Power Consumption Characterization of Flash Memory SSD

Seungyong Shin, Dongkun Shin
School of ICE, Sungkyunkwan University
Suwon 440-746, Korea
{space13, dongkun}@skku.edu

Abstract – ***Solid state disks (SSDs) are replacing hard disk drives (HDDs) due to their high random access performance and low power consumption. Although SSDs are low power storage devices, the power consumptions of SSDs are not negligible as SSDs adopt parallel architectures. In this paper, we analyze the power and energy consumption behaviors of SSDs depending on the I/O request patterns and provide several hints on energy optimization. In addition, our analysis technique enables to extract several architectural features of target SSD which vendors do not provide to users.***

1. Introduction

Recently, flash memory solid state disks (SSDs) are replacing hard disk drives (HDDs). SSDs require no mechanical moving parts unlike HDDs. Therefore, SSDs are superior to HDDs in terms of random access performance and power consumption. In addition, SSDs can provide high sequential access performance using parallel schemes that enable multiple flash chips to be accessed simultaneously. Due to the power efficiency of SSDs, they are attractive to mobile systems and power-hungry data centers. Though there are several studies on the power consumption analysis and optimization of HDDs [1], the power efficiency of SSDs has gotten little attention from researchers.

However, as recent SSDs use more intensive parallel schemes so as to increase I/O performances and large amounts of DRAM buffers, the energy consumption of SSDs is comparable to those of HDDs. To optimize the energy consumption of SSD, we first need to characterize the power/energy consumption of SSD. In this paper, we analyze the power and energy consumption behaviors of SSDs depending on the I/O request patterns and provide several hints on energy optimization. In addition, our analysis technique enables to extract several architectural features of target SSD which vendors do not provide to users.

The rest of this paper is organized as follows. In Section 2, we describe related studies on SSD power consumption. Section 3 provides backgrounds of flash memory and SSD. We present experimental environment and power analysis results in Section 4. Section 5 summarizes the paper.

2. Related Works

Seo et al. [2] provided the first literature on power analysis of SSD. They showed the superiority of SSD over HDD in energy consumption. However, only simple access patterns are examined and no detailed analysis is provided. Park et al. [3] proposed a power consumption simulator for SSD. Though the simulator estimates the power consumption considering parallel flash chip accesses, the power model is too simple. Lee et al. [4] showed that a single SSD can outperform several HDDs comprising redundant array of independent disks (RAID) for both power consumption and I/O performance.

We analyze the power consumption of SSD with various workloads and derive several parameters affecting the power consumption.

3. Flash Memory and Solid State Disk

Flash memory has several special features unlike the traditional magnetic hard disk. The first one is its "erase-before-write" architecture. To write a data in a block, the block should be first erased. The second feature is that the unit sizes of the erase and write operations are asymmetric. While the write operation is performed by the unit of a page, the flash memory is erased by the unit of a block that is a bundle of several sequential pages. Due to these two features, special software called the flash translation layer (FTL) is required, which maps the logical page address from the host system to the physical page address in flash memory devices. Flash memory SSDs also need an embedded FTL, which executes on the SSD controller.

The address mapping schemes of the FTL can be divided into three classes depending on the mapping granularity: page-level mapping, block-level mapping, and hybrid mapping. Hybrid mapping [5] is a compromise between page-level mapping and block-level mapping. In this scheme, a small portion of physical blocks is reserved as a log buffer. While the log blocks in the log buffer use the page-level mapping scheme, the normal data blocks are handled by the block-level mapping. Therefore, data can be written to the log block by the out-of-place manner, i.e., each logical page can be written to any physical page with the log block. However, the data block has logical pages in the in-place manner, i.e., each logical page is located at the physical page with the same offset within the block. When a write request is sent to the flash memory, the data is written to a log block, and the corresponding old data in the data block is invalidated. When there is no empty space in the log buffer, the garbage collection (GC) is invoked. GC selects one of the log blocks as a victim and moves all of the valid pages in the log block into the data blocks to make free space for on-going write requests.

To enhance I/O bandwidth, current flash memory SSDs access multiple flash chips with multi-channel and multi-way architecture as shown in Fig. 1 [6]. Two flash chips using different channels can be operated independently and therefore the page transfer times (from the NAND controller to the flash chip) and page program times for different chips can overlap. To utilize such parallel architectures, sequential data are distributed across multiple flash chips. Therefore, the parallel architecture can provide a high bandwidth for sequential requests. However, random I/O performances are poor compared to sequential I/O performances. The data from host is stored in the DRAM buffer temporarily to provide high write performance.

