Performance Improvement of the Cloud Computing System using I/O Acceleration

[1]Kijo Yu, *[2] Daesik Ko
[1]*Dept. of Electronic Engineering, Mokwon University, kjyoocf@gmail.com*
[2]*Dept. of Electronic Engineering, Mokwon University, kds@mokwon.ac.kr*

Abstract

The cloud system has been rapidly expanded (has been expanding/ has expanded) based on its rationality and convenience. Its performance is a sensitive issue for both the service provider and user. In the past, if there was a lack of server performance for a cloud service, one of the common solutions was to increase the number of servers. However, this solution had its limits because the bottleneck phenomenon was concentrated in the I/O. This study designed a x-86 server equipped with the I/O accelerator and conducted a comparative analysis of its performance with other servers equipped with a SSD-based I/O accelerator. The findings showed that the IOPS performance of the server using the SSD-based I/O accelerator was improved 32 times than that of the normal server. This result will contribute to the improvement of the cloud computing service based on the x-86 server.

Keywords: *cloud computing, SSD, I/O performance, virtualization, I/O acceleration, cloud design*

1. Introduction

SSD is drawing great attention as the alternative storage device to HDD. Compared to hard disk that is performed mechanically, flash SSD ensures better performance in damage, noise, heat, and power consumption, as well as fast and even access speed due to its electromagnetic characteristics. Flash SSD is increasingly recognized as a superior storage device to HDD in terms of price and performance not only in the mobile and pc industry but also in the enterprise environment for high-volume big data [1-3].

However, it is impracticable to apply it to everywhere because SSD is more expensive than HDD. For this reason, studies on improving the whole system performance by configuring a server system with relatively cheap HDD and the minimum cache with relatively expensive SSD have been widely performed in the IT industry. Most of these studies are to configure the storage medium with Flash SSD or DRAM SSD, or to optimize the types of file systems, and thus, have difficulties in obtaining competitive price [4, 5].

Although there are many influential factors on the server performance, I/O interfaces and I/O bottlenecks are reported to be the most influential ones. This study designed the x-86 server equipped with the minimal-SSD based I/O accelerator and conducted a comparative analysis of its IOPS performance with that of the normal x-86 server.

2. Cloud computing based on the x-86 server with an I/O accelerator

2.1. Cloud design based on the x-86 server

When designing the x-86 server for clouding computing, effective resource use by using virtualization software and sharing server resources should be considered in advance. The whole I/O is duplexed and the network is with VLAN tagging in order to obtain the security of customer agency. For improved operation efficiency, the HW management port is connected with an operation

* Corresponding Author
Received: Jan. 28, 2015, Revised Mar.01, Accepted: Apr.02, 2015

management network and an HW remote control function is provided. An example of the x-86 server diagram in the cloud environment is presented in Figure 1 [6-8].

