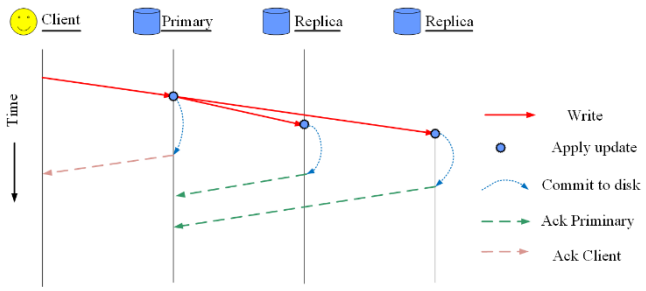

Fig. 5. Improving write strategy.

3.2 Multi-node Read Optimization Strategy

When Ceph reads an object, it only reads the object from the primary node, and does not play the I/O performance from the secondary node. At the same time, the I/O pressure of the primary node is relatively large. In order to make full use of the I/O performance from the secondary node and reduce the I/O pressure of the primary node, this paper introduces a random factor selection read replica node based on the performance of each replica node to solve the dense concurrent read bottleneck problem. Here, the comprehensive performance of the storage node is comprehensively considered according to the CPU, memory, hard disk, and distance of the storage node, as shown in (1), where wi represents the weight of each performance indicator.

$$P_i = w_1 Dis\tan ce_i + w_2 Disk_i + w_3 CPU_i + W_4 Mem_i \quad (1)$$

When there is a dense concurrent read operation, the overall performance has not changed, and concurrent reads will hit the same storage node. Therefore, a random number multiplied by performance Pi is further used here, as shown in (2). The largest node of Decision is selected from the replica node to provide read services. I/O performance from replicas is used to improve Ceph's reading efficiency.

$$Decision_i = P_i \ * \ Random_i \quad (2)$$

Since the writing strategy is changed from the strong consistency model to the weakly consistent model, when the object is read, there is a case where the primary node is written and the secondary node has not been written yet. Although the probability of this happening is extremely low, this article must also consider the occurrence of this situation. Therefore, when reading an object, it is

first determined whether the object is written from the secondary node, and then different processing is performed. As shown in Algorithm 1:

Algorighm 1 osdChoice

```
Input: object ID
Output: osd ID
1: Client gets the latest cluster map from MON
2: osds=CURSH (object ID,cluster map)
3: if write OK then
4:   P=w1*Distance+w2*Disk+w3*CPU+w4*Mem;
5:   Decision=p*Random;
6:       return the largest osd of Decision
7: else
8:   return the osd of the primary replica node;
9: end if
```

3.3 Combined Storage Strategy based on Replica Weak Consistency Model

Ceph does not perceive heterogeneous storage. That is to say, in the Ceph cluster consisting of SSD and HDD, the performance advantages of SSD cannot be exploited. This paper optimizes and modifies the CRUSH algorithm to optimize the primary replica storage to achieve heterogeneous storage characteristics and improve the overall performance of the cluster. Since this article uses a replica weak consistency model, the write latency of the primary replica determines the write latency of the object. In order to effectively reduce the write latency of the primary node and the heterogeneous advantages of the whole cluster, this paper improves the performance characteristics of SSD by modifying the CRUSH algorithm and using the SSD as the primary node. Specifically, as shown in Algorithm 2.

Algorighm 2 selectObjectStroageNode

```
Input: object ID ,N,Tries
Output: osds[]
1: for rep=0;rep<N;rep++ do
2:    tries=0,reject=false;
3:    repeat
4:      i=set();
5:      item = crush_bucket_choose()
6:      if item!=osd then
7:         bucket=item;
8:      end if
9:    util(item != osd)
10:   if collide||reject then
11:     reject=true;
12:     tries++;
13:   end if
14:   if rep==0&&reject==false then
15:     if item.type==hdd then
16:       reject=true;
17:       tires++;
18       end if
19:   end if
20:   if tries==Tries then
21:     countine;
22:   end if
23:   if(reject==false) then
25:     osds[]<- item;
26:   end if
27:end for
28:return osds;
```

When the first storage node OSD is selected by the CRUSH algorithm, it is determined whether the OSD is an SSD, and if it is an SSD, the second node is selected. Otherwise, the node is discarded and the first node is reselected until an OSD of the SSD is selected. The node does not need to perform the type judgment of the storage node when selecting the second and third storage nodes. Therefore, it is ensured that the first storage node of the three nodes selected by CRUSH is the SSD. The weak consistency model proposed in this paper shows that the object storage of the primary node determines the write efficiency of the object. By setting the primary node as SSD, the write latency can be fully reduced and the performance of heterogeneous node can be played.