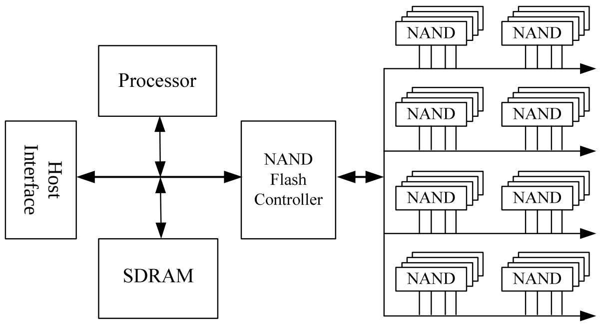


Fig. 1 The internal architecture of SSD

4. Measurement and Analysis

4.1 Experimental Setup

In our empirical study, we used two different SSDs as shown in Table 1. The performance of SSD is determined by the cell type of flash memory, DRAM buffer size and the number of parallel flash chips. NAND flash memory can be categorized into single-level-cell (SLC) type or multi-level-cell (MLC) type. The MLC flash memory provides a high density by storing multiple bits in a cell while the SLC flash memory can store only one bit. However, MLC flash has less life time and lower latency than SLC flash does. While SSD(L) is composed of SLC flash chips, SSD(H) uses MLC flash chips. However, SSD(H) provides higher I/O performances than that of SSD(L) since SSD(H) uses a larger internal buffer and a more intelligent FTL. We measured the power consumption using DAQ with the sampling rate of 250 kS/s and used the benchmark tool which is a modified version of uFlip [7]. The host file system is MS Windows NTFS.

	SSD(L)	SSD(H)
Capacity	64GB	30GB
Memory Type	SLC	MLC
Buffer size	32MB	64MB
Read	100MB/s	220MB/s
Write	80MB/s	130MB/s

Table 1: SSD specifications

To observe the effects of garbage collection on energy and power consumptions of SSD, we experimented with clean SSDs and dirty SSDs. To make the dirty SSD, we first write files up to the amount of SSD capacity and then delete all the files in file system level. Therefore, each write operation for the dirty SSD invokes a garbage collection. The clean SSD has only clean flash memory blocks, which have been erased thus data can be written at the blocks without garbage collections. We used a vendor-provided garbage collection tool or the HDDerase [8] tool which executes the secure erase command in order to erase flash blocks.

4.2 Power/Energy Analysis

We first observed the change of power consumption depending on the access patterns. Fig. 2 shows the power change graph of dirty SSD(L) for five types of access patterns, i.e., random read (RR), sequential read (SR), sequential write (SW), random write (RW) and multiple random write (MRW). Each access pattern is examined with 4 KB size I/Os and 32 KB size I/Os. The power consumption of SSD changes significantly depending on the access patterns. The idle state requires 0.6 watt, which is about 40% of the peak power consumption in the active state and is about twice the power consumption of a typical application processor (≈0.3 watt). Therefore, the idle state can consume a large portion of total power consumption and thus it is necessary to shut down SSD when it is idle. Although both SSDs have DRAM write buffers, there are precipitous increases at the moment the requests are sent to SSD. From this result, we can know that the data from host does not remain at the DRAM buffer during a long time.

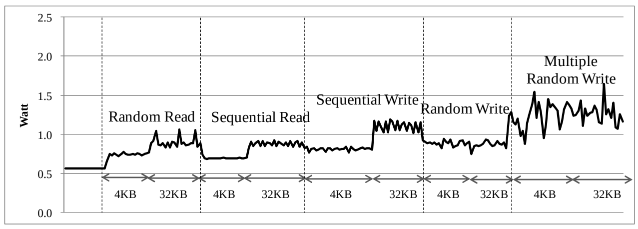


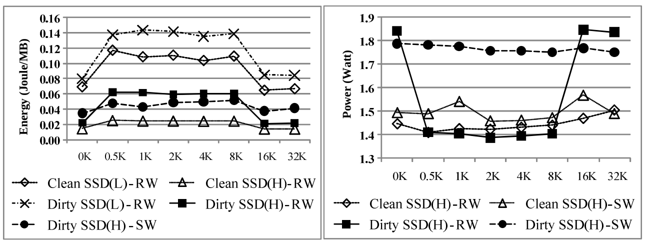
Fig. 2 Power change graph of dirty SSD(L)

We measured the energy consumption per MB (Joule/MB) and the power consumption (watt) of SSDs under four test benchmarks: alignment, granularity, parallelism and mixture. The alignment benchmark examines the effects of unaligned I/O requests, which are generated by shifting the start address of the baseline requests that have the I/O size of 32 KB and are aligned by the I/O size. Using the alignment benchmark, we can know the address mapping unit of target SSD as well as the adverse effect of unaligned requests.