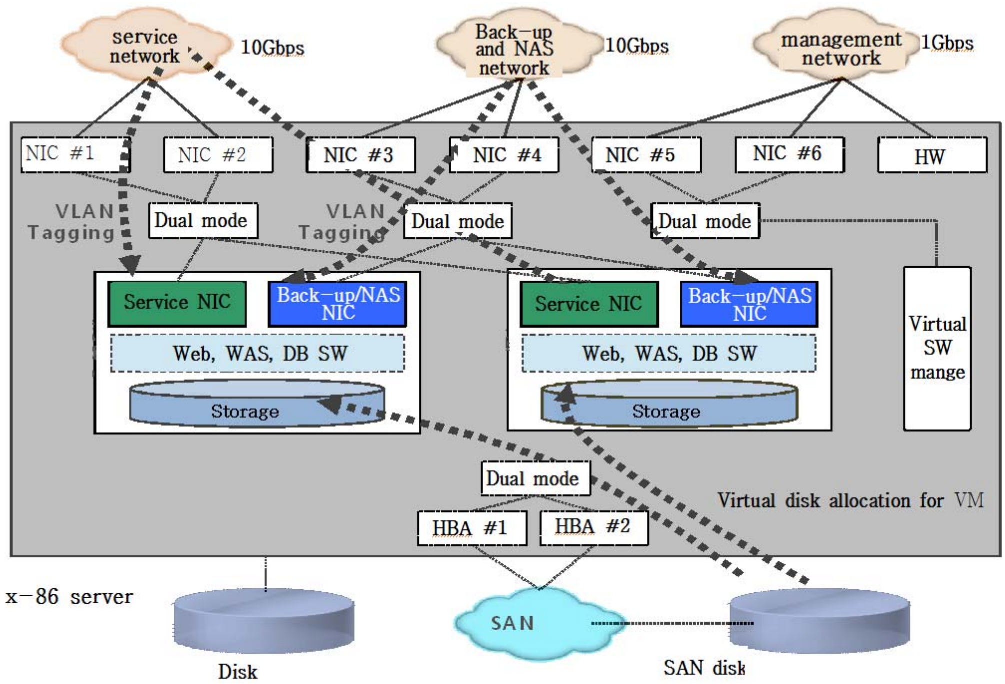

Figure 1. Blueprint of clouds x-86 server

The fundamental design criteria in realizing the cloud include physical resource sharing, duplex configuration of important parts, security considerations, and physical separation of operating circuit. For physical resource sharing, virtual resource allocation techniques are applied to make a virtual machine effectively use the server resources (CPU, Memory, Disk). For duplex configuration of important parts, the duplex configuration of important I/O channels such as network and HBA is performed to be prepared against a network or storage disorder. For security considerations, the importance of security for each customer agency is considered to secure the absolute independency of each network through VLAN Tagging techniques. For physical separation of operating circuit, service (including backup/NAS) operation management network and HW management channel are physically entirely separated.

An example of a physical configuration plan is as follows: for port allocation, the dual port NIC for service, the dual port NIC for backup and NAS network, the dual port NIC for operation management, and the one port NIC for HW management (using onboard NIC) are set up. For storage device configuration, the dual port HBA for sharing storage connection is set up and the SAN storage is connected. Also, the two 300Gb internal disks are configured as RAID-1, and one 300GB is configured as the working space.

Virtualization SW configuration involves the installation of virtualization software (hypervisor) and configuration and duplexing of an virtual device for each interface. When we configure service network, backup, and NAS network, VLAN Tagging techniques are applied. Through the activation of the virtualization function, duplexing configured CPU, Memory, and I/O resources are allocated on each dynamically generated virtual machine.

2.2. SSD flash memory operations

SSD is a data storage device using flash memory. Inside SSD, a number of flash memories and RAM buffer are connected to the bus. The SSD firmware, FTL, controls and manages the operations transferred to each flash memory and manages the limited number of writes for a flash memory unit through the sector remapping and garbage collection. The SSD Block diagram is presented in Figure 2

[2, 3].

It consists of the ARM Processor Core executing the SSD firmware code and the SRAM buffer used as cache inside SSD, and each SDRAM buffer transfers data through the AMBA bus.

