Advances in Flash Memory SSD Technology for Enterprise Database Applications

Sang-Won Lee
School of Info & Comm Engr
Sungkyunkwan University
Suwon 440-746, Korea
wonlee@ece.skku.ac.kr

Bongki Moon
Dept. of Computer Science
University of Arizona
Tucson, AZ 85721, U.S.A.
bkmoon@cs.arizona.edu

Chanik Park
Samsung Electronics Co., Ltd.
San #16 Banwol-Ri
Hwasung-City 445-701, Korea
ci.park@samsung.com

ABSTRACT

The past few decades have witnessed a chronic and widening imbalance among processor bandwidth, disk capacity, and access speed of disk. According to Amdhal's law, the performance enhancement possible with a given improvement is limited by the amount that the improved feature is used. This implies that the performance enhancement of an OLTP system would be seriously limited without a considerable improvement in I/O throughput. Since the market debut of flash memory SSD a few years ago, we have made a continued effort to overcome its poor random write performance and to provide stable and sufficient I/O bandwidth. In this paper, we present three different flash memory SSD models prototyped recently by Samsung Electronics. We then show how the flash memory SSD technology has advanced to reverse the widening trend of performance gap between processors and storage devices. We also demonstrate that even a single flash memory drive can outperform a level-0 RAID with eight enterprise class 15k-RPM disk drives with respect to transaction throughput, cost effectiveness and energy consumption.

Categories and Subject Descriptors

H. Information Systems [**H.2 DATABASE MANAGEMENT**]: H.2.2 Physical Design

General Terms

Design, Measurement, Performance

Keywords

Flash-Memory SSD, TPC-C Benchmark, Energy

*This work was partly supported by MKE, Korea under ITRC IITA-2009-(C1090-0902-0046) and IT R&D program of IITA-2006-(S-040-03), and by KRF, Korea under KRF-2008-0641. This work was also sponsored in part by the U.S. National Science Foundation Grant IIS-0848503. The authors assume all responsibility for the contents of the paper.

SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, U.S.A.

1. INTRODUCTION

Lately there has been a substantial influx of solid state drives (SSD) based on NAND type flash memory in the storage market. Despite its superiority in access latency and energy consumption, the full-blown adoption of flash memory SSD is yet to be seen, partly because there still remain concerns about endurance, tardy random writes and price.

Since the market debut of flash memory SSD a few years ago, we have made a continued effort to overcome its poor random write performance and to provide stable and sufficient I/O bandwidth. This has been a challenging but critical mission for enterprise database applications, because write operations randomly scattered over a large data space are very common in typical OLTP workloads and high transaction throughput cannot be achieved without providing sufficient I/O throughput. Following up the previous work that suggests flash memory SSD as a storage medium for logging, temporary table spaces and rollback segments [10], this study aims at demonstrating that, with the latest advances in the SSD technology, flash memory drives have emerged as a viable option even for database table spaces and indexes where randomly scattered reads and writes are a dominant pattern of I/O.

In this paper, we present three different flash memory SSD models prototyped recently by Samsung Electronics, namely, Personal Class SSD Models A and B (PC-A and PC-B), and Enterprise Class SSD (EC). We then show how the flash memory SSD technology has advanced to reverse the widening trend of performance gap between processors and storage devices. The most recent prototype EC SSD is equipped with several notable architectural enhancements such as fat provisioning, a larger DRAM buffer, inter-command parallelism with more channels, and native command queuing.

The I/O benchmark tests carried out with public domain tools show that the access density of a storage system could increase significantly by up to several orders of magnitude by adopting the advanced flash memory drives. We believe that the magnitude of this improvement was large enough to be taken as an evidence of the *reversed trend* in the processor-storage performance gap, which is being enabled by this new SSD technology. We have also observed in the TPC-C benchmark tests that even a single Enterprise Class flash memory drive can outperform a level-0 RAID with eight enterprise class 15k-RPM disk drives in terms of transaction throughput, cost effectiveness and energy consumption. The amount of energy consumed by the EC flash memory drive

to deliver comparable peak transaction throughput was far less than ten percent of what the RAID-0 with eight disks consumed.

2. I/O CRISIS IN OLTP SYSTEMS

In the past few decades, we have witnessed a chronic and widening imbalance between the capacity and the access speed of magnetic disk drives. According to a recent survey [13], while disk capacity had increased about 2500 times during the period from 1983 to 2003, disk bandwidth and latency had improved only about 140 times and 8 times, respectively, during the same period. The imbalance between the capacity and the access speed is often measured by a metric called *access density* [6]. Access density is the ratio of the number of I/O operations per second to the capacity of a disk drive. Apparently, the access density of disk drives has steadily declined in the past few decades, and is expected to decrease even further in the future.

Amdhal's law states that the performance enhancement possible with a given improvement is limited by the amount that the improved feature is used [5]. In an OLTP system that processes a large number of small random I/O operations continually, the processor bandwidth will be only a small fraction of the 'features to be improved.' This implies that the performance enhancement of an OLTP system will be seriously limited without a considerable improvement in I/O throughput.

