International Journal of Hybrid Information Technology
Vol.8, No.9 (2015), pp.81-88
http://dx.doi.org/10.14257/ijhit.2015.8.9.09

Optimized Architecture Design of Shared MDS Storage Systems with Hotspot Buffer

Peng Hai-Yun and Niu Ling

Department of computer science and technology, Zhou Kou Normal University
Zhoukou,China
hangfan_2007@163.com

Abstract

The ability of storage and energy consumption are the core problem faced by the data centers. The shared MDS storage system takes obvious advantage of large-scale , especially PB-level scale storage systems in solving the problem of efficiency and energy consumption.So designed the optimized architecture with hot-spot buffer, and put forward a new method of performance evaluation for the system. We also analyzed the energy consumption in parallelized data requests in the system. We find out that this architecture improved the throughout in the request of chunk data, and reduced the energy consumption. The serving efficiency and performance of the system are also improved significantly.

Keywords: *Shared MDS storage system; Energy Consumption; Performance evaluation; Hot-Spot Data Buffer*

Introduction

With the explosive growth of the amount of data, the energy consumption of data center becomes more and more crucial. The energy consumption of computing system is estimated to be 13 percent of power in America. Furthermore, the amount of computing system in the world are growing rapidly, and in these computational resources, the energy consumption of data center is occupied about 27 percent of total energy. Hence, to make sure that the computing system can be energy-saving and environmental, people must design the new storage structure to reduce the energy consumption of data center. Meanwhile, mass storage system (MSS) has been widely applied in modern engineering, and the storage system scale and the densities of data access continue to increase. When the data scale reaches to PB level, not only the present storage structure but also the data management needs to enact a fundamental change to ensure that the system performance meets the requirement for service. Shared MDS Storage System is an important solution to the problem caused by MSS. Shared MDS comprehensively considered the advantages of all network storage system, and provided a safety data shared storage structure, which has high performance, high reliability and cross-platform characteristics. Hence, it is very meaningful to accompany this storage structure with green storage concept to realize high speed information service computer system, high-capacity and going green environmental protection.

Through utilizing distributed structure, Shared MDS Storage system can make the storage device, Meta data server (MDS) and transaction server be distributed in optical network, and then divided the file system management and data I/O path, thus the speed of I/O can increases linearly with memory capacity. However, parallel distributed buffer architecture (PDBA) is mostly adopted in present memory hierarchy. Two main problems can be solved successfully by this structure. Firstly, single node failure problem in network interconnection, i.e., if the transaction server system malfunctions, the whole system will be terminated, and then the reliability will be severely affected. Secondly, high load performance of the system. In the PB level data environment, the amount of user development process is very large, the load of MDS may reach saturation, and thus the whole service system will become inefficiency. Although the distributed MDS management is the best way to handle these problems, it will certainly produce new challenges. Distributed MDS storage structure make MDS, object storage device (OSD) and RIAD as a whole, and to formulate a huge service system. Because the devices which provided service are limited, and at the same time, the other devices need no work, these systems run long time and the energy consumption is huge.

In network storage system, the data access is limited. Generally speaking, the object, which has been frequently accessed, is akewed access. Some data is large amount of user requests, high loaded. This HF-response data is the so-called Hotspot data[4-5]. To mass storage system, the generally data access present Zipf-akin distribution, Zipf law is first proposed by Prof. G. K. Zipf, when he investigate English word. This law not only has been applied in linguistics, but also widely exists in the random access of storage system data. Zipf distribution is defined as:

$$P(i) = \frac{D}{i^{\alpha}}, \quad D = [\sum_{i=1}^{n} \frac{1}{i^{\alpha}}]^{-1}, \ (1)$$

Where, I is the sort of data scale form high to low, D is constant and alpha is distribution characteristic. Generally speaking, in the large scale concurrence data access, alpha=2.1. This work is based on Shared MDS, improving the hierarchical storage structure of system. Furthermore, we design a buffer from the dispatch of metadata server, the frequently accessed data which has major influence on the load balance can be write on the buffer, and thus can reduce the access of MDS to OSD, and then the bottleneck problem of the dispatch of storage system can be solved. Besides, the energy consumption of system can also be reduced.

