The Performance Analysis of a Service Deployment System Based on the Centralized Storage

Zhu Xu Dong
School of Computer Science and Information Engineering
Zhejiang Gongshang University
310018 Hangzhou, China
zhuxd@zjgsu.edu.cn

Journal of Digital Information Management

ABSTRACT: *Because of its data centralized management characteristic, which conveniently realize resources sharing and the data fast clone, the service deployment technology based on the centralized storage can achieve fast deployment for the system service through the network boot methed or the method of downloading disk image from the remote storage server. However, for the data centralism the I/O load is also congested on the storage server, which creates the drop of the overall system performance and the reducing of the service quality. So we need to test and analyze its performance factors, which provide the clue for the solution of the performance bottleneck. In this paper, we use the SonD deployment system, which is based on the centralized storage, as the analysis prototype to carry on the analysis to the client boot process, evaluate its performance influence factors such as the memory size of storage server, data distribution on the storage server through the real or simulated method, and give suggestion of the performance optimized.*

Categories and Subject Descriptors
D.4.2 [Storage Management]: Distributed memories; **E2 [Data Storage Representation]**

General Terms
Data Storage, Remote Server, Disk Management

Keywords: Performance Analysis, Centralized Storage, Service Deployment

Received: 1 July 2011, Revised 24 August 2011, Accepted 30 August 2011

1. Introduction

With the rapid increase in the number of computers, the management of large-scale computer clusters has become an increasingly serious problem. The modern computing centers and data centers supervise thousands of computer nodes, the system installation, setup, maintenance and software service of which lead to extra time and labor costs. Thus the deployment of computer system and service has become a focus of study in academic and industrial circles.

Present studies have used a variety of techniques to solve deployment issues, which includes disk mirror as the focus of study owing to its simplicity and high efficiency, such as Symantec's Ghost [1], Frisbee system [2] from University of Utah, IBM's Tivoli Provisioning Manager for OS Deployment [3] and COD (Cluster-on- Demand) [4] system from Duke University. All these systems can install disk data from net to local storage devices through mirror and network multicast technology. At the back-end of these deployment systems generally there is a centralized storage space for saving the original disk image. But in the deployment process, it won't boot until the disk image has been downloaded to local disk synchronously, seriously affecting the deployment speed.

The development of diskless boot and network storage technology makes the appearance of service deployment system based on centralized storage possible. emBoot [5], IBM's Blutopia [6], IOMan [7] from Hefei Industrial University and SonD [8] developed by National High-Performance Computer Engineering Center all belong to this service deployment system based on centralized storage. In this way all the client data are stored in the back-end storage system and it can only boot and provide system service on net through diskless technology, which is characterized by its avoidance of downloading data to local disk, shortening service deployment time.

But the data centralization leads to the concentration of system load at the back-end storage, affecting system availability and scalability. For example, when the clients supported by SonD increase to a certain number, the client back-end data access latency will be far more than that of ordinary systems, making it no longer available. Aiming at this problem, with SonD system as prototype, we analyze data access features of service deployment based on centralized storage, model and test the deployment system from data cache and disk I/O these two factors. Then optimization is given, making it possible to support more number of clients under the conditions of data access latency.

This paper is organized as follows: section 2 introduces the data service model of deployment system; section 3 tests and analyzes deployment system performance on its storage server cache and disk I/O for model validation; section 4 gives the deployment system performance optimization; section 5 evaluates the optimization program through actual system load; section 6 are the conclusion of this paper and description of further research plans in near future.

2. Data service model of deployment system

2.1 SonD system overview

SonD system is a typical service deployment system based on centralized storage, which consists of the backend virtual shared volume management system (VSVM), the front-end client system and the service management system. Virtual shared volume management system is responsible for maintaining each client's network disk mirror and service templates; the front-end client system includes diskless boot client in its software form and nHD card in its hardware form, and it functions as a virtual machine which supports remote boot in the cloud computing platform; the client is responsible to run the deployed service and support data exchange and user authentication with back-end storage server and management server; service management system is mainly responsible for the mapping of network disk to clients, and the monitoring of system operation. SonD system firstly creates network disk and allocates storage resources at back end, while pre-installs system and the data required for service in network disk; then it is bound to a specific client, you can have access to network disk through standard nbd protocol or iSCSI protocol once the client system is validated in management server so as to initiate the system and corresponding service, thus completing the service deployment.