According to the aforementioned survey, processor bandwidth measured in MIPS had improved about 2250 times during the period from 1982 to 2001. It is noteworthy that disk capacity and processor bandwidth had improved at almost identical pace during the period of two decades. If this trend continues in the future, the access density will be a convenient metric that succinctly measures how much disk drives lag behind processors in terms of processing speed.

To close the gap between processor and I/O bandwidths, a balanced system often requires a large disk farm to exploit I/O parallelism and adopts the *short-stroking* strategy to reduce disk seek latency by using only a portion of disk capacity [4, 7]. A large-scale TPC-C system reported recently [8], for example, is equipped with approximately 170 disk drives per each of 64 processor cores. Such a balanced TPC-C system will be able to yield significantly improved transaction throughput by increasing the access density of a disk subsystem. However, it will in turn raise other concerns such as cost effectiveness and energy consumption.

The improvement of processor speed is expected to continue following Moore's law, while disk bandwidth is likely to grow at a much slower pace (by at least an order of magnitude) for the foreseeable future. Consequently, the number of disk drives a balanced system requires will continue to grow, and so will the concerns about cost effectiveness and energy consumption.

Flash Opportunities

Without a breakthrough in magnetic disk technology, disk-based storage subsystems would continue to experience the I/O crisis described above. It is mainly because the only plausible way of increasing the capacity of a disk-based storage system without sacrificing the access speed is to increase the number of spindles instead of increasing the capacity of individual disk drives. In contrast, the access speed of a flash memory drive is largely insensitive to the number of flash chips contained in a drive. Furthermore, the multi-channel architecture of flash memory SSD can even improve the access speed by exploiting parallelism available from an increased number of flash chips contained in a drive. Therefore, the capacity of a flash memory drive is expected to keep expanding without sacrificing the access speed (or without decreasing the access density). Given the steady trend of price reduction (40 percent or more annual reduction in average sale price) of NAND flash memory, the flash memory SSD devices lend themselves to being the most pragmatic solution to the I/O crisis.

3. ADVANCES IN SSD FOR OLTP

The previous work has shown that the low latency of flash memory SSD can alleviate the problems of commit time delay and the increased random reads for multi-version read consistency [10]. Without improving its random write performance considerably, however, flash memory SSD may not be considered a pragmatic replacement of magnetic disk for enterprise database applications. This is because randomly scattered write requests are common in the database table and index spaces for on-line transaction processing and similar database applications.

In this section, we present three different flash memory SSD models prototyped recently by Samsung Electronics. With these SSD prototypes, we describe how the flash memory SSD technology has advanced from personal class storage devices to enterprise class storage devices and how the random write performance has been improved up to the level adequate for enterprise database applications. The architectural differences of the SSD prototypes are discussed in this section. The design characteristics of the SSD prototypes are summarized in Table 1, and their impact on the I/O and transaction throughput will be presented in Section 4 and Section 5.

Personal Class SSD Model A (PC-A)

This is the first prototype of Samsung SSD developed for personal and mobile computing platforms. Since it was aimed at replacing commodity disk drives, its design goal was to match magnetic disk drives in sequential read/write performance. Through 4-channel parallelism and interleaving, the PC-A SSD achieved sequential read/write bandwidth at the level comparable to that of commodity disk drives. In the previous work [10], we have demonstrated that the PC-A SSD can successfully cope with the I/O bottlenecks for transaction log, rollback and temporary data, because the access patterns dominant in these data spaces (*e.g.*, writes in sequential or append-only fashion) can best utilize the superior characteristics of flash memory such as extremely low read and write latency without being penalized by the *erase-before-update* limitation of flash memory.

When it came to random writes, however, the throughput of the PC-A SSD was quite poor at about 30 IOPS, due to the erase-before-write limitation of flash memory. When the PC-A SSD was used to store database tables and indexes for a typical OLTP workload, the transaction throughput from the SSD was no better than half of what a commodity magnetic disk would yield. This was due mainly to the lack

SSD Class	Capacity	DRAM Buffer	Provisioning	# Channels	Parallelism	NCQ
Personal Class A	32 GB	< 1 MB	Thin	4	Intra-command	No
Personal Class B	64 GB	32 MB	Fat	4	Intra-command	No
Enterprise Class	128 GB	128 MB	Fat	8	Inter-command	Yes

Table 1: Design Characteristics of Personal and Enterprise Class Flash Memory SSDs

of write buffering and thin provisioning of flash storage. The PC-A SSD was not able to hide write latency for the OLTP workload, because its data space was just too large for the SSD to deal with.

Personal Class SSD Model B (PC-B)

To improve random write performance, two major enhancements were made for the second prototype of Samsung SSD. The PC-B SSD was equipped with a DRAM buffer of 32 MBytes and fat provisioning of flash storage. The DRAM buffer was added to absorb write bursts so that the number of physical writes could be reduced.[1] As a matter of fact, in our own test, we observed that a DRAM buffer of 32 MB was large enough to improve write throughput up to an order of magnitude for a relatively narrow data space of one GBytes.