2. The Design of Architecture

There are two ways to design the distributed MDS architecture, i.e. unshared and shared structure. As shown in Figure 1 (a), every MDS in unshared structure has independent storage device, i.e. the metadata in the whole system has been divided into several mutual independent part. This procedure can effectively reduce the system load of every single MDS. However, if one need to migrate large amounts of data, the data synchronization, consistency cannot be guaranteed, and it is also very difficult to divide the metadata. Shared structure is that all MDS share storage device, this way not only reduce the system load, but also avoid to the case of data synchronization (as seen in Figure 1 (b)). All of the metadata is stored in metadata server, which plays the role of load controller. The user request is handle by metadata server uniformly, and MDS use high speed interconnection bus to connect object based storage device OSD. The connection between OSD and disk array is by array control unit to store object data.

In the design of shared structure, when all of OSD of MDS receive the service request, load controller arbitrate the request, and then the task is distributed to the corresponding OSD. Every OSD store the catalog data and file data, among which the stored catalog data in the OSD are almost identical, but the stored file data is different with each other, and the file data is stored in the OSD respectively. When people want to move data, only to change the catalog metadata in the file path. This is only logical, the practical data exists in the original OSD. It is thus to reduce the time wastage and performance reduction caused by data transfer from each device.

3. Systematic Assessment Model and Performance Analysis

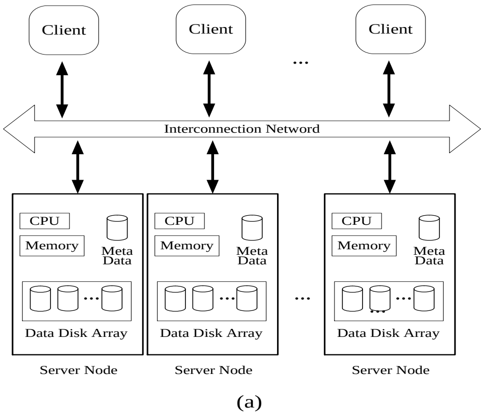

Figure 1: (A) MDS of the Unshared Structure

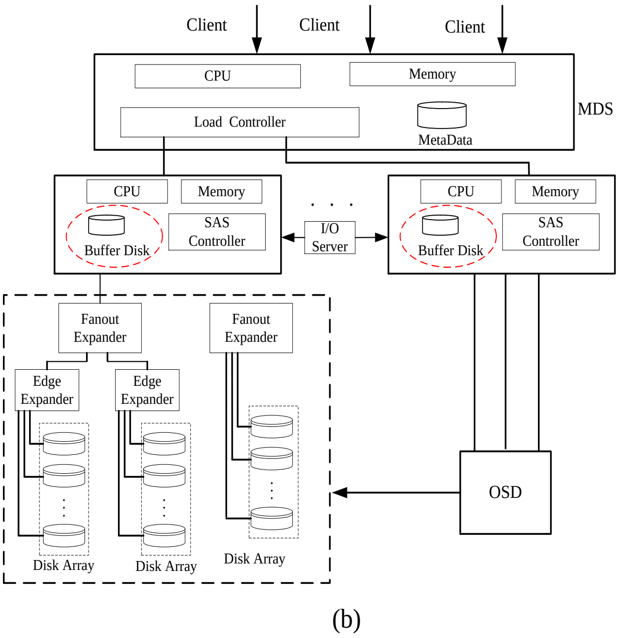

Figure 2: (B)) MDS of the Share Structure

The storage device of system is consist of abundant of disk array, these disks consume most energy of service system. The disk has different work mode at different time[6], when data request happens, the disk is at active work mode. When the disk run at high speed to accomplish reading and writing data, the energy consumption of the disk is the highest. When the data transmission is over and waiting for next data request, the disk is idle and then the running speed of disk is low, energy consumption is reduced. When there are no data request, the disk will turn to standby mode, the disk stop working, the energy consumption is the lowest. So, from the perspective of efficient and energy saving, the disk in the service system must be standby as much as possible, reducing the transform of disk working condition, distributing the access node reasonably, balancing the load, accelerating fetch speed, these ways not only guarantee the system performance, but also reduce the energy consumption of system significantly.

3.1. Systematic Assessment Model

In order to make most of the storage device of service system to standby as much as possible, we configurate a buffer disk at each service node, thus every data request can be stored at buffer area first of all, and the target disk is standby, the data is write to it until the buffer disk is full. This not only decrease the transform of disk work condition, reducing the disk energy consumption, but also guarantee the normal data request. Through the following program, the effect of energy saving of this system can be tested:

1）Constructing service system. This system is consist of N service nodes and a load balancing scheduling node. N service nodes have the same capability of storage and processing, and accept the dispatch of MDS uniformly.

2）Producing request load. The client request obey the normal distribution of which the mean value isμ and variance isσ 2 at each time interval.