In this section, this paper proposes a replica weak consistency model and a replica reading strategy to reduce the read-write delay of Ceph cluster, and proposes a combined optimal storage strategy for Ceph without perceiving heterogeneity. By modifying the CRUSH algorithm, this paper sets the primary node as SSD to fully exploit the characteristics of the heterogeneous node.

3.4 Experiments and Evaluation

For the Ceph native strategy and the strategy of this paper, the experiment tested the read and write efficiency and the hit rate of the SSD. In terms of writing, Ceph's native strong consistency strategy is compared with the weak consistency strategy and combined optimization storage; In terms of reading, Ceph's native reading strategy is compared with the multi-node read optimization proposed in this paper.

Experimental environment: four virtual machines, each virtual machine allocates 512M memory, 20G storage space, one monitor node, nine OSD nodes, one virtual machine with monitor node, and the remaining each virtual machine has one SSD node and two HDD nodes.

Test tool: Use the rados bench that comes with Ceph. The grammar of the tool is as follows: rados bench -p <pool_name> <seconds> <write|seq|rand> -b <block size> -t --no-cleanup.

Experiment and analysis:

(1) The first group of experiments tested the write latency of Ceph under the native strategy and the proposed strategy by using the testing tool Rados bench.

Compared with the write latency of the 3x replication, no replication and the method in this paper, as shown in Fig. 6. It can be found that the write delay of this method is greatly reduced. Compared with the 3x replication, the maximum write delay is reduced by ten times, and the maximum write delay is also reduced by more than four times.

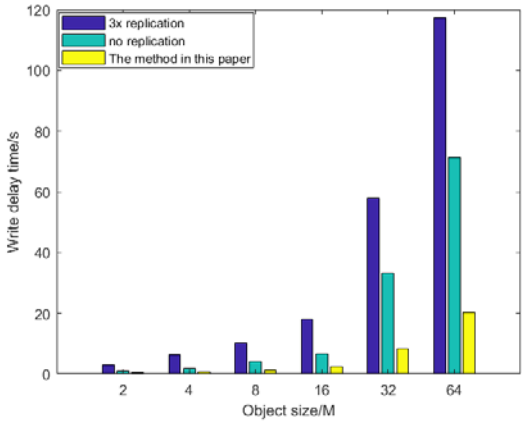

Fig. 6. Write latency under objects of different sizes.

(2) The second group of experiments tested the hit times of SSD in different times of reading objects under the native strategy of Ceph and the mehod is this paper.

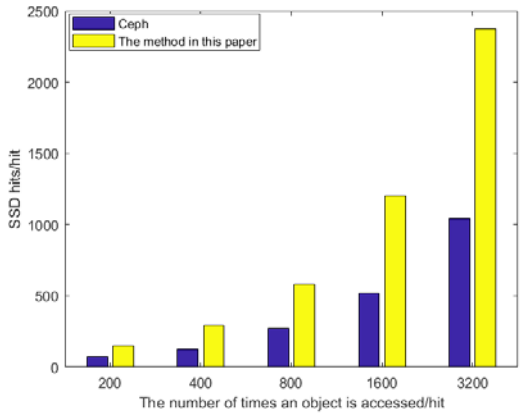

Fig. 7. Hits of SSDs under different read times.

The number of SSD hits under the native strategy and this strategy is shown in Fig.7. It can be found that the hit rate of our SSD storage nodes is more than twice as high as that of the native when reading objects.

(3) The third group of experiments tested the write throughput of the cluster in the case of writing different size objects through the Rados bench test tool under the native strategy and the proposed strategy.

The write throughput of 3x replication, no replication and the method in this paper are shown in Fig. 8. It can be found to be significantly improved by this paper. The write throughput of this paper is 38 times higher than that of 3x replication at the highest level, and more than 15 times higher than that of no replication at the highest level. With the increase of the size of the write object, the write throughput of this method increases gradually. At the I/O bottleneck of SSD replica nodes, the relative performance improvements show a downward trend, but at 64M, the relative 3x replication still increase by more than 6 times, and the relative no replication write throughput also increases by more than 4 times.