As the data space grows larger, however, write buffering becomes less effective and the number of physical writes inevitably increases. To cope with this problem, fat provisioning was adopted to minimize the average latency of a page write by avoiding costly erase operations. When a portion (typically a page) of a flash block is updated, the block is assigned an extra flash block called a replacement block, and the updated portion is written to the replacement block without updating (or erasing/relocating) the original block. Since a replacement block can absorb multiple page write requests, this scheme of fat provisioning can reduce the average number of block erase operations. Considering the fact that each write could cause a block erase operation without fat provisioning, the potential performance gain by fat provisioning could be quite substantial.[2]

Enterprise Class SSD (EC)

The primary design goal of the Personal Class SSD Models A and B was to increase the bandwidth for individual read or write requests by utilizing 4-channel parallelism (and DRAM buffer and fat provisioning for Model B). While this architecture can help reduce the response time of individual I/O operations, it may not realize the throughput for multiple concurrent I/O requests up to the full potential. For example, if a single write request is followed by multiple read requests, relatively slow processing of the write request by a flash memory drive may block the subsequent read operations for an extended period of time. Since most of the read requests are processed synchronously by a transactional database system, even a single small write operation may limit the overall throughput of concurrent transactions considerably. (Refer to Section 5.1 for more discussions about the common pattern of *write followed by read* operations.)

To address this issue, which is critical to achieving a high throughput OLTP system dealing with a large number of concurrent read and write requests, a few architectural changes have been made to the Enterprise Class SSD. First, the number of channels is increased from four to eight, and the eight channels are allowed to process different I/O requests in parallel. This enables read requests to proceed much faster without being blocked by relatively slow write operations.

Second, the EC SSD supports the native command queuing (NCQ), which is one of the major features introduced by the SATA II standard [3]. NCQ allows multiple outstanding commands to be stored in an internal queue where the commands can be dynamically rescheduled or reordered. Coupled with the inter-command parallelism out of the eight channels, the NCQ support with the maximum queue depth of 32 can improve the transaction throughput of an OLTP system significantly. Besides, the EC SSD is equipped with a larger DRAM buffer and a more powerful processor to further improve the I/O buffering and to cope with the increased complexity of the controller logic.

4. THROUGHPUT OF RANDOM I/O

To evaluate the performance impact the different SSD designs have on I/O throughput, we compared the three flash memory SSD models described above (*i.e.*, PC-A, PC-B and EC) as well as a magnetic disk drive with respect to small random I/O operations, which are quite common in OLTP applications. The magnetic disk drive was an enterprise class model Seagate ST373455SS with 73.4GB capacity, 15k-RPM and a Serial Attached SCSI (SAS) interface. All the benchmark tests were run on a Linux system (kernel version 2.6.18-5) with an Intel Quad Q6600 processor and 2 GB RAM. The size of a random I/O operation was 8 KBytes and the I/O calibration tool used in this experiment was Oracle Orion [11].

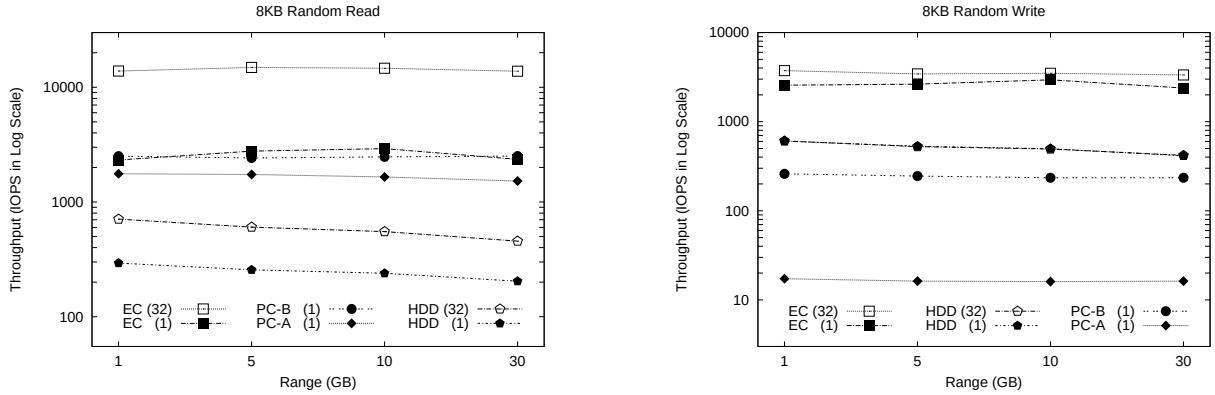
4.1 Throughput and Access Density

The read and write throughput of the three flash memory SSD prototypes and the magnetic disk are shown in Figures 1(a) and 1(b), respectively. For the devices equipped with a command queuing mechanism (namely, the EC SSD and the magnetic disk), the depth of a command queue was set to either one or 32.

