SynergyFS: A Stackable File System Creating Synergies between Heterogeneous Storage Devices

Keun Soo Yim and Jae C. Son
Samsung Advanced Institute of Technology
{keunsoo.yim, jcson}@samsung.com

Abstract

Hybrid storage architecture is one efficient method that can optimize the I/O performance, cost, and power consumption of storage systems. Thanks to the advances in semiconductor and optical storage technology, its application area is being expanded. It tries to store data to the most proper medium by considering I/O locality of the data. Data management between heterogeneous storage media is important, but it was manually done by system users.

This paper presents an automatic management technique for a hybrid storage architecture. A novel software layer is defined in the kernel between virtual and physical file systems. The proposed layer is a variant of stackable file systems, but is able to move files between heterogeneous physical file systems. For example, by utilizing the semantic information (e.g., file type and owner process), the proposed system optimizes the I/O performance without any manual control. Also as the proposed system concatenates the storage space of physical file systems, its space overhead is negligible. Specific characteristics of the proposed systems are analyzed through performance evaluation.

1 Motivation

The primary design objectives of storage systems include high performance, low cost, and low power consumption. In practice there exists no single storage device satisfying this design requirement. The faster the access speed of storage device is, the higher the cost per bit ratio. Despite the efforts toward an ideal storage device, we now have storage devices, partially ideal and biased to the requirement of specific applications. This stresses the importance of hybrid storage architecture because this can utilize the strong points of storage devices and hide the weaknesses in an efficient manner.

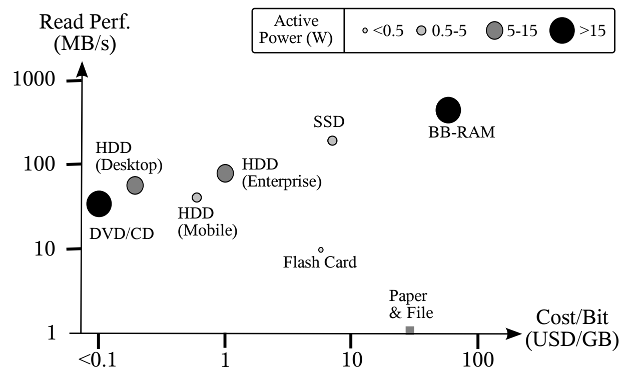


Figure 1: Characteristics of modern storage devices

2 Heterogeneous Storage Devices

The advances in semiconductor and optical storage technology are magnifying the efficiency of hybrid storage architecture. Figure 1 shows the spectrum. Semiconductor storage devices (e.g., solid-state disk, flash card, and battery-backed DRAM) are being used as a mass storage medium [1]. This is because the price of NAND flash becomes cheaper than the price of paper and film, often considered as an entry point of mass storage, and that of DRAM gets closer to this barrier. Also, optical storage (e.g., CD/DVD+RW) can be used for a mutable storage solution thanks to its rewrite support. Not only that, there are several different types of HDD and each has a different characteristic that the hybrid storage system can take advantage of.

The advent of novel storage technology facilitates the use of hybrid storage systems, especially in desktops and entry-level servers where Linux is popular. Let us assume that a device A has x times higher I/O operations per second (IOPS) than device B. If frequently accessed data are stored in A, both I/O performance and the ensuing user experience will be improved. The high cost-per-bit ratio of A can be compensated by its performance, as device A can replace x number of device B

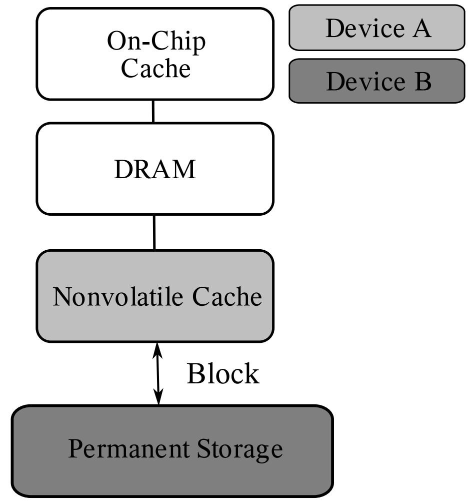
in terms of IOPS. Also the energy consumption can be optimized. If *A* and *B* are used together and *A* absorbs a significant portion of I/O traffic, *B* faces a longer idle period, which gives more chances to the power management technique applied in *B*. If *A* consumes less energy than the energy saving of *B*, the overall energy consumption is lowered.