2.2 SonD system network disk mirror storage model

SonD, based on centralized storage, achieves at the back end data sharing, physical resources distribution on-demand and rapid service cloning. Meanwhile diskless technology makes the dynamic service switching with great flexibility possible. The actual use shows that SonD system in a single storage server configuration can effectively support the use of more than 20 clients.

Both the management server and storage server of SonD system use Linux operating system as their platform, thus we employ the Linux kernel block device level tracking tool blktrace [9] [10] provided by open source community to record disk related events in tests, and take the recorded trace as the basis of our simulation test analysis. Detail information of blktrace will be given in the following chapter.

2.3 Data path analysis of SonD system

Although the SonD system has obtained certain results in practical application, we found in actual use that there is no difference in boot time and using experience between SonD system and local system when deploying single client. However, when the SonD system deploys 40 clients at the same time, its boot time will last for more than ten minutes, making the system unavailable. In order to analyze the cause of this problem, we firstly do analysis of SonD system data path.Since the hardware solution using nHD card enables the SonD system to support various operating systems, giving it more application value and universal significance, we make SonD system which uses nHD card as client system the object of analysis.

According to Figure 1, the read requests and write requests of the SonD system start from AS, reach bdserver through network, and then pass the file system of the storage device (SN). If hit by cache, they return directly, if not, the requests will be sent to VSVM and finally reach the disk. To facilitate our test, we firstly divide the factors that may affect performance in the entire io path into the following parts: cache processing, VSVM mapping and physical disk access. Cache processing is a critical step, and cache refers to the page cache of file system in SN. In the test, we conduct indepth tracking and analysis on cache behavior. VSVM is the core of the entire storage system, which is responsible for I/O mapping and COW operation between source and snapshot. The I/O mapping in VSVM processing is a very simple process to modify the memory pointers, thus it does not consume too much CPU resources. Disk is the component that finally completes data access.

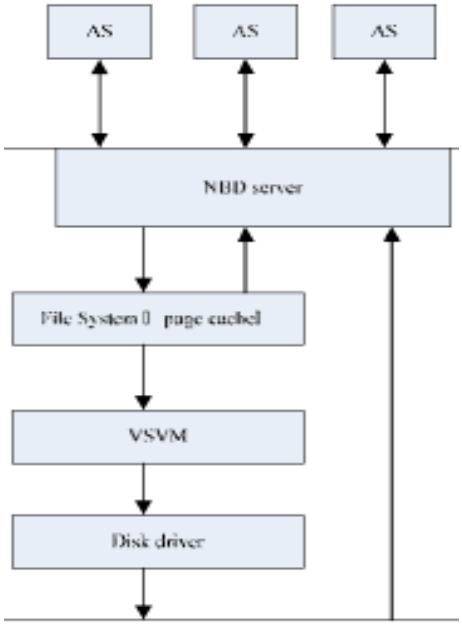


Figure 1. I/O path of SonD system

Blktrace, monitoring tool used in this paper, is a tool based on relayfs file system to monitor and track the Linux kernel block level I/O operation, and to provide users with detailed information on queue operation request. According to the need of actual test, we modified this tool and defined a new monitoring point in the kernel Meanwhile we evaluated the load blktrace brings to the system and the test results show 5% to 10% increase of load, which is acceptable for conducting this test.