The most noticeable was the impressive improvement in throughput repeatedly gained by advancing the flash mem-

[1]A previous study reports that even a small write buffer (*e.g.*, about 0.01% of the storage for PC workloads) can hide most of the write latency [7].

[2]Similar approaches have been applied to the design of flash translation layers (FTL) and flash-aware database systems. For example, in the ANAND and FMAX FTL schemes, replacement pages are used to absorb sector-level updates in a page-mode flash memory file system [1]. Under the In-Page Logging (IPL) scheme, which has been proposed for flash-aware buffer and storage managers of database systems, updates are accumulated in both in-memory and flash log sectors as a form of physiological log record, such that a large number of updates can be absorbed in a small number of physical writes [9].



(a) Read throughput (a) Write throughput

Figure 1: Small Random I/O Throughput vs. Data Access Range

ory SSD technology from the personal class PC-A to PC-B to the enterprise class EC model. In particular, while the PC-B SSD achieved more than an order of magnitude improvement in the write throughput over PC-A, the EC SSD improved the write throughput even further by another order of magnitude over PC-B, with an increased level of channel parallelism and a command queuing mechanism.

It was also notable that the write throughput of the EC SSD was high enough to surpass that of the enterprise class magnetic disk by a wide margin. When the depth of a command queue was set to 32 for both the EC SSD and the magnetic disk (denoted by `EC(32)` and `HDD(32)`), the write throughput of EC was higher than that of the magnetic disk by more than a factor of five. The read throughput of EC was higher than that of the magnetic disk by more than an order of magnitude.

Another observation we have made from the results shown in Figure 1 is that the throughput of the magnetic disk was quite sensitive to the range of data accesses. The read and write throughput of the magnetic disk decreased constantly as the range of data accesses increased. For example, when the depth of its command queue was set to 32, the read throughput of the magnetic disk decreased from about 700 IOPS for a one-GByte data range to about 400 IOPS for a 32-GByte data range. This is approximately 40% reduction in throughput. On the other hand, the throughput of all the flash memory SSD devices was largely insensitive to the range of data accesses. For example, the EC SSD with the depth of its command queue set to 32 maintained its read throughput approximately at 10,300 IOPS regardless of the range of data accesses.

This result is not surprising though. The obvious reason is that the average seek time of a magnetic disk is elongated proportionally as the average seek distance increases with the range of data accesses. In contrast, a flash memory SSD is a purely electronic device without any mechanical component, and its throughput is in all practical senses independent of the range of data accesses. This is clearly an evidence that flash memory drives can decelerate or even reverse the declining trend of access density for storage systems, while magnetic disks will continue to exacerbate it.

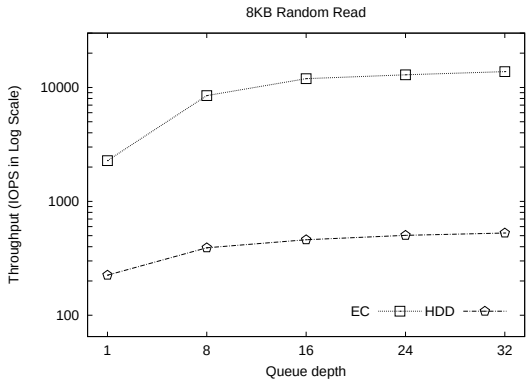
4.2 Effect of Command Queuing

For a large scale OLTP system, it is not uncommon to have several dozens of or more processes running concurrently and issuing numerous I/O operations simultaneously. Most modern disk drives with a SCSI or SATA-II interface support a *command queuing* mechanism that allows a disk controller to receive multiple outstanding I/O requests from a host system and lets the controller determine the best execution order to maximize the I/O throughput of the disk drive [15]. Dealing with multiple outstanding I/O requests from many concurrent transactions using a command queuing mechanism is instrumental in achieving high transaction rate for OLTP systems.

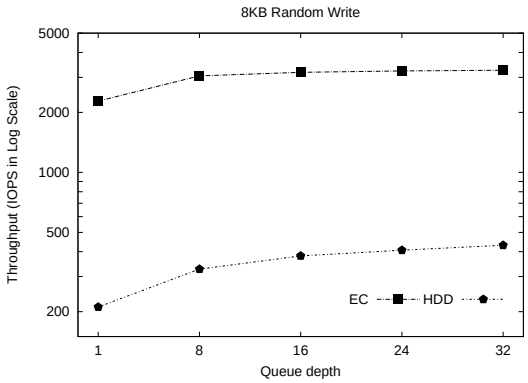
To evaluate the effects of command queuing, we measured the I/O throughput for the magnetic disk drive and the EC SSD drive with a varying number of outstanding I/O requests. The range of data accesses was set to 16 GBytes. Both the PC-A and PC-B were excluded from this experiment because neither of them supports command queuing. Figure 2 shows the random read and write throughput with varying queue depths from one to 32, which is the maximum queue depth supported by the native command queuing (NCQ) of the SATA II interface. Although the tagged command queuing (TCQ) of the SAS interface (for the magnetic disk) supports a command queue much deeper than 32, the maximum number of outstanding I/O requests was limited to this number for a fair comparison of the two drives.