3 Previous Management Techniques

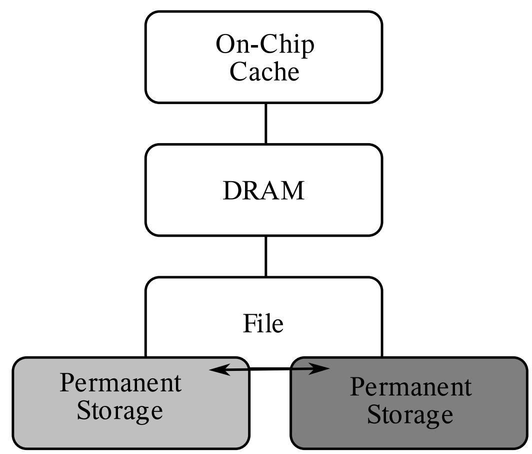
The efficiency of hybrid storage depends heavily on the I/O locality of the I/O pattern of the storage systems, and how this locality is exploited when placing data between heterogeneous storage devices. Strong locality is typically found in desktops and servers, as they are controlled by humans and human behavior causes this. For example, a file used in yesterday is likely to be used today, implying temporal locality, and a software package has its own working directory where most of its I/O operations are done, producing spatial locality. By storing files having strong locality to a faster device, the overall performance, cost, and energy consumption of a hybrid storage system can be optimized.

Until recent years, data placement between heterogeneous storage media was explicitly controlled by computer administrators. For example, a software package is installed to a fast medium if its expected usage frequency is high. Similarly the administrator stores large multimedia files to a slow but cost-efficient medium. Conventional users are not good at placing and migrating files. Even when they are good at this, automatic management is still desired, as it is more convenient. Automatic management techniques shall abstract the address spaces of heterogeneous storage devices in order to provide a unified view of them to applications and users.

In the previous automatic management techniques, abstraction was done by using the memory hierarchy shown in Figure 2(a). Specifically, a faster medium is used as a nonvolatile cache memory, and block-level management technique is used to place data into nonvolatile memory and evict the data to a permanent storage device. Both in Vista [2] and Solaris ZFS [3], NAND flash storage is utilized as a buffer memory of HDD. Vista identifies blocks used for booting and application launching, and the identified blocks are kept in the flash storage. In ZFS, two SSD devices are used as a write buffer and the storage of file system log data, respectively.



(a) Previous Architecture



(b) Proposed Architecture

Figure 2: Hybrid storage architectures

4 Proposed Synergy File System

This paper presents the design and analysis of synergy file system (SynergyFS) that can manage hybrid storage architecture in an efficient manner. The following is a summary of the key features of the proposed technique. Because of this, SynergyFS is able to create synergies between heterogeneous file systems and also heterogeneous storage devices.

First, the memory hierarchy assumed by SynergyFS is shown in Figure 2(b). Both faster medium and cheaper medium are used as a permanent storage. Each medium has its own physical file system that is particularly optimized for its I/O characteristics. This is helpful, espe-