The granularity benchmark generates I/O requests with different I/O sizes from 1 KB to 4 MB. The purpose of granularity benchmark is to identify the number of parallel flash chips accessed simultaneously. The parallelism benchmark generates 32 KB-sized I/O requests from parallel processes each of which accesses a different region of storage space. Finally, the mixture benchmark generates 4

KB-sized write requests by interposing RW requests between SW requests. The mixture pattern is represented by the ratio between the number of SW requests and the number of RW requests. For example, if the ratio of a mixture pattern is 1:64, 64 number of RW requests are interposed between each SW request thus it is a quiet random access pattern. The ratio of 64:1 represents that one RW request is interposed at every 64 number of SW requests thus it is a quiet sequential access pattern.

Energy and power consumptions of the alignment benchmark are shown in Fig. 3. X-axis represents the sift size. The request pattern with shift size 0 means the baseline aligned pattern. The energy consumption in the RW patterns with the shift size between 0.5 KB and 8 KB is larger than the baseline pattern. However, when the shift size is a multiple of 16 KB, the energy consumption is similar to that of the aligned baseline pattern. The SW requests show little changes depending on the shift size. This implies that SSD performs additional works if requests are not aligned by 16 KB unit. When a write request is not aligned by 16 KB, SSD reads two 16 KB units, modifies a part of them and writes them. Since such a read-modify-write operation invokes the read operation that consumes a smaller power than the write operation, the average power consumption of unaligned RW requests is smaller than that of aligned requests as shown in Fig. 3(b). However, the power consumption of clean SSD has little changes since the SSD does not require the read operation. From this result, it can be inferred that the examined SSDs use 16 KB address mapping unit, i.e., the whole 16 KB unit should be modified even when only a portion of the unit is modified.



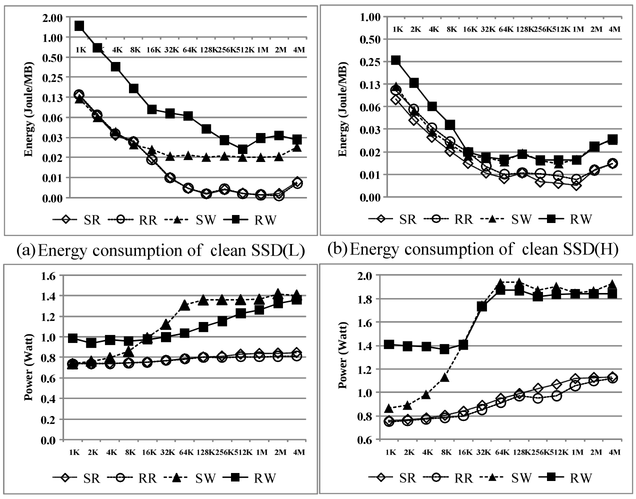
(a) Energy consumption (b) Power consumption

Fig. 3 Alignment benchmark

Fig. 4 and Fig. 5 show the results of granularity benchmark. The write requests require higher energy and power consumptions than those of read requests. RR and SR patterns show almost similar energy and power consumptions. Since SSD(H) uses MLC flash chips which require more power consumption than SLC flash chips, SSD(H) consumes higher power than SSD(L). However, the energy consumption of RW requests at SSD(L) is significantly higher than that at SSD(H). This is because SSD(L) provides lower performance for the random write requests. From this results, we can know that SSD(H) uses a more intelligent FTL algorithm which can handle RW requests efficiently.