As is shown in Figure 2, the read and write throughput consistently improved as the number of outstanding I/O requests increased. Let alone the clear and considerable gap in the scale of throughput between the two drives, the EC SSD improved the read throughput at a higher rate than the magnetic disk drive, as the queue depth increased. By increasing the queue depth from one to 32, the EC SSD improved its read throughput by about a factor of 5, while the magnetic disk did by less than a factor of 3. In the case of write throughput, however, the trend was reversed. The EC SSD improved its write throughput by about a factor of 1.5, and the magnetic disk did by about a factor of 2.

The throughput improvement gained by the EC SSD is attributed to the fact that the multi-channel architecture of the EC SSD makes it easier to exploit inter-command parallelism by distributing multiple outstanding I/O operations



(a) Read throughput



(a) Write throughput

Figure 2: Small Random I/O Throughput vs. Queue Depth

across available channels efficiently. By increasing the number of channels and the depth of a command queue, which appears to be the trend in the industry, the flash memory SSD is expected to further enhance the effectiveness of command queuing to yield higher throughput.

4.3 Effect of Page Size

Traditionally, for magnetic disk drives, large page sizes are considered more desirable than small ones for better performance of an I/O system. This is because the seek and rotational latency is the dominant portion of time required to access a disk page and the latency can be hidden by performing I/O operations in large units more effectively. The flash memory SSD with extremely low latency, however, does not benefit from using large page sizes for I/O operations. As a matter of fact, by adopting a small page size, a flash memory drive can avoid wasting the bandwidth by not accessing unnecessary data and improve the I/O throughout in inverse proportion to the unit size of I/O operations.

When the size of pages was reduced from 8 KBytes to 2 KBytes, both the read and write throughput of the flash memory drives (for all three prototypes) improved by 50 percent or more, while that of the magnetic disk drive improved only slightly by a little more than one percent.

5. TPC-C BENCHMARK

This section presents the benchmark results from the TPC-C workloads. The TPC-C benchmark is a mixture of read-only and update intensive transactions that simulate the activities found in OLTP application environments. Readers are referred to the TPC-C Specification [2] for the detailed description of the TPC-C benchmark such as database schema, transaction types and data distributions.

This benchmark was carried out by running a commercial database server on the same computing platform as described in Section 4. When multiple magnetic disks were attached to a RAID controller, the level of the RAID controller was set to RAID-0 (striped disks), and the cache on the controller was turned off. A commercial tool was used to create TPC-C workloads. The I/O requests from the database server were made in the default page size of 8 KBytes and were randomly scattered over the entire data space.[3]

To ensure that the I/O activities related to the database tables and indexes become dominant on the critical path for the transaction throughput, data spaces for log, rollback segments and temporary data were created on separate disk drives with caching enabled.

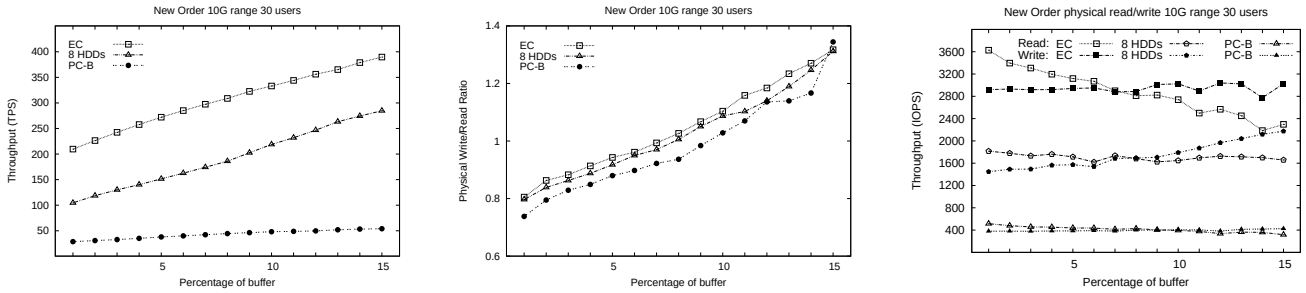
5.1 Impact of Read-Write Ratio

The `order status` is one of the five distinct types of transactions specified by the TPC-C benchmark. Transactions of this type are read-only, and the read requests are randomly scattered over the database. Therefore, a workload made only of this type of transactions tends to be I/O bound, and the transaction throughput of the workload is directly impacted by the throughput of random read operations. We observed that, for this read-only workload, the transaction throughput yielded by the EC SSD was approximately 5200 TPS, which was about an order of magnitude higher than the throughput yielded by the RAID-0 with eight 15k-RPM enterprise class disk drives. This should not be surprising because we already observed the read throughput of the EC SSD was higher than that of the magnetic disk by much more than an order of magnitude (as shown in Figure 1 and Figure 2).