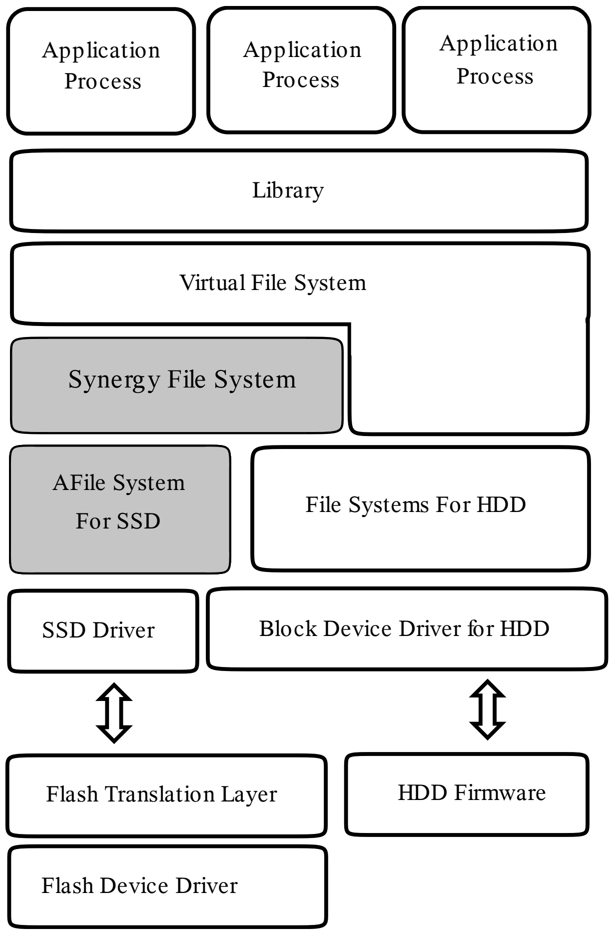


Figure 3: The proposed software architecture

cially in novel storage devices having distinct I/O characteristics.

Second, SynergyFS is a descendant of the stackable file system; thus it unifies the multiple physical file systems it manages. Figure 3 shows the software stack. It has ability to place and move files between the physical file systems. This ability is not available in the existing stackable file systems, as they were not designed for this purpose. Since SynergyFS is an intermediate software layer between virtual and physical file systems, modification of existing kernel modules is not required.

Third, the proposed SynergyFS supports file-level data placement and migration, and this enables the utilization of semantic information (e.g., file type and owner process) that were not accessible in the previous block-level management techniques. It decides where a file will be stored in by taking into account the locality observed in the access pattern of file. On the other hand, the block-level management techniques need a large size of memory to maintain the mapping information and a relatively long lookup time is involved in their I/O operations. Also the block-level management is not good at exploiting the spatial locality. For example, a file can be spread all over the storage address space, but it is difficult for the block-level technique to know which physical blocks belong to which file.

Fourth, SynergyFS concatenates the storage space of physical file systems and thus its space overhead is negligible. However, in the previous techniques, the non-volatile cache capacity is not visible to users and is not added up to the whole storage capacity. As the non-volatile cache size becomes increasingly larger, the proposed scheme is more efficient in terms of user-visible storage capacity.

5 Effectiveness Evaluations

In order to evaluate the effectiveness of hybrid storage architecture, we have analyzed the user experience (UX) of computer systems using an HDD, an SSD, and a device combining these two. The UX affected by storage device includes cost per bit ratio, boot-up time, file system I/O performance, power consumption, and storage device portability. Figure 4 shows the evaluation results of these factors where a generic Linux-based PC system was used.

First, in terms of cost per bit ratio both HDD and SSD/HDD hybrid drive are better than SSD. Currently an SLC(Single-Level Cell) SSD is about five times more expensive than a 2.5-inch HDD having the same capacity. As the price of NAND flash memory is being dropped quickly and high-density NAND flash memory technologies (e.g., Multi-Level Cell) are being commercialized, the price gap between HDD and SSD will be

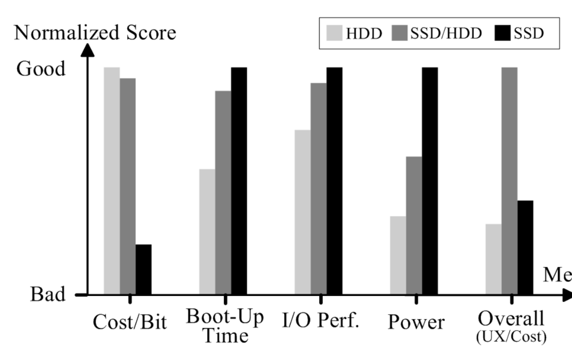


Figure 4: Performance evaluation results

alleviated. In this analysis, the cost per bit ratio of the hybrid drive is about four percent higher than that of HDD. This gap in fact depends on the ratio of SSD and HDD storage capacities used in the hybrid drive. The hybrid drive used in this analysis had a 16GB SSD and a 320GB HDD. SSD capacity was limited to 16GB because this capacity is sufficient to hold all files related to operating system and basic application programs. This shows that with only a small amount of additional cost, hybrid storage devices can be realized in practice.