2.4 Test environment simulation

Our test environment is as follows:

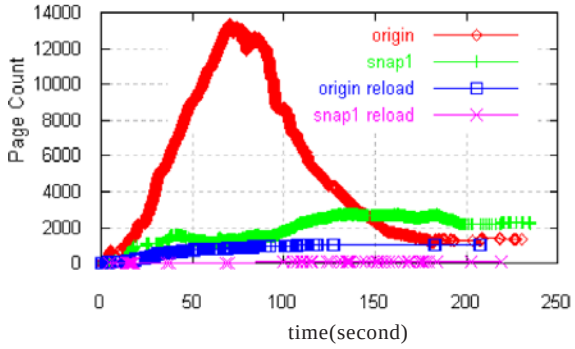
storage server	
CPU	Intel®Xeon(TM) CPU 3.00GHz
Mem	1~4G
Ethernet Controller	Broadcom BCM5721Gigabit
RAID Controller	3Ware 9500-12 SATARAID
Disk Subsystem	Local 160G + Raid0 12*160G

Table 1. Storage server configuration

AS	
CPU	Intel®Xeon(TM) CPU 2.40GHz
Mem	1G
Ethernet Controller	Intel 8254EI Gigabit
Disk Subsystem	Local IDE 120G
Operation System	RedHat 2.6.11-1.1369_FC4

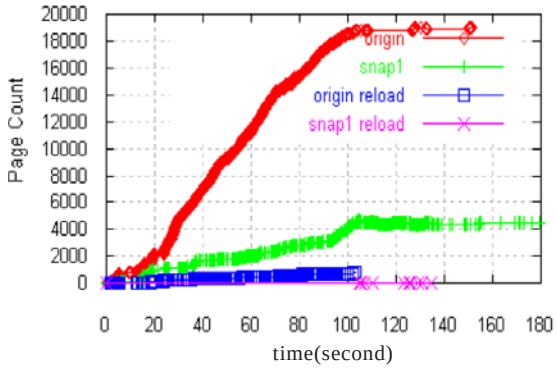
Table 2. Application server configuration

1G-alone Cache Usage and Cache reload-64 Clients for 64 bit SN



(a)

2G-alone Cache Usage and Cache reload-64 Clients for 64 bit SN



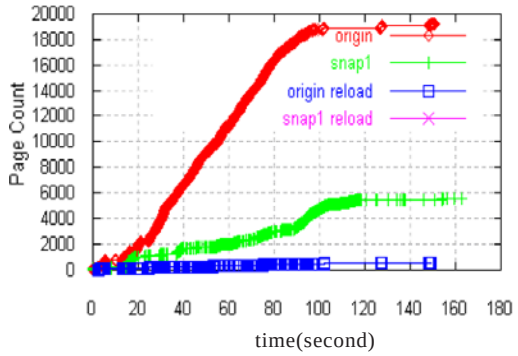
(b)

3. Test results and analysis

3.1 Test and analysis of cache behavior

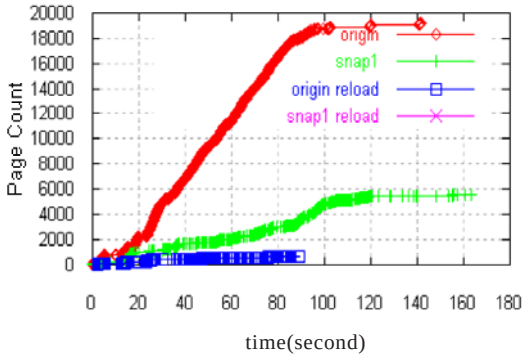
Our system test is conducted on the released RedHat FC4 x86_64 which is based on linux-2.6.15.4 kernel +blktrace kernel patch. In the 64-bit system, we do similar tests according to different memory sizes, and the test results of 64 clients booting together are shown in Figure 2.