On the other hand, for a mixture of read and write transactions, it is not as straightforward to predict the transaction throughput from the I/O throughput for flash memory SSD devices, because the read and write speeds of flash memory are asymmetric. For example, if the amounts of read and write requests are equal in a given workload, the average processing speed of a flash memory drive will be determined by a harmonic mean of the read and write speeds of the drive. The average I/O throughput would deteriorate quickly if the relative speed of write continued to lag behind.

In addition, the I/O model adopted by most database servers is also implicated in the transaction throughput for a mixed workload. While a read operation is requested *synchronously* by a transaction (or its process), a write operation is requested *asynchronously* by a transaction so that the actual write can be performed by a database server process

[3]We repeated the same benchmark test with 2 KByte pages to confirm the effect of page size on the TPC-C performance. With the size of pages reduced from 8 KByte to 2 KByte, the flash memory drives increased the transaction throughput for the TPC-C workload by about 50 percent, which matches the result shown in Section 4.3.



(a) Transaction Throughput (b) Write-to-Read Ratio (c) Read and Write IOPS

Figure 3: Write-to-Read Ratio and Buffer Size

(commonly known as a database writer) in the background on behalf of the transaction. Upon a page miss, a buffer manager then attempts to allocate a buffer frame for the data page. If there is no free buffer frame available or the number of free buffer frames is below threshold, the buffer manager will evict dirty pages from the buffer pool. This will result in a common pattern of *write followed by read* operations. The negative impact on transaction throughput by slow writes will be exacerbated if this pattern of I/O behaviors is prevalent.

Figure 3(a) shows the transaction throughput we observed for the `new order` read-write transactions of the TPC-C benchmark. Over the entire spectrum of a database buffer size we tested, the EC SSD outperformed the RAID-0 with eight disks consistently by a wide margin. On the other hand, the PC-B SSD underperformed significantly despite the fact that its random read throughput was about 4 times higher than that of a single magnetic disk. This was because the read requests from the transactions were often blocked or delayed by the slow write and the lack of inter-command parallelism and command queuing of the PC-B SSD model.

5.2 Impact of Database Buffer Size

The benchmark results from the `new order` transactions also show how the transaction throughput and I/O behaviors are affected by the size of a database buffer. Figure 3(b) shows that by increasing the size of a database buffer from one percent of the database size up to 15 percent, the ratio of write requests to read requests consistently increased from the range of 70 to 80 percent to the range of 110 to 130 percent irrespective of the storage devices used. Figure 3(c) shows the same trend in the absolute number of read and write throughput.

The change in write-to-read ratio appears to be affected by buffer hit ratio, which is mostly determined by the database buffer management independently of the characteristics of storage devices. We conjecture that the write-to-read ratio increases because the improved hit ratio reduces the number of read requests and thereby increasing the transaction throughput, which in turn increases the number of dirty pages in the buffer pool.

The increased write-to-read ratio does not affect the transaction throughput yielded by the RAID-0 negatively, since the read and write speeds of a magnetic disk drive are symmetric. The RAID-0 achieved about 170 percent throughput improvement by increasing the buffer size from one percent to 15 percent. On the other hand, due to the asymmetric read and write speeds, the average processing speed of a flash memory drive deteriorates as the write-to-read ratio increases. Both the EC SSD and PC-B SSD achieved only about 80 to 90 percent improvement in the transaction throughput, when the database buffer size increased by the same amount.

The replacement cost of a dirty page is always higher than that of a clean page regardless of an underlying storage medium. Again, due to the asymmetric read and write speeds of flash memory, however, the relative difference in the replacement cost will be more pronounced for flash memory drives than magnetic disk drives. We expect that a flash-aware buffer replacement algorithm (*e.g.*, CFLRU [12]) may help flash memory drives counter the negative performance impact from the increased write-to-read ratio to some extent.

5.3 Temporal Variation in Throughput

The amount of time taken to access a page in a magnetic disk drive varies considerably depending on the physical location of the page, because the physical location of the page determines the seek and rotational distances. In contrast, the variation in time taken to read a page from a flash memory drive is negligible regardless of the physical location of the page, because there is no mechanically moving part in the flash memory drive.

The same is true for writing a page to a flash memory drive too. However, writing a page into a flash memory drive may trigger a block-level operation such as a block relocation, a block erasure or a garbage collection. Since such a block-level operation takes much more time than a page-level operation, the time taken to process a write request may fluctuate severely depending on whether a block-level operation is triggered or not. The scope of reactions triggered by a write operation will be determined by several architectural design choices of flash memory SSD such as address mapping, garbage collection, channel parallelism, provisioning, write consolidation, wear leveling and so forth. Therefore, the amount of write time fluctuation can vary substantially from an SSD architecture to another.