3）When the write request coming every time, the buffer area disk must be examined whether it fulfill or not at first. If it has enough capacity, the request data must be write in the buffer disk at first, and when the buffer area is fulfill or the threshold time is coming, the data is return from buffer disk to target disk. The principles of the threshold time is that when the probability of no task request for the system exceeds PT，T is set to threshold time. It is supposed that the arrival of the task obey Poisson stream of which the rate parameter is nλ, the threshold should satisfy P{X(s)=k, X(t)=n}>PT, and k=n, where X(s),X(t) are the totally tasks from the beginning time to time s and t, respectively.

4）The service system use generalized supermarket model (GSM) to dispatch all of the service request, i.e. for the new task request, it is delivered to MDS dispatch node at first. From the loaded condition of present multi-server systems, MDS balanced it to allocate decision, and then this task will be oriented to appropriate server node and waiting for the service of this node. This algorithm can enhance the performance of average task time consuming exponential, further reduce energy consumption.

5）Computing the practical energy consumption of service system. To simplify model, reduce the computing complexity, ignore the time of collection of load information and task scheduling, without regard to the no change factor of service node chip energy consumption, the parts of energy consumption of this service system are: the energy consumption of dispatch node and the write back to target node, the expression is,

$$E_{sys} = \sum_{p=1}^{N} E_p^A + \sum_{q=1}^{N} E_q^S + \sum_{t\in T} E^{TP} + \sum_{m=1}^{M} E_m^T \quad (2)$$

Where E^A is the energy consumption of single disk work condition, E^S is the energy consumption of single disk at dormant condition, E^{TP} is the energy consumption of data transfer, E^T is the energy consumption of buffer area at T, M and N are the total number of one service node and data disk, respectively.

Table 1. The Working Parameter of IBM36Z15 Ultrastar

Rotations per Minute	15000 RPM	Spin down Energy	13 J
Working Power	13.5 W	Spin up Time	10.9 s
Standby Power	2.5 W	Spin down Time	1.5 s
Spin up Energy	135 J	Transfer rate	52.8 MB/s

3.2. Performance Analysis

The system has a dispatch node, i.e. MDS. It is used for store metadata and distribute the load reasonably. The dispatch node is used for manage service node, and it totally has 13 service nodes to store data. Every service node is consist of 127 fast disk. The connection between disk array and service node is used for SAS structure, and thus the handling capacity of the system can reach to 12 GB/s. We choose IBM36Z16 UltraStar as the test disk, the working parameter of this disk can be found in Table 1.

To ensure the randomness of the request, the request number obey the normal distribution with mean value μ = 25, and varianceσ2 = 10. Meanwhile, every requested data obey distribution of the mean value with 5-1000 MB. It ensure the robustness of the system at every kinds of load. The results can be seen in Figure 2.

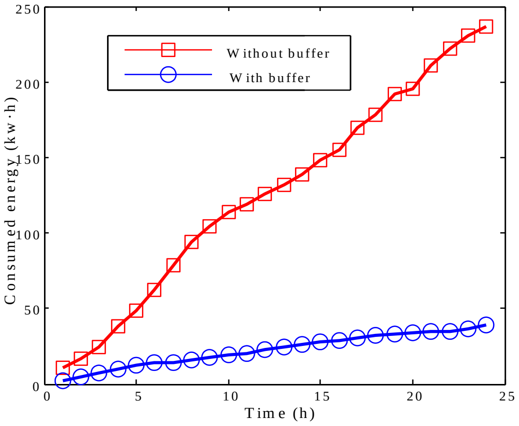

Figure 2. Comparison of Energy Consumption between the System with Buffer Area and Without Buffer Area

It can be seen from Figure 2 that for the service system without using buffer area in MDS, the energy consumption time tends to be linear rise. However, for the service system using buffer area, the increase of the energy consumption time is not significant, and is greatly reduced compared with without using buffer area. It can be explained that the task is the Poisson stream with a certain speed after some time, and the totally number of the task is a random quantity which obey normal distribution. When the task is coming, it will be stored in the buffer area at first. And when the buffer area is fulfill or the threshold time is reached, it will be write back to the target service node, thus the energy consumption of the system is the scheduling node and the service node which accepted data for a time slot, and the other node is dormant. It is greatly reduced the total energy consumption of system. If there is no buffer area, the task is coming randomly, the write back target service node is also random, so all of the service nodes is working normally to respond the distributed task by the scheduling node. Because the effect of energy saving of the system with buffer area is remarkable, all the following test is accomplished by the system with buffer area.