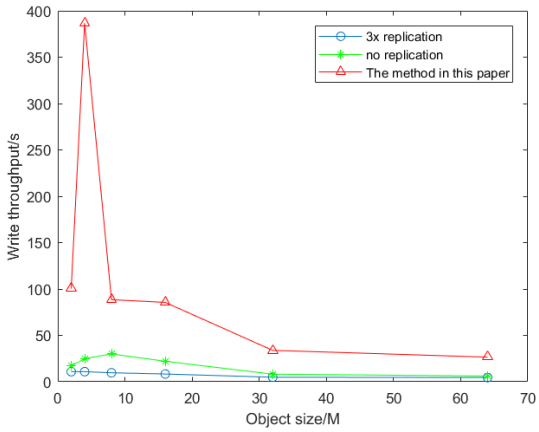

Fig. 8. Write throughput under different strategies.

(4) The fourth group of experiments tested the sequential read throughput of different size objects by Rados bench test tool under the native strategy and this strategy.

The sequential read throughput of 3x replication, no replication and the method in this paper is shown in Fig. 9. Compared with 3x replication and no replication, the sequential read throughput of the proposed method is basically consistent with that of three replicas and single replicas, when the objects are small. When the object is larger than or equal to 16M, the advantage of the proposed method in sequential read throughput begins to appear, and at 64M, the throughput of this method is more than three times higher than that of the 3x replication, no replication.

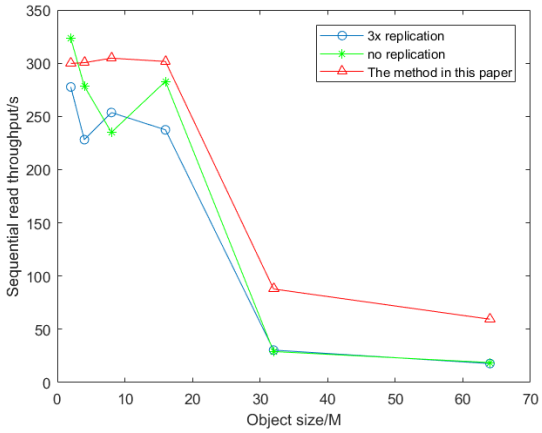

Fig. 9. Sequential read throughput under different strategies.

(5) The fifth experiment is to test the random read throughput of different size objects by Rados bench test tool under the native strategy and this strategy.

The random read throughput of the cluster under different strategies is shown in Fig.10. When the object is smaller than 32M, the random read throughput of the proposed method is basically the same as that of the 3x replication, no replication. Because when the file is too small to reach the I/O bottleneck of the HDD. When the object size is greater than or equal to 32M, the proposed method advantages of this paper begin to appear, which is about 3 times higher than the native 3x replication, no replica in random read throughput.

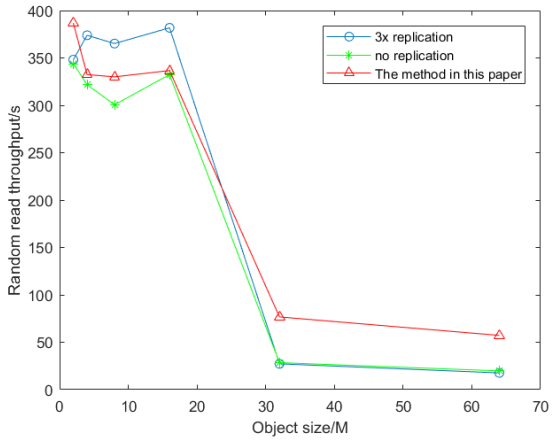

Fig. 10. Random read throughput under different strategies.

The experimental results show that the weak persistence strategy of this paper is to return the write success message to the client immediately after the primary replica is written, and combine the proposed combination optimization storage to ensure that the primary node is SSD, thereby improving the performance of the primary node. It also reduces write latency and improves write throughput for the whole cluster. This paper proposes a multi-node read optimization scheme, which improves the SSD hit rate and read throughput by comprehensive performance as the basis for selecting read nodes, and increases the SSD hit rate by more than two times and the read throughput by more than six times.

4. Conclusion

In this paper, the strong consistency strategy and read model adopted by Ceph cause low reading and writing efficiency. Therefore, a weak consistency model and multi-node read optimization strategy are proposed, which effectively improves the efficiency of reading and writing. Aiming at the heterogeneous characteristics of Ceph, a combined optimal storage model is proposed to

effectively exploit the performance characteristics of heterogeneous replicas. Experiments show that the proposed strategy effectively improves the read and write performance of Ceph clusters and exerts the heterogeneous nature of replicas.

In this model, we simply consider the case of SSD or HDD as the primary node, and do not consider the problem of storage medium from the secondary node and object heat. Next, we will do more research on this problem in order to find a heterogeneous storage solution that can target the heat of different objects.

References

[1] Xu, Jungang, et al, "A Novel Performance Evaluation and Optimization Model for Big Data System," 2016 15th International Symposium on Parallel and Distributed Computing (ISPDC) IEEE, 2016.