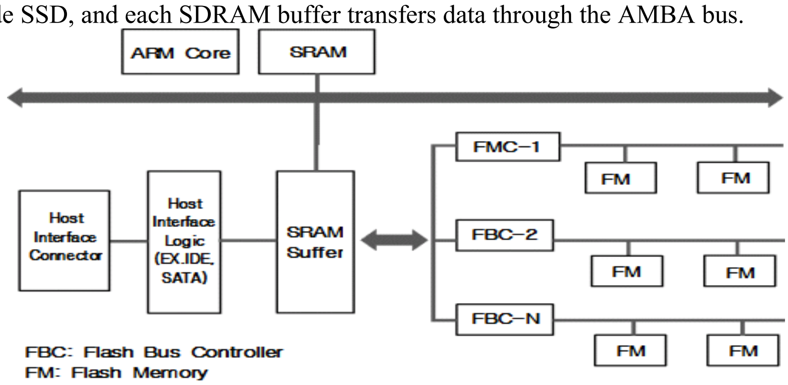

Figure 3. SSD Block diagram

Address translation techniques of FTL include page mapping technique and block mapping technique. The address mapping table is configured at the page-level of the logical sector size and the numbers of each logical sector are synchronized by mapping to the whole physical page numbers of SSD. The physical page numbers consist of physical block numbers and relevant page numbers in the physical blocks. In the page mapping technique, the logical sector is used in sequence regardless the page location if there are write requests. As the page mapping technique allocates a new block after using a block page that is being used for write requests, its flash memory utilization is significantly high. However, the memory use amount increases because the whole pages of the flash memory are controlled using the page mapping table [9, 10].

In the block mapping technique, the block size of the flash memory is matched to the size of the address mapping table. The logical sector numbers consist of the logical block numbers of the flash memory and the page offset in the blocks. Each logical block is mapped to one physical block of the flash memory and the page offset indicates the relevant page location in the physical block. As a logical block consists of a series of logical sector numbers, it is appropriate for the management of sequential read and write requests.

In the block mapping technique, a logical sector number can refer to a given page location in a physical block and the size of the address mapping table is proportional to the number of the whole physical blocks in the flash memory. Therefore, the address mapping table can be configured with less memory compared to the page mapping technique. However, because there remain some unused pages in the physical blocks due to the fixed location to logical address in the physical blocks, the usage efficiency of the flash memory is low.

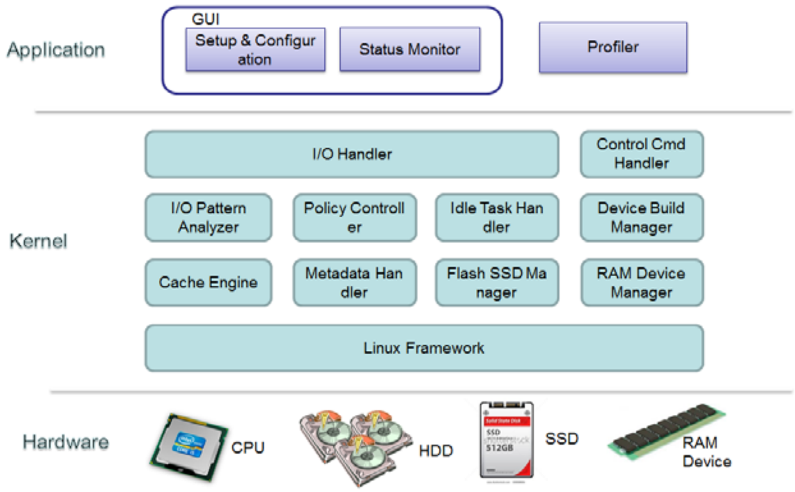

Figure 3. x86 server diagram with I/O accelerator

2.3. x-86 server with SSD-based I/O accelerator

It is difficult for the x-86 server based on HDD for cloud computing to show the necessary performance for cloud service due to the lack of I/O performance. This study suggests a x-86 server system with I/O accelerator using SSD. This x86 server, which has the IO acceleration program, can accelerate the performances of both local disk and external storage. The x86 server can be roughly divided into hardware, kernel, application program. Similar to normal servers, the hardware consists of CPU, HDD, RAM device, and SSD device. The kernel includes the cache engine module that accelerates IO. In addition, the application program consists of the programs that provide a GUI environment for configuration, creation, deletion, management, and monitoring of cache domain in order for users to easily use the cache engine module function.

3. Experiment and Consideration

3.1. Configuration of the experimental system

To confirm the IOPS performance of a general-purpose x-86 server and a x-86 cache application server, an experimental system was configured. The specification of the system is shown in Table 1.