The most outstanding change while varying the RW I/O size occurs when the I/O size is large than 16 KB. When the write request size is smaller than 16 KB, FTL should perform the read-modify-write operation and therefore the energy consumption is significantly high but the power consumption has no change. When the write request size is larger than 16 KB, both random and sequential writes increase as the I/O size increases. Especially, when the I/O size is larger than 64 KB, both SW and RW requests have little change on the power consumption in SSD(H). This means that the largest I/O size which can be handled in parallel is 64 KB and the number of parallel flash chips is 16 (= 64 KB / 4KB) since the flash page size is 4 KB. SSD(L) also have no change on the power consumption when the I/O size is larger than 64 KB. However, the power consumption of RW requests is smaller than that of SW requests since SSD(L) cannot utilize the parallel chips efficiently for random requests.

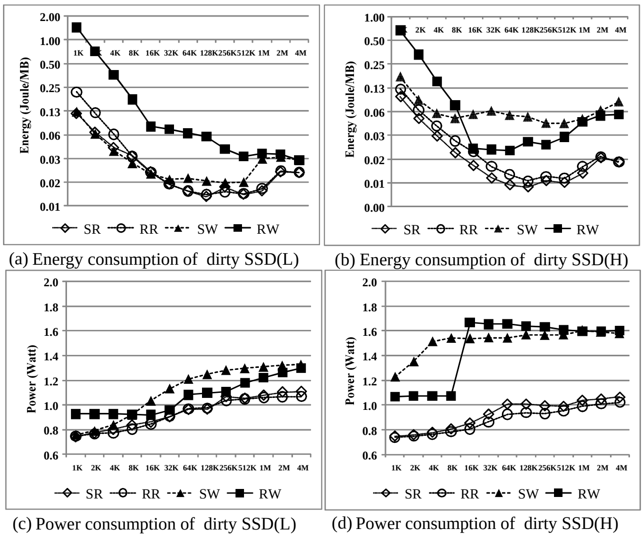
A noticeable difference on energy consumption patterns at dirty SSDs is that SW requests consume higher energy at the dirty SSD(H). In addition, the differences between power consumptions of SSD(H) depending on I/O request sizes are reduced at the dirty SSD. This is because the dirty SSD requires garbage collections. While dirty and clean SSD(L)s show large differences on energy and power consumption for read operations, SSD(H)s show large differences for write operations. For clean SSDs, SSD(H) provides better energy efficiency than SSD(L) in spite of its higher power consumption. However, the dirty SSD(H) shows worse energy efficiency than dirty SSD(L) since the MLC flash chips require high costs for garbage collection.



(a) Energy consumption of clean SSD(L) (b) Energy consumption of clean SSD(H)

(c) Power consumption of clean SSD(L) (d) Power consumption of clean SSD(H)

Fig. 4 Granularity benchmark (clean SSDs)



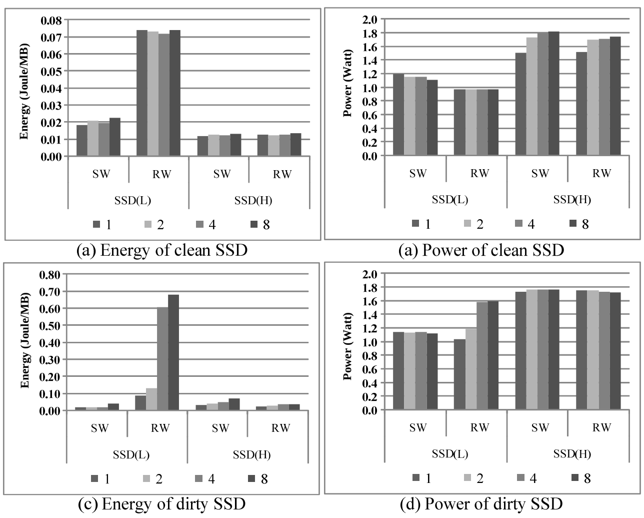
(a) Energy consumption of dirty SSD(L) (b) Energy consumption of dirty SSD(H)

(c) Power consumption of dirty SSD(L) (d) Power consumption of dirty SSD(H)

Fig. 5 Granularity benchmark (dirty SSDs)

Fig. 6 shows the results of parallelism benchmark. For the clean SSDs, there are no large changes on the energy consumption depending on the number of parallel I/O requests. However, the energy consumptions of dirty SSDs increase as the parallelism increases. Especially, the parallel RW requests on the dirty SSD(L) significantly degrade the energy efficiency since SSD(L) cannot handle the random requests efficiently. Since the garbage collections, which require erase operations, are frequently invoked for parallel RW requests on SSD(L), the energy consumption for RW requests increases due to the long latency of the erase operations.

