An SSD-HDD Integrated Storage Architecture for Write-Once-Read-Once Applications on Clusters

Cailiang Xu, Wei Wang, Deng Zhou, and Tao Xie
{*cxu, wang, zhoud*}*@rohan.sdsu.edu, txie@mail.sdsu.edu*
San Diego State University, 5500 Campanile Drive
San Diego, CA 92182, USA

***Abstract*—After investigating the data processing characteristics of several scientific applications in various disciplines from bioinformatics to geology, we discover that they share one common feature: raw data is written once onto a storage system and then it is read into memory once for analyzing, after which it will seldom be used in the future. Typically, these scientific applications are running on a cluster where the storage system of each node is composed of an array of hard disk drives (HDDs). Although HDDs are economical, they become increasingly incompetent to meet the high I/O performance requirements imposed by these applications. Flash memory based solid-state-drives (SSDs), on the other hand, can provide a high performance and energy-efficiency. Still, they are relatively expensive than HDDs. In this paper, we propose a cost-effective yet high-performance storage architecture called SOHO (SSD-Workshop-HDD-Warehouse) for these write-once-read-once scientific applications like seismic wave analysis. Its basic idea is to process raw data in the workshop (i.e., SSD), and then, the processed data is moved to the warehouse (i.e., HDD) later. Experiments using both real-world scientific applications and synthetic traces demonstrate that on average SOHO outperforms a pure HDD storage system in mean response time by 78.25%. Compared to a pure SSD system, it only degrades mean response time by less than 3.11%.**

***Keywords*-solid-state-drive; hybrid storage architecture; scientific applications; write-once-read-once; cluster.**

I. Introduction

Batch-processing of big data in real-time or near real-time has steadily become indispensable due to the possibilities given by the emerging hardware techniques and the requirements of scientific research such as seismic wave analysis and gene sequences analysis. However, one of the most challenging is how to efficiently store and process the data with very large sizes. Due to the cost-effectiveness of traditional storage media, HDDs are still the dominant secondary storage devices to offer high capacity for scientific applications. Unfortunately, the performance of big data analyzing is impeded by HDDs because of their low I/O performance [3].

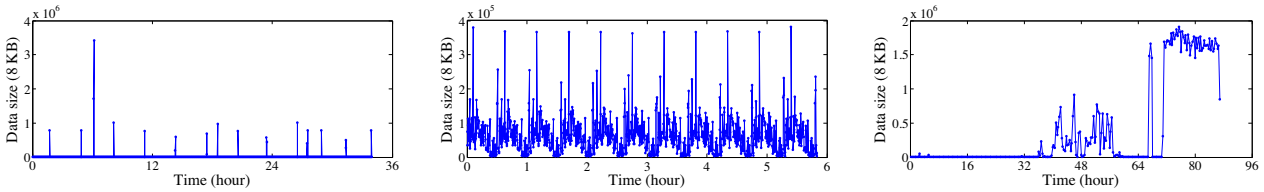
Recently, NAND flash memory based solid state disk (hereafter, SSD) has been introduced as a secondary storage device for a range of applications from servers to clusters [2] due to its appealing characteristics such as energy-efficiency, ruggedness, and capacity [5]. However, the dollar per gigabyte of SSDs is still significantly higher than that of HDDs. Thus, using SSDs as a warehouse for the only purpose of storing huge amounts of data remains a luxury choice in a current computing infrastructure.

We collaborate with certain domain scientists and find that their scientific applications share some common features in data processing. First of all, the amount of raw data is always massive and their data processing is intensive. Secondly, raw data analyzing is a batch-processing job in a sequential way. Lastly, most of the data is usually only processed once (see section III). The scientific applications with such properties in data processing are called write-once-read-once applications.

To better serve the I/O needs of these applications while controlling the cost, in this paper we design and implement a novel storage architecture called SOHO (SSD-Workshop-HDD-Warehouse) to deliver a near pure-SSD I/O performance without largely increasing the overall cost of storage system in clusters. The rationale behind is to exploit the complementary merits of HDDs and SSDs. In the SOHO architecture, an SSD and an HDD are integrated into a hybrid array, which is referred as a SOHO module in this paper (see Figure 2). The SSD is served as a workshop (or called factory) where raw data is stored and processed, whereas the HDD only offers a huge capacity as a warehouse for storing processed data. The SOHO architecture has the following advantages: by utilizing a massive capacity HDD and a moderate size SSD, it noticeable reduces stoage system cost in terms of dollar per gigabyte compared with a pure SSD storage system. From the performance point of view, SOHO delivers a very similar I/O performance to that of a pure SSD storage architecture assisted by the intelligent data management scheme (see Section V).