3G-alone Cache Usage and Cache reload-64 Clients for 64 bit SN



(c)

4G-alone Cache Usage and Cache reload-64 Clients for 64 bit SN



(d)

Figure 2. Memory usage of storage server under the 64-bit environment

In 32-bit system test, in addition to the system disk in the server, there is only one other storage device made of raid card and 12 disks which functions as a centralized storage resource. The source and snapshot that provide service are both stored in this device, so their data are scattered in 12 disks. Through 32-bit system test, we find cache pages allocated to source data are the maximum, indicating that data on it are accessed most often. If we distribute it and other data to the same disk, it would form competition with the read of snapshot private data, resulting in performance degradation. Thus in 64-bit system, with the same number of disks, we take source data out independently and use a certain disk as their physical storage resource, while the other 11 disks and raid card form a storage device for snapshot. In this way the competition on magnetic heads between shared data and private data is reduced and the performance is improved. In the 64-bit system test, we find that when memory grows to a certain size, simply adding physical cache will no longer be able to effectively improve system performance. From the figure we can see when physical memory changes from 1 G to 2 G, the boot time of system is shortened quite obviously, reduced from about 210 seconds to about 140 seconds, the performance improved by nearly 1/3; but when it changes from 2 G to 3 G, or from 3 G to 4 G, the time of clients booting together does not show any significant decrease. And we can also see in reload number that the snapshot reload number drops

from 114 to 32 as storage server memory grows from 1 G to 2 G and there is limited range to drop when memory size grows.

3.2 Test and analysis of disk behavior

Through 64-bit test we find when we separate source data from snapshot data on physical distribution, the completion time of booting together descends from 258 seconds to 210 seconds, a very large drop, in the simulation test where the memory size of storage server is 1 G. And in the simulation test of 64 clients, the effect of changing data distribution has no great difference from that of adding memory size to 4 G, the boot time of both are about 210 seconds. It shows data layout and disk access behavior have obvious impact on the overall system performance. In this test, we use a bit of the average disk response time to describe disk load at this time period. We test both situations in which source and snapshot are separated or distributed in the same physical storage resource, the test results are shown in Figure 3.

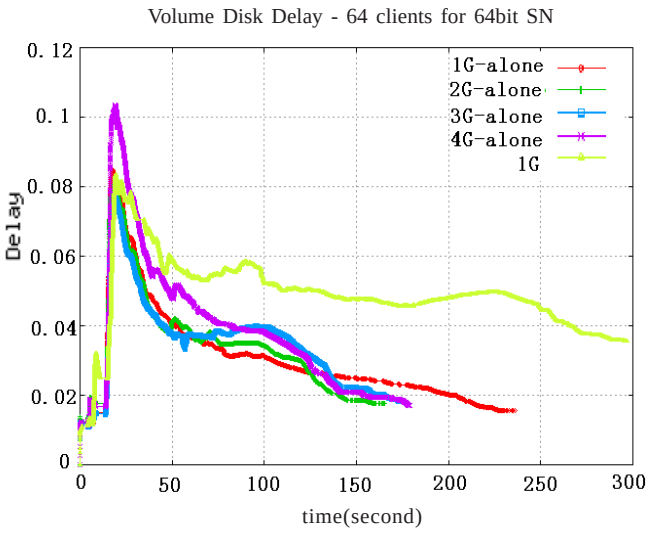


Figure 3. Mean value of disk response time

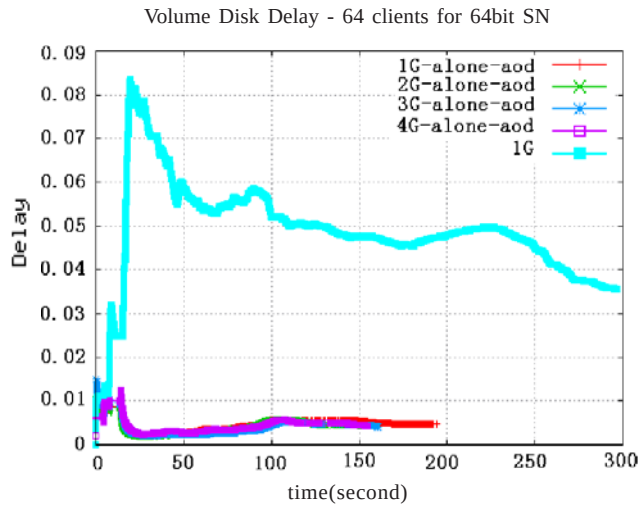


Figure 4. Mean value of disk response time with allocation on demand mechanism support

We can see from Figure 3, when source data are distributed in an independent disk, compared with that when source data and snapshot data are distributed in the same physical device, the average response time is significantly smaller, so its start time is much less. In the first 30 seconds, the average disk response time in both cases have no fundamental differences. This is because at this time clients read shared data from the source and there is no write operation of snapshot private data, thus no competition among large amount of requests. At the same time we can also see from the figure, disk request response time is not improved as the memory size grows, indicating that when memory reaches a certain size, main factors which affect disk request response time are client number and data distribution on disk.