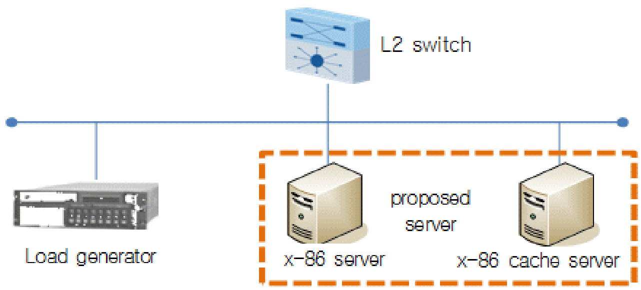

Figure 4. Test environment

Table 1. Testing system

Property	*x-86 server*	*x-86 cache server*
CPU	Intel Xeon E5-2600, 16core	Intel Xeon E5-2600, 16core
MEM	64GB	64GB
LAN	2x1Gbps ports, 2x10Gbps ports	2x1Gbps ports, 2x10Gbps ports
DISK	4 x 300GB	**SSD : 2x128, 2x200GB**, SATA3 : 4 x 300GB
FC	8Gbps FC Dual port	8Gbps FC Dual port
OS	RHEL 6.3	RHEL 6.3

Benchmark Factory is a tool to check TPS (Transaction per Second, TPC-B, TPC-C) data processing and response time, while Vdbench is to test and verify the random read/write performance. The scenario and configuration for the test is as follows: Benchmark Factory simulated 100 concurrent users and checked the TPS (TPC-B, TPC-C) and response time. Vdbench randomly checked the read/write ratio for a block size of 4k. Next, it accessed to the test server and check its performance in full load operating condition.

Here the TPC is a representative benchmark, having its strength in measuring the OLTP server's performance. The TPC versions include TPC-C, TPC-H, and the recently released TPC-E which was released as a follow-up model of TPC-C in 2006 in order to solve the problems presented in TPC-C released in the mid-1980s. As TPC-C was not completely out of the market due to complex market environment, it coexists with TPC-E.

3.2. Consideration on experimental results

The experimental results for TPC-B are shown in Table 2. This study confirmed that the performance (TPS and the average response time of work unit) of the cache application server improved about 4 times than the general-purpose x-86 server. The response time refers to the average response time to process one transaction

Table 2. TPC-B testing report using benchmark factory

Property		*Performance comparison (cache:x-86)*	*x-86 server*	*Cache server*
TPS[4)]	1st	**4.0**	1957.8	7778.9
	2nd	**4.1**	1942.7	7878.0
Response time[5)](msec)	1st	**4.3**	51.0	12.0
	2nd	**4.3**	51.0	12.0

Table 3 shows the experimental results for TPC-C. This study confirmed that the performance of the server applied I/O accelerator improved about 26 times than a normal x-86 server.

Table 3. TPC-C testing report using benchmark factory

Property		*Performance comparison (cache:x-86)*	*x-86 server*	*Cache server*
TPS	1st	**26.1**	110.7	2890.4
	2nd	**26.3**	108.4	2845.9
Response time (msec)	1st	**30.9**	865.0	28.0
	2nd	**31.6**	884.0	28.0

Table 4 shows the experimental results for 100 % Read IOPS. The IOPS, transfer rate, and response time of the cache server improved about 25 times than those of the x-86 server.

Table 4. 100% read test report using Vdbench

Property		*Performance comparison (cache:x-86)*		*x-86 server*		*Cache server*	
		Max.	*Avg.*	*Max.*	*Avg.*	*Max.*	*Avg.*
IOPS	1st	**26.2**	**26.1**	4324.3	4286.3	113271.3	111815.8
	2nd	**26.2**	**26.1**	4326.0	4292.1	113169.1	111911.6
Transfer rate (MBps)	1st	**26.2**	**25.8**	16.9	16.8	442.5	432.9
	2nd	**26.1**	**25.7**	16.9	16.8	440.8	431.8
Response time (msec)	1st	**25.6**	**26.1**	30.4	29.7	1.2	1.1
	2nd	**26.1**	**26.1**	30.1	29.7	1.2	1.1

Table 5 shows the experimental results for 100% Write IOPS. The IOPS, transfer rate, and response time of the cache server improved about 30 times than those of the x-86 server.