The remaining of this paper is organized as follows. Section II illustrates three typical scientific applications. Section III presents the design and implementation of SOHO including the data management scheme. Experimental results and analysis are discussed in Section IV. The last section concludes this paper and points out the future work.

978-1-4673-6598-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CLUSTER.2015.20



(a) DNA sequences analysis. (b) Seismic wave analysis. (c) Geographic information analysis.

Figure 1. Write request distributions of the three scientific applications.

II. Scientific Applications and Their I/O Traces

DNA Sequences Analysis (DSA): DNA sequencing and analyzing require a high-performance computing platform and a high-capacity storage system due to the huge amount of genomes in one DNA sequence. The typical size of input data of the DNA sequences ranges from several megabytes to tens of gigabytes. The size of output files is proportional to that of the input data.

Seismic Wave Analysis (SWA): Seismic wave analysis in geology is another example of batch-processing scientific applications. In geologists' research, waveform data are downloaded from selected data stations to a local cluster. The size of data archives varies from several gigabytes to hundreds of gigabytes. Normally, the size of one batch output seismograms is just several megabytes, which is much smaller than that of the raw waveform data. However, in order to effectively conduct seismic wave research, geologists generally need hundreds of batchs of seismograms. Thus, the size of the data generated could reach more than ten gigabytes. The raw seismic wave data that have been analyzed would seldom be used again in the future.

Geographic Information Analysis (GIA): In geography domain, geographers also need to process big data to solve some critical problems such as the impacts of anthropogenic activities on global climate change. Usually, different sizes of geographic data are gathered from satellites or a huge amount of remote sensors. The size of the raw data ranges from tens of kilobytes to tens of gigabytes. Data analysis is preformed on a cluster.

Figure 1 illustrates the write request distributioins of the three scientific applications. We can clearly see that during most time write activities of the three applications are non-intensive. A large amount of data is written into storage only in bursts either periodically (Figure 1a and Figure 1b) or randomly (Figure 1c). For DSA, write spikes are almost evenly distributed and their data sizes are very similar with one exception. SWA write distribution exhibits a high regularity. Its disk I/O behiavor is predictable, which makes the disk management more efficient. The GIA disk activities manifest the most wildness as write spikes occur irregularly. Moreover, the heights of spikes vary dramatically. Due to the space limit, we only provide write distributions.

We found that domain scientists usually analyze data in the following steps: (1) download the huge size of raw data onto a cluster; (2) read the data into memory in batches for analyzing; and (3) export the analysis results to the storage system. In short, the huge size raw data are written into storage once, and then, they are read into memory once for analyzing. At last, the generated results will be written back to storage and the raw data will be seldom used in the future. From the angle of disk I/O, the three scientific applications share one common feature in data processing: write-once-read-one.

III. Design and Implementation

SOHO is designed as a standard storage system module for write-once-read-once scientific applications. For every cluster various number of SOHOs can be organized together to deliver different performance. Each single SOHO module is capable to provide a similar I/O performance compared with a pure SSD in write-once-read-once scientific applications and only needs an HDD-level cost. Our goal is to provide a storage module that can be easily integrated into a cluster, which is designed to provide a high performance in big data applications. Fig. 2 illustrates the architecture of a cluster and the organization of a SOHO module. Each node of the cluster is comprised of one central process unit and a RAID controller. SOHOs are connected to RAID controller directly through the standard SATA interface. Each SOHO is comprised of one SSD with small capacity and one larger capacity HDD. In our experiments, we use a 32 GB SSD and a 128 GB HDD.

To achieve the high performance in write-once-read-once scientific applications with SOHO, a dedicated data management scheme is developed.Three key data structures are designed in the scheme: address mapping table (AMT), SSD usage recorder (SUR), and HDD usage recorder (HUR).AMT records the mappings between external logical addresses and SSD/HDD logical addresses. SUR reflects the mapping relationship between SSD logical address and external logical address (ELA). HUR has the similar function with SUR and reflects the usage of HDD.

IV. Performance Evaluation

A. Experimental Environment

To evaluate our SOHO storage module, we largely extend a validated open-source storage simulator named Microsoft

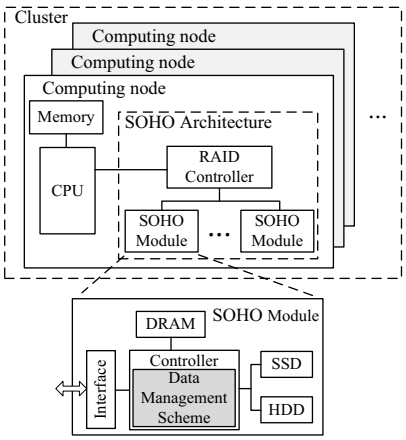


Figure 2. The architecture of SOHO.