Figure 4 shows the transaction throughput measured every 10 seconds for a mixture of `new order` and `payment` read-write transactions of the TPC-C benchmark. In this test, we included another flash memory SSD product from a different vendor (denoted by `SSD X`) to demonstrate the architectural

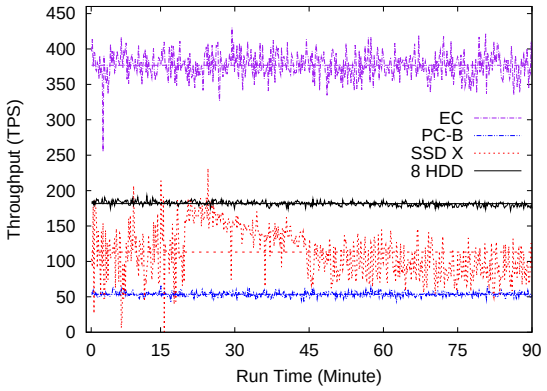


Figure 4: Variation in Transaction Throughput

effects on the throughput fluctuation. The SSD X is considered one of the most advanced SSD products available in the market.

Despite the variations in seek and rotational distances, the transaction throughput by the RAID-0 with eight disks was the most stable in terms of the coefficient of variation (*i.e.*, the ratio of the standard deviation to the mean). The standard deviation was only about 3.1 TPS with a mean of 181.6 TPS. The RAID-0 striping appears to have absorbed the latency variations by distributing I/O requests to multiple disk drives, but the effect of striping needs to be analyzed more carefully. The EC SSD was the second best (with standard deviation 18.3 TPS and mean 377.0 TPS) followed by the PC-B SSD (with standard deviation 3.5 TPS and mean 53.6 TPS) and the SSD X (with standard deviation 34.9 TPS and mean 113.2 TPS). Fluctuation in throughput is often tied up with surge in the amount of energy consumed by a flash memory drive. Evidently, this is one of the challenges we are faced with for the design of more stable and more energy efficient flash memory drives.

5.4 Energy Consumption

A recent study reports that disk drives are the largest energy consumer of a storage subsystem and the amount of energy consumed by disk drives is more than half of the total energy consumed by a typical TPC-C system [14]. With the rising cost of energy, the energy efficiency of storage systems has become one of the most serious concerns of large enterprises and data centers. As an electro-mechanical device, a magnetic disk consumes much energy for spinning its spindle and moving its arms. On the other hand, as a pure electronic device, a flash memory drive consumes a considerably less amount of energy for accessing data from or to flash chips. In this section, we demonstrate empirically how much energy can be saved by adopting flash memory drives for OLTP workloads.

For the sake of practicality and fair comparison, we examined the amount of energy consumed to achieve comparable peak transaction throughput for both the EC SSD and the RAID-0 with eight magnetic disks. Table 2 shows the amount of energy consumed when the computing platforms were idle, peak transaction throughput (with CPU utilization at the peak), and the amount of energy when they were fully utilized to reach the peak level of transaction throughput. The peak transaction throughput was obtained by adjusting the size of a database buffer, specifically by increasing the buffer size to 13 percent and 70 percent of the database size for EC SSD and RAID-0, respectively. This experiment was done with the `order status` read-only transactions of the TPC-C benchmark. For the ease of instrumentation, the amount of energy consumption reported in the table was obtained by measuring the power consumed by the entire system including either an EC SSD drive or a RAID-0 with eight disks rather than measuring for individual components.

System Configuration	Idle Energy	Peak TPS (CPU Util.)	Peak Energy
Basic + EC SSD	115.2W	4269 TPS (50%)	141.0W
Basic + RAID-0	197.2W	4274 TPS (45%)	231.4W

Table 2: Idle and Peak Power Consumption

When neither the EC SSD nor the RAID-0 was attached to the basic configuration, the computing platforms consumed approximately 113.1W when they were idle. As is shown in the second column of Table 2, the magnetic disk drives consume a non-trivial amount of energy even when the system is completely idle. When the system was idle, the additional amount of energy consumed by the EC SSD was very small at about 2.1W (115.2W−113.1W), but the additional amount of energy consumed by the RAID-0 was more than an order of magnitude than the EC SSD at about 84.1W (197.2W−113.1W).

The fourth column of Table 2 compares the computing platforms equipped with the two different storage devices in terms of the total amount of energy consumed to achieve the peak transaction throughput. With respect to the measurements, the system equipped with the EC SSD consumed about 39 percent less energy than the one with the RAID-0. However, it will be even more insightful to measure the amount of energy consumed only by the storage devices themselves, after discounting the energy consumed by the rest of the systems.

At the level of the CPU utilization shown in the third column of Table 2 with no I/O activity, both the computing platforms consumed approximately 135.0W of power. Therefore, we estimate that the amount of energy consumed by the storage devices at this level of transaction throughput was approximately 6.0W (141.0W−135.0W) by the EC SSD and 96.4W (231.4W−135.0W) by the RAID-0. This implies that the EC SSD consumed less than ten percent of the energy that would be consumed by the RAID-0 with eight disks to achieve the peak transaction throughput.