To test the anti-pressure ability of the service system (the extreme case that the distribution of the amount of the request data is in a much bigger value when the amount of request data is large). It is supposed that the mean value of the task request data isμ s =25, varianceσ s2 =5. Meanwhile DS, which is the amount of the requested task data, obey the normal distribution with mean value asμ DS, and variance asσ DS2. To decrease the fluctuating of DS, and to make the amount of data DS distributed densely in a small range, the variance should be small. In the test,σDS2=25, and the mean value of DS is increased gradually to add the system pressure caused by load number. It is can be seen from Figure 3 thatμDS increases gradually between 100 and 1000, and the mean response speed increases first, and reaches to a certain value and decreases later. Meanwhile, the energy consumption of the system tends to increase with the increase of μDS. It can be interpreted that the system has enough buffer memory space to store much more task request data when μDS is small. And then the mean respond speed of the system increases with the increase of μDS. However, whenμDS increases to a certain value, the volume of the system buffer is limited, we need to write back the buffer area data, and to dispatch new task. So the mean respond speed decreased gradually, and then the disk keep running, further to switch working condition, thus the energy consumption keep increasing.

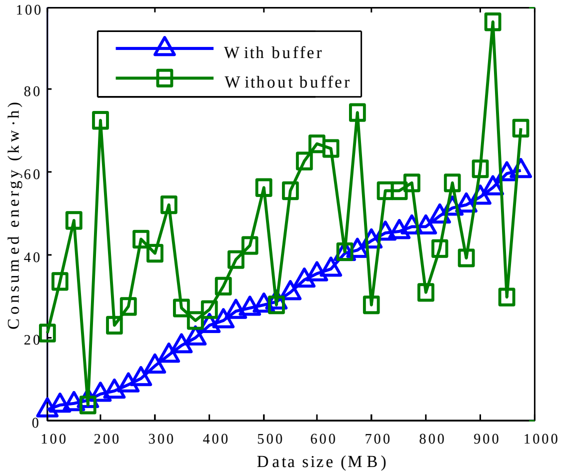

Figure 3. The Effect of the Value of Requested Data on the Mean Respond Speed of System and Energy Consumption

The energy consumption of system will increases sharply under the condition of extremely dense data load, but it will reduce lower under the condition of normal task request. It is depicted in Figure 4 that the totally energy consumption of the system has great different under different task request. RS is the random number, i.e. the task number at every time, and RS obey the normal distribution with mean value asμ_{RS} and variance as σ_{RS}^2. The value of every requested task obey the distribution of the mean value at [5, 1000]. It can be seen from Figure 4 that with the increase ofμ_{RS}, the energy consumption of system is also increased. That is because more of the amount of request every, more possibility of the full of the buffer area. It need additional energy consumption to write back to target disk. To analyze the effect of the fluctuating of request number on the energy consumption of system, σ_{RS}^2 varied at the value of 5, 10, 15, 20. It can be seen from Figure 4 that when the fluctuating values are 5 and 10, the energy consumption of the system is very close due to the vicinity between the random of the requested number and the fluctuation. When σ_{RS}^2=15, the degree of dispersion of the requested task is much bigger, the buffer area is much harder to manage. It bring the additional energy cost by dispatch, so the energy consumption of the system increased greatly [7].