Table I
SIMULATION PARAMETERS

Parameter	Value Fixed - (varied)
Configuration of SSD	
Page Size (KB)	4
Pages per Block	64
Blocks per Plane	8,192
Planes per Die	4
Dies per Package	4
SSD Capacity (GB)	32 - (8, 16, 32)
Block Erase Time (μs)	2,000
Page Write Time (μs)	200
Page Read Time (μs)	20
Critical threshold	0.8 - (0.2, 0.4, 0.8)
Regular threshold	0.3
Configuration of HDD	
Number of data surfaces	8
Number of cylinders	48,624
Number of Blocks	286,749,479
Head switch time (ms)	0.165
Rotation speed (in rpms)	10,017
Total Capacity (GB)	128

SSD model [5], which is built on DiskSim 4.0 [1]. Disksim provides the main functions of extended simulator and SSD model simulates the single SSD in the SOHO architecture.

Our experiments are carried out on a Dell PowerEdge 1900 server with two Quad Core Inter Xeon E5310 1.60 GHz processors and 8GB memory. The operating system is Linux Ubuntu 11.10 with Kernel 3.0.0-17. Table I shows the hardware configuration of SSD and HDD.

B. Overall Performance Analysis

Figure 3 illustrates the performance of three different storage architectures using default configuration. The total capacity of each storage system is 160 GB. In SOHO the capacity ratio of SSD to HDD is 0.25. It is clear that for all

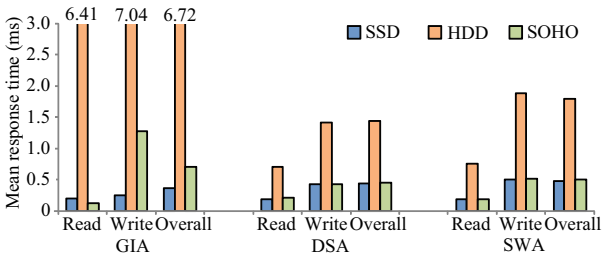


Figure 3. Performance of three different storage architectures.

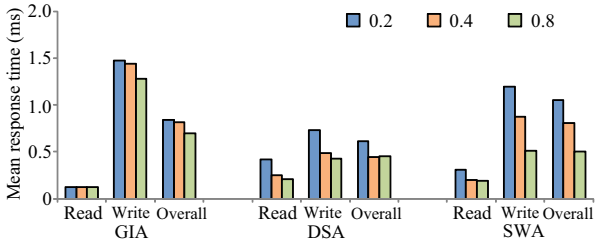


Figure 4. Performance of SOHO in different critical thresholds.

the three applications SSD constantly outperforms the other two storage systems. On average, the performance of pure SSD storage system is 79.1% better than that of the pure HDD storage system. In the DSA and SWA applications, SOHO delivers a very similar performance compared with the pure SSD storage system that its mean response time increases less than 3%. However, in GIA applications the write performance of SOHO is worse than that of pure SSD. The reason behind it is that the GIA application is write dominant and the average size of request is 26.9% and 64.9% larger than SWA and DSA applications, respectively.

C. Critical Threshold

When the size of data stored in the SSD reaches the critical threshold in SOHO, a data moving process is invoked. Figure 4 illustrates the mean response time under the three different settings. It is obvious that the mean response time decreases while the value of critical threshold increases. The overall performance improvement between 0.2 threshold and 0.8 threshold is 15.6% on average. In the GIA application, the size of one batch raw data is around 16GB. Hence, the number of data moving processes under 0.2 and 0.4 threshold is 2 and 1, respectively. During each data moving procee the storage system is blocked so that the overall mean response time is increased. In the 0.8 threshold situation, on the contrary, no data moving process is performed during responding the batch requests.

D. Average Request Size

In this section, the performance of three different storage systems under various average request sizes is studied. Three synthetic traces with average request size of 4 KB, 40 KB,

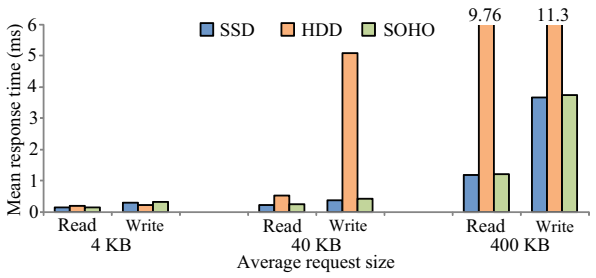


Figure 5. Performance of SOHO with different average request sizes.