[2] Zheng, W., et al, "J-TEXT distributed data storage and management system," Fusion Engineering and Design 129(2018):207-213.

[3] Bollig, Evan F., et al, "Leveraging OpenStack and Ceph for a Controlled-Access Data Cloud." (2018).

[4] Zhang, Jiayuan, Y. Wu, and Y. C. Chung, "PROAR: A Weak Consistency Model for Ceph," IEEE International Conference on Parallel & Distributed Systems IEEE, 2017.

[5] Weil S A, Brandt S A, Miller E L, et al, "Ceph: a scalable, high-performance distributed file system," Symposium on Operating Systems Design and Implementation. USENIX Association, 2006:307-320.

[6] Weil S A, Brandt S A, Miller E L, et al, "CRUSH: controlled, scalable, decentralized placement of replicated data," SC 2006 Conference, Proceedings of the ACM/IEEE. IEEE, 2006:122.

[7] Zhan, Ke, and A. H. Piao, "Optimization of Ceph Reads/Writes Based on Multi-threaded Algorithms," IEEE International Conference on High Performance Computing & Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science & Systems IEEE, 2017.

[8] Wu, Chun Feng, et al, "File placement mechanisms for improving write throughputs of cloud storage services based on Ceph and HDFS," International Conference on Applied System Innovation IEEE, 2017.

[9] Meyer, Stefan, and J. P. Morrison, "Supporting Heterogeneous Pools in a Single Ceph Storage Cluster," International Symposium on Symbolic & Numeric Algorithms for Scientific Computing IEEE, 2016.

[10] Shankar, Veda, and R. Lin, "Performance Study of Ceph Storage with Intel Cache Acceleration Software: Decoupling Hadoop MapReduce and HDFS over Ceph Storage," IEEE International Conference on Cyber Security & Cloud Computing IEEE, 2017.

[11] Azginoglu N, Eren M, et al, "Ceph-based storage server application."6th International Symposium on Digital Forensic and Security, ISDFS 2018, March 22, 2018 - March 25, 2018.

[12] Sha, H. M., et al, "Optimizing Data Placement of MapReduce on Ceph-Based Framework under Load-Balancing Constraint," 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS) IEEE Computer Society, 2016.

[13] Van, der Ster, Daniel C, et al, "Ceph-based storage services for Run2 and beyond," 2015.

[14] Han, Yunjung , K. Lee , and S. Park, "A Dynamic Message-Aware Communication Scheduler for Ceph Storage System," Foundations & Applications of Self* Systems, IEEE International Workshops on IEEE, 2016.

[15] Weil S A, Leung A W, Brandt S A, et al, “RADOS:a scalable, reliable storage service for petabyte-scale storage clusters,” International Petascale Data Storage Workshop. DBLP, 2007:35-44.

[16] Liu, Gaoang , and X. Liu, “The Complexity of Weak Consistency.” International Workshop on Frontiers in Algorithmics Springer, Cham, 2018.

[17] Fei, Liu, et al, “Heterogeneous Storage Aware Data Placement of Ceph Storage System,” Computer Science (2017).

[18] J. Weng, C. Yang and C. Chang, “The Integration of Shared Storages with the CephFS and Rados Gateway for Big Data Accessing,” 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 93-98.

[19] Y. Arafa, A. Barai, M. Zheng and A. A. Badawy, “Fault Tolerance Performance Evaluation of Large-Scale Distributed Storage Systems HDFS and Ceph Case Study,” 2018 IEEE High Performance extreme Computing Conference (HPEC), Waltham, MA, 2018, pp. 1-7.

[20] K. Uehara, Y. R. Chen, M. Hiltunen, K. Joshi and R. Schlichting, “Feasibility Study of Location-Conscious Multi-Site Erasure-Coded Ceph Storage for Disaster Recovery,” 2018 IEEE International Conference on Cloud Engineering (IC2E), Orlando, FL, 2018, pp. 204-210.

[21] Liu Xinwei, “Research on Replica Consisgtency Based on Ceph Distribute Storage System,” Wuhan, Huazhong Unicersity of Science and Technology, 2016.

[22] Liu Sha, “The Research of Ceph Distributed File System Based on Object Storage,” HangZhou, Hangzhou Dianzi University,2016.

[23] ZHAN Ling, ZHU Cheng-hao, WAN Ji-guang, “Research and Implementation of Heterogeneous Object Replication Storage Policy for Ceph File Systems,” Journal of Chinese Computer Systems, 2017, 38(9): 2011-2016.