To analyze the energy use in a more typical operational condition, we carried out another set of experiments with a database buffer fixed to 10 percent of the database size. Table 3 shows the transaction throughput, CPU utilization and the energy consumption measured for the `order status` read-only transactions of the TPC-C benchmark. This result clearly demonstrates that the transaction throughput can be improved by more than a factor of four with about 36 percent less energy just by replacing eight magnetic disk drives with a single EC SSD.

System Configuration	TPS (CPU Util.)	Energy Consumed
Basic + EC SSD	4026 TPS (54%)	142.0W
Basic + RAID-0	845 TPS (14%)	232.0W

Table 3: Power Consumption for Read

Although flash memory uses more energy to inject charge into a cell until reaching a stable status than to read the status from a cell, the difference is very small and negligible when compared with the energy consumed by disk drives. Table 4 shows the results from the TPC-C benchmark carried out with `new order` read-write transactions at a typical level of workload with a database buffer set to 10 percent of the database size. The trend of energy consumption was not notably different from the results from the read-only workload shown in Table 2 and Table 3.

System Configuration	TPS (CPU Util.)	Energy Consumed
Basic + EC SSD	307 TPS (26%)	130.1W
Basic + RAID-0	233 TPS (17%)	222.4W

Table 4: Power Consumption for Read-Write

6. CONCLUSION

The previous study has shown that the overall throughput of an OLTP system can be improved considerably by adopting flash memory drives for transaction log, rollback and temporary data spaces [10]. This is because the burden of processing I/O requests occurring in these data spaces can become a serious bottleneck for transaction processing, but this bottleneck can be alleviated considerably by flash memory SSD.

The most recent advances in the SSD technology have enabled enterprise class flash memory drives to cope with randomly scattered I/O requests common in the database table and index spaces. The TPC-C benchmark results show that even a single Enterprise Class SSD drive can be on a par with or far better than a dozen spindles with respect to transaction throughput, cost effectiveness and energy consumption. The TPC-C benchmark results also suggest that traditional and new performance issues such as buffer replacement and fluctuation in transaction throughput need to be revisited and analyzed thoroughly for the design of flash-aware database systems and flash memory SSD for enterprise database applications.

Acknowledgment

The authors thank Mr. Sung-Tan Kim and Mr. Sung-Up Moon for assisting with the experiments.

7. REFERENCES

[1] Amir Ban and Ramat Hasharon. Flash File System Optimized for Page-Mode Flash Technologies, August 1999. United States Patent 5937425.

[2] Transaction Processing Performance Council. TPC Benchmark C Standard Specification (Revision 5.9). http://www.tpc.org, June 2007.

[3] Brian Dees. Native Command Queuing - Advanced Performance in Desktop Storage. *IEEE Potentials*, 24(4):4–7, Oct/Nov 2005.

[4] Jim Gray. Rules of Thumb in Data Engineering. In *Proceedings of ICDE*, pages 3–12, 2000.

[5] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach, 4th Edition*. Morgan Kaufmann, 2007.

[6] W. W. Hsu and A. J. Smith. Characteristics of I/O Traffic in Personal Computer and Server Workloads. *IBM Systems Journal*, 42(2):347–372, 2003.

[7] W. W. Hsu and A. J. Smith. The Performance Impact of I/O Optimizations and Disk Improvements. *IBM Journal of Research and Development*, 48(2):255–289, 2004.

[8] IBM. IBM Power 595 Server Model 9119-FHA Using AIX 5L Version 5.3 and DB2 Enterprise 9.5, June 2008. TPC Benchmark C Full Disclosure Report First Edition.

[9] Sang-Won Lee and Bongki Moon. Design of Flash-based DBMS: an In-Page Logging Approach. In *Proceedings of the ACM SIGMOD*, pages 55–66, 2007.

[10] Sang-Won Lee, Bongki Moon, Chanik Park, Jae-Myung Kim, and Sang-Woo Kim. A Case for Flash Memory SSD in Enterprise Database Applications. In *Proceedings of the ACM SIGMOD*, pages 1075–1086, 2008.

[11] Oracle. ORION: Oracle I/O Numbers Calibration Tool. http://www.oracle.com/technology/software/-tech/orion/.

[12] Seon-Yeong Park, Dawoon Jung, Jeong-Uk Kang, Jin-Soo Kim, and Joonwon Lee. CFLRU: a Replacement Algorithm for Flash Memory. In *the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES'06)*, pages 234–241, October 2006.

[13] David A. Patterson. Latency Lags Bandwidth. *Communications of the ACM*, 47(10):71–75, October 2004.

[14] Meikel Poess and Raghunath Othayoth Nambiar. Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results. In *Proceedings of VLDB*, 2008.

[15] Chris Ruemmler and John Wilkes. An Introduction to Disk Drive Modeling. *IEEE Computer*, 27:17–28, 1994.