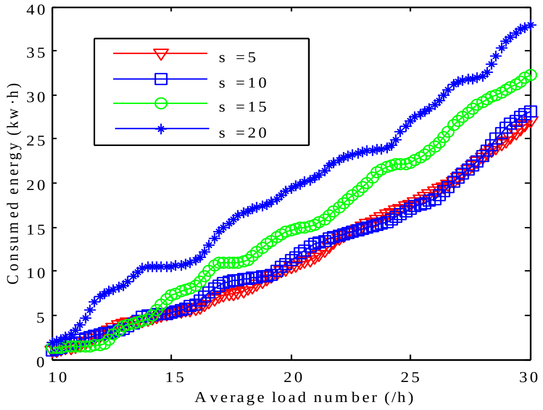

Figure 4. The Energy Consumption with Different Load

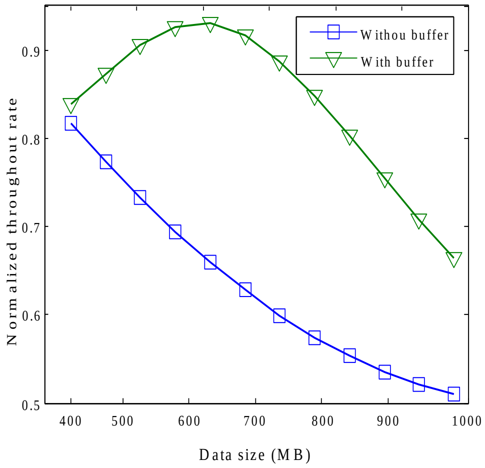

Figure 5. The Effect of the Request Data on the System Handing Quantity

The effect of the value of the requested data on the handing capacity of the normalized system is shown in Figure 5. The definition of the handing capacity of the normalized system is the ratio of the system processing data at unit interval and the totally requested data. The results show that when there is no suffer area, the handing capacity decrease greatly with the increase of data request, and it is related to the system scale. Meanwhile, when the buffer area exists, the handing capacity of the system increases first and decreases later. That is because the little scale data can be contained in the buffer area, decreased the distributed time to OSD. However, when the data scale is large, it exceeds the buffer area volume, and then the system must process data by dispatch to decrease the handing capacity.

4. Conclusion

In this paper, we proposed an optimization design for the shared MDS storage structure with tropical data buffer. At first, we introduced the advantage of shared MDS storage system in the large scale data request. Based on the tropical data buffer, we design a storage structure, and carried out several tests. The results show that this service system not only guarantee the server efficient, and also decrease the energy consumption of system greatly.

Acknowledgment

This research is supported henan province natural science foundation (2010b520036).

This research is supported natural science of china foundation (61103143).

References

[1] Zhang Guigang, LI Chao, and XING Chunxiao .A Green Computing Model Based on Cloud Environment[J]. Journal of Chinese Computer Systems. 2013, 34(5): 1016-1020.
[2] Qilin Li and Mingtian Zhou. The Survey and Future Evolution of Green Computing[C]. IEEE/ACM International Conference on Green Computing and Communication(GreenCom), 2011:230-233.
[3] Fang Yuan, Du Zhu-Ping, and Zhou Gong-Ye. MetaData Management Policy based on OBS System, Computer Engineering and Application, 2012,3 (02):25-27.
[4] Zhou Wengang, ZHAO Yu, and WANG Feng. Cancer Gene Clustering Algorithm Based on Quantum-Behaved Particle Swarm with Comprehensive Learning Strategy[J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37(4): 59-63.
[5] Zhou Gong-Ye, Wang Yan, and Lu Chun-Huai. Research and Implementation of Object-Oriented Intelligentized Storage Devices, Computer Engineering and Science, 2007, 29(3):124-127.
[6] Lin Ziyu, Lai Mingxing, and Zou Quan. Probability-Based Buffer Replacement Algorithm for Flash-Based Databases [J]. Chinese Journal of Computers,2013,36(8):1568-1580
[7] PENG Hai-yun , and ZHOU Wen-gang. Based on WDM of Disk Immunity Systems[J]. International Journal of Security and Its Applications, 2014,8(2) : 659-669
[8] Cai Tao, Niu Dejiao, and Liu Yankuan. NVMMDS-Metadata Management Method Based on Non-Volatile Memory [J]. Journal of Computer Research and Development,, 2013,50(1):69-79.
[9] Song Lina, Dai Huadong, and Ren Yi.A learning Method of Hot-Spot Extent in Multi-Tiered Storage Medium Based on Hyge Data Storage File System[J].Journal of Computer Research and Development, 2012, 49(9): 6-11.
[10] Lin Ziyu, Lai Mingxing, and Zou Quan. Probability-Based Buffer Replacement Algorithm for Flash-Based Databases [J]. Chinese Journal of Computers, 2013,36(8):1568-1580
[11] Michael Mitzenmacher, The Power of Two Choices in Randomized Load Balancing[C], IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2001, 12(10):1094-1104.
[12] Manzanares,A.; Xiaojun Ruan; Shu Yin; Jiong Xie; Zhiyang Ding; and Yun Tian. Energy Efficient Prefetching with Buffer Disks for Cluster File Systems[C], 2010 39th International Conference on ICPP, 10.1109/ICPP.2010.48,:404-413.
[13] Steve Greenberg, Evan Mills, Bill Tschudi, Peter Rumsey and Bruce Myatt. Best Practices for Data Centers: Lessons from Benchmarking 22 Data Centers[C], ACEEE Summer Study on Energy Efficient in Buildings, 2006, p.377-380

Authors

Peng Hai-Yun, received the B.Eng degree in Computer science from Henan University and M.Eng degree in Computer science from Huazhong University of Science and Technology. She is currently researching on computer application technology.

Niu Ling, received the B.Eng degree in Computer science from Henan normal university and M.Eng degree in Computer science from Chengdu University of Technology. She is currently researching on computer application technology.