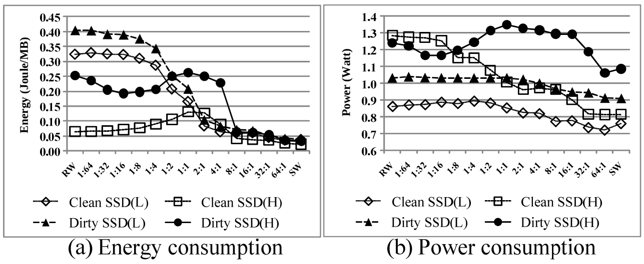
The power consumption of the clean SSD(H) increases as the number of parallel requests increases. This is because more flash chips are accessed for the parallel requests. Even for random requests, SSD(H) can fully utilize the parallel flash chips by reordering requests. However, the power consumption of clean SSD(L) shows no change since it cannot utilize the parallel flash chips efficiently.



(a) Energy of clean SSD (a) Power of clean SSD

(c) Energy of dirty SSD (d) Power of dirty SSD

Fig. 6 Parallelism benchmark



(a) Energy consumption (b) Power consumption

Fig. 7 Mixture benchmark

Fig. 7 shows the results of mixture benchmark. For this benchmark, SSD(L) and SSD(H) show completely different results. As the portion of SW increases, the energy consumption of SSD(L) decreases. However, the energy consumption of SSD(H) is highest when the portions of SW and RW are similar rather than when the access pattern is SW or RW dominant. We presume that SSD(H) uses the sequential log block (SLB) where data are written by the in-place manner. The SLB is used for efficient handling sequential write requests. However, when random and sequential write requests are mixed, the sequential log block cannot present its advantage since the interposed random requests obstruct the in-place write. However, SSD(H) can handle the random requests efficiently and thus the energy consumption gap between SSD(H) and SSD(L) is large for random-dominant request patterns.

5. Concluding Remarks

Although SSD is a low-power storage device compared to HDD, its power consumption is increasing as SSD adopts the higher degree of parallelism at I/O data path. We analyzed the energy and power consumptions of different types of SSDs for various benchmarks. We found that the energy and power consumptions are significantly different depending on the access patterns and we derived several valuable points for energy optimization. The unaligned access pattern, random access pattern with small I/O requests and random/sequential mixed access pattern can significantly degrade the energy efficiency. The internal architectures of SSD such as cell type, parallel I/O architecture and FTL algorithm also affect the energy efficiency of SSD. In addition, the dynamic power management technique, which shuts down SSD when it is idle, is critical since the idle state power is not negligible. By analyzing the power consumption of SSD, we can probe the architectural parameters of SSD such as the mapping unit, the maximum number of allowable parallel chips and the handling policy for random and sequential requests.

We have a future plan to study the effects of file systems, operating systems, and device drivers on the energy/power consumption of SSD.

References

[1] L. Benini , A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” IEEE Transactions on VLSI Systems, v.8 n.3, p.299-316, June 2000

[2] E. Seo, S. Y. Park, and B. Urgaonkar, “Empirical Analysis on Energy Efficiency of Flash-based SSDs,” Proc. of HotPower, 2008

[3] J. Park, S. You, S. Lee, and C. Park, “Power Modeling of Solid State Disk for Dynamic Power Management Policy Design in Embedded Systems,” Proc. of SEUS, pp. 24-35, 2009

[4] S. Lee, B. Moon, and C. Park, “Advanced in Flash Memory SSD Technology for Enterprising Database Applications,” Proc. of SIGMOD, 2009

[5] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho, “A space-efficient flash translation layer for compact flash systems,” IEEE Transactions on Consumer Electronics, 48(2):366--375, 2002.

[6] C. Dirik and B. Jacob, “The performance of PC solid-state disks (SSDs) as a function of bandwidth, concurrency, device architecture, and system organization,” Proc. of ISCA, pages 279–289, 2009.

[7] L. Bouganim, B. Jonsson, and P. Bonnet, “uFLIP: Understanding Flash IO Patterns,” Proc. of Intive Data Systems Research (CIDR), 2009

[8] G.F. Hughes, D.M. Commins, and T. Coughlin, “Disposal of disk and tape data by secure sanitization,” IEEE Security and Privacy, Vol. 7, No. 4, 2009, pp. 29-34.