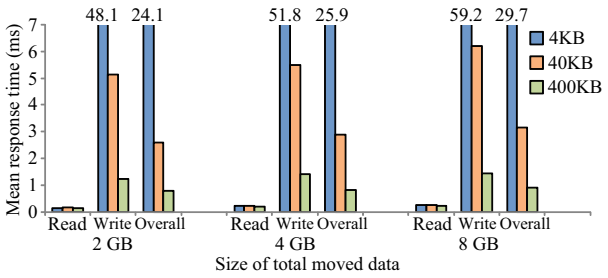


Figure 6. Performance of SOHO with different moving block sizes.

and 400 KB were generated. It is clear that mean respones time of three traces with different data sizes varies largely between pure SSD and pure HDD systems. Figure 5 shows the trend of performance when average request size changes. For read performance, we find that the mean response time of HDD in the 40 KB situation is only a little larger than that in the 4 KB situation. While in the 400 KB situation, the mean response time of HDD largely increases compared with the other two situations. The same trend exists in pure SSD and SOHO. The reason is that in SOHO design the internal data block is 4 KB, a large request will be divided into multiple sub-requests and finally increases the overall workload of SOHO. For write request, the same trend is illustrated. However, in 400 KB situation, the performance of SSD and SOHO decreases largely compared with 40 KB situation. On average, mean response time decreases 91.7%. From the figure, we can also find that even under the toughest situation, the SOHO still can deliver a very similar performance compared with pure SSD storage media.

E. Block Size in Data Moving Process

During data moving process, program running in SOHO is responsible for moving data from SSD to HDD. Since the data is moved block-by-block, different size of block will result in different times of move in each moving process.We change the block size during moving process from 4 KB to 400 KB and test the performance under different total size of moving data, which ranges from 2 GB to 8 GB. Figure 6 shows the result. From the figure, we can see that the algorithm with larger block size outperforms the other two consistently. Meanwhile, differences between 40 KB block size and 400 KB block size under three workloads of moving processes are small. On the contrary, the performance of algorithm with 4 KB block size decreases 16.7% when workload increases from 2 GB to 8GB.

V. Conclusions

Write-once-read-once scientific appliations running on clusters normally process huge amounts of data in a batch manner, which demands a high performance storage system for each computing node. Although several studies on SSD-HDD hybrid storage systems have been reported in the literature [4], none of them takes the special I/O needs of these scientific applications into accout. Besides, they generally split the I/O workloads between SSD and HDD, which is not suitable for write-once-read-once scientific applications. This is because HDDs are incompetent to provide a sufficient I/O performance. In this paper, we examine the I/O characteristics of three scientific applications. Based on our discoveries, we develop a novel storage architecture called SOHO to boost the I/O performance for these applications. It uses an SSD as a workshop or factory to process large size of raw data. Eventually, it stores analyzed results and processed data onto an HDD, which is utilized as a warehouse. Although the idea of SOHO is straightforward, experimental results from the three real applications demonstrate its effectiveness. On average SOHO improves overall I/O performance by 78.25% compared to a pure HDD storage system while delivering a very similar performance to a pure SSD storage.

Acknowledgments

We would like to thank our science collaborators Rob Edwards, Dave McKinsey, and Shou Ma at the San Diego State University for sharing us their scientific appliations data. This work was supported in part by the National Science Foundation under grant CNS (CAREER)-0845105.

References

[1] J.S. Bucy, S. Jiri, S.W. Schlosser, and G.R. Ganger, "The disksim simulation environment version 4.0 reference manual (cmu-pdl-08-101)," Parallel Data Laboratory (2008): 26.

[2] J. He, and A. Jagatheesan et. al, "DASH: a recipe for a flash-based data intensive supercomputer," In Proc. ACM/IEEE SC, New Orleans, LA, 2010.

[3] J. Hsieh, T. Kuo and P. Wu, "Energy-Efficient and Performance-Enhanced Disks Using Flash-Memory Cache," In Proc. ISLPED, pp. 334-339, 2007.

[4] J. Matthews, and S. Trika et. al, "Intel R turbo memory: Non-volatile disk caches in the storage hierarchy of mainstream computer systems," TOS, 4(2):124, 2008.

[5] W. Wang, T. Xie, and D. Zhou. "Understanding the impact of threshold voltage on mlc flash memory performance and reliability," In Proc. of the 28th ACM ICS, pp. 201-210, 2014.