Second, for the performance evaluation, the system boot-up time and file system I/O performance were analyzed. Application launching time was not taken into accounted as it can be predicted by using the boot-up time analysis result. SSD provides superb I/O performances as compared to HDD thanks to its internal I/O parallelism. The performance of the hybrid drive is comparable to that of SSD, and this shows the effectiveness of the proposed SynergyFS. Specifically, the hybrid drive had a short boot-up time because most files required for boot-up were stored in the SSD part. This drive was also able to provide high-performance file I/Os as the SSD part maintained files that were frequently-accessed. The hybrid drive used in this experiment [4] has a dedicated data path between SSD and HDD. This is helpful at reducing the time required for copying data from one device to the other and the I/O traffics observed in the host system I/O bus.

Third, the energy consumption of storage device was analyzed. Unlike HDD having power-hungry mechanical arm and motor, SSD has no mechanical parts and thus it consumed significantly lower power than HDD. Even the hybrid drive consumed lower power than HDD. In the hybrid drive, most I/O operations were handled by the SSD part. This means that its HDD part had more chances to stay in either idle or stand-by power mode.

On the other hand, in terms of portability SSD is better than both HDD and the hybrid drive. Portability of storage device means the weight, volume, shock resistance, and power consumption. Although the hybrid drive uses a small size of SSD, the volume and weight of the hybrid drive is slightly larger than that of HDD. Tight-integration of HDD and SSD using system-on-chip (SoC) and multi-chip packaging (MCP) technologies can address this problem [4].

Overall effectiveness of these three storage devices is analyzed. The UX score is calculated by multiplying the normalized scores of boot-up time, file system I/O performance, and power consumption. The UX score is then divided by the normalized score of cost per bit ratio, and this is the overall score depicted in Figure 4.

These evaluation results clearly suggest the application area of each type of devices. The hybrid drive has a high potential to replace both HDD and SSD in desktop and workstation computers because it gives better user experience and the extra cost it brings out is tolerable in this market. Note that the hybrid drive consists of both HDD and SSD and thus the amount of HDDs supplied to these computers will not be reduced. Also as HDD is one of the most cost effective solution, it will be widely used in large-capacity storage clusters continuously. On the other hand, SSD is appropriate for high-performance servers where the high cost per bit ratio can be compensated by the fast I/O performance. Portable computers are another application area for SSD because these computers use relatively small size of storage device that can hide the high price of SSD.

6 Conclusion

In this paper, we reviewed the requirement of storage systems and suggested the hybrid storage architecture as a solution. An automatic management technique for the hybrid storage architecture was presented and its effectiveness was analyzed by using an SSD/HDD hybrid storage drive. The proposed SynergyFS can be effectively used for the desktop and workstation computers in order to improve the user experience of the system in a cost-efficient manner.

References

[1] Samsung Electronics, "Samsung Solid-State Drive (SSD)," `http://www.samsung.com/global/business/semiconductor/products/flash/ssd`

[2] Microsoft Corporation, "Windows PC Accelerators: Performance Technology for Windows Vista," `http://www.microsoft.com/whdc/system/sysperf/accelerator.mspx`

[3] Sun Microsystems, "ZFS: the last word in file systems," `http://www.sun.com/2004-0914/feature`

[4] Samsung Electronics, "Samsung S-Drive: Imagine Revolutionary Convergence," http://www.samsung.com

Proceedings of the Linux Symposium

Volume Two

July 23rd–26th, 2008
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*

C. Craig Ross, *Linux Symposium*

Review Committee

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*

Dirk Hohndel, *Intel*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
Matthew Wilson, *rPath*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
Eugene Teo, *Red Hat, Inc.*
Kyle McMartin, *Red Hat, Inc.*
Jake Edge, *LWN.net*
Robyn Bergeron
Dave Boutcher, *IBM*
Mats Wichmann, *Intel*