Table 5. 100% write test report using Vdbench

Property		*Performance comparison (cache:x-86)*		*x-86 server*		*Cache server*	
		Max.	*Avg.*	*Max.*	*Avg.*	*Max.*	*Avg.*
IOPS	1st	**36.6**	**35.2**	625.0	610.2	22883.0	21479.4
	2nd	**33.7**	**31.0**	630.2	617.4	21232.2	19160.6
Transfer rate (MBps)	1st	**37.1**	**36.5**	2.4	2.3	89.1	83.9
	2nd	**33.2**	**32.8**	2.5	2.4	82.9	78.6
Response time	1st	**32.8**	**35.1**	216.6	207.2	6.6	5.9

(msec)	2nd	**26.4**	**30.1**	206.2	204.8	7.8	6.8

Lastly, the experimental results for 50% Read and 50% Write IOPS is presented in Table 6. The IOPS, transfer rate, and response time of the cache server improved about 5 times than those of the x-86 server.

Table 6. 50% read and 50% write test report using Vdbench

Property		***Performance comparison (cache:x-86)***		***x-86 server***		***Cache server***	
		Max.	***Avg.***	***Max.***	***Avg.***	***Max.***	***Avg.***
IOPS	1st	**5.1**	**5.1**	1034.8	1015.2	5268.4	5192.8
	2nd	**5.2**	**5.2**	1034.6	1019.4	5365.8	5335.4
Transfer rate (MBps)	1st	**5.2**	**5.2**	4.0	3.9	20.6	20.2
	2nd	**5.1**	**5.3**	4.1	3.9	21.0	20.7
Response time (msec)	1st	**5.2**	**5.1**	127.9	125.4	24.8	24.6
	2nd	**5.3**	**5.2**	128.5	124.2	24.2	23.8

5. Conclusion

This study designed the x-86 server with I/O accelerator and conducted a comparative analysis of its performance with that of general x-86 servers. The analysis results showed that the performance of the x-86 server system with SSD I/O accelerator improved 5.3 times in terms of transfer rate and 32 times in terms of IOPS, compared to the HDD system. Through these results, a 4 times (⇔ 5 times×0.82) performance improvement is expected if the existing general server, IBM p570 DB Server, to a x-86 cache server. If a department operates 4 WEB servers and 5 WAS servers during an increased workload period, it is expected through the application of the x-86 cache server to use only 1 WEB server and 2 WAS servers. Further research on the optimization through effective data structures and cache algorithms is needed using DRAM cache as well as Flash memory SSD cache.
is also needed.

6. References

[1] IDC, The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011, Mar. 2008.

[2] S.W. Lee, B. Moon, and C. Park, “Advances in flash memory SSD technology for enterprise database applications”, Proceedings of SIGMOD, 2009.

[3] Binny S. Gill, Dharmendra, and S. Modha., “WOW:Wise Ordering for Writes – combining Spatial and Temporal Locality in Non-Volatile Caches”, 2005 4th USENIX Conference on File and Storage Technologies, CA, USA, 2005, Vol. 4, pp. 10-10.

[4] Intel Corporation, “OLPT performance comparison: Solid state drives vs. hard disk drives”, Test report, Jan. 2009.

[5] Mtron Co., “Solid State Drive MSP-SATA7035 Specification”, 2007.

[6] Malladi, Denis Quan, Linda Legregni, and Harold Hall, “Cloud Computing”, http://www.ibm.com/developerswork/websphere/zones/hipods/library.html, pp. 4-4, Aug. 2007.

[7] Kandukuri, B.R., “Cloud Security Issues”, Services Computing, 2009. SCC ‘09, IEEE International Conference, 21-25 Sept., Bangalore, 2009, pp. 517-520.

[8] Government Cloud Strategy A sub strategy of the Government ICT Strategy, March 2011.

[9] Bill Claybrook, “Differences explained: Private vs. public vs. hybrid cloud computing”, Search Cloud Computing.com E-Guide.

[10] Jeffrey Shafer, “I/O Virtualization Bottlenecks in Cloud Computing Today”, WIOV'10 Proceedings of the 2nd conference on I/O virtualization, CA, USA, 2010, pp. 5-5.