To further verify the impact of data distribution on system performance, we changed the physical storage resource allocation method of source and snapshot in storage server so that it can support the allocation of physical resources on-demand. In the use of on-demand allocation strategy, it only dynamically allocates the space required for source and snapshot in case of a write operation. Therefore source data and snapshot data are distributed continuously on the disk to shorten the distance of head movement, thus lessen disk response time. The test results in Figure 3 and Figure 4 show that the disk response time in each test is within 0.02 seconds in on-demand allocation situation, while under fixed allocation strategy, the average disk response time is more than 0.02 seconds. We can find by comparing these two tests that data distribution has a significant impact on disk response time.

4. Conclusion

We find through the test the main factors that affect performance in service deployment system based on centralized storage as follows:

Memory size of storage server. The memory size of storage server has certain effect on overall system performance. But once the memory size of storage server is sufficient to support the data required to boot a single client, there is no great help to the service quality provided by storage server by increasing its memory size. Thus memory size is not the decisive factor of system performance.

Data distribution on physical disk. Firstly, we can increase the concurrency and performance of storage system by scattering the source that saves frequently accessed read-only data and the private data that need read and write operation in different devices. Secondly, when the storage device allocates physical storage resources on-demand, the data accessed are distributed continuously in storage device, which helps improve the performance of this IO operation.

5. Acknowledgment

This work is supported in part by Natural Science Foundation of China “Research on the snapshot data security storage technology for authorization of release.”, and by Natural Science Foundation of Zhejiang Province, China under Grant No. Y1101316.

References

[1] SYMANTEC.http://sea.symantec.com/content/article.cfm?aid=99. 2004.

[2] Hibler, M., Storller, L., Lepreau, J., Ricci, R., Barb, C. (2003). Fast, scalable disk imaging with frisbee, *In*: USENIX03 (San Antonio, TX, June), USENIXASSOC, p. 283–296.

[3] Tivoli, (2006). Tivoli provisioning manager for os deployment. http://www.rembo.com/index.html.

[4] Moore, J., Irwin, D., Grit, L., Sprenkle, S., Chase, J. (2002). Managing Mixed-Use Clusters with Cluster-on-Demand. Technical report, Department of Computer Science, Duke University.

[5] http://www.emboot.com/products_netBoot-i.htm

[6] Oliveira, F., Patel, J., Van Hensbergen, E., Gheith, A. Rajamony, R., Blutopia. Cluster Life-cycle Management. Technical Report, IBM.

[7] Xia Nan Zhang Yaoxue Yang Shanlin Wang Xiaohui, IOMan, (2007). IOMan: An I/O Management Method Supporting Multi- OS Remote Boot and Running, Journal of Computer Research and Development, (2) 317-325.

[8] Liu Zhenjun, Xu Lu, Yin Yang, Blue Whale SonD, (2005). A Service-on-sDemand Management System,Chinene *Journal of Computers,* (7) 1110-1117.

[9] Alan D. Brunelle. Block I/O Layer Tracing: blktrace. http://www.gelato.org/pdf/apr2006/gelato_ICE06apr_blktrace_b runelle_hp.pdf

[10] http://git.kernel.org/?p=linux/kernel/git/axboe blktrace.git;a =blob;f=README

[11] Yang, Y., Zhenjun, L., Shuqing, Y., Shuo, F., Zhiyong, S., Huan, Z., Lu, X.(2006). Vsvm-enhanced: a volume manager based on the evms framework. *In*: Grid and Cooperative Computing Workshops. p. 424 – 431.