Evaluating Phase Change Memory for Enterprise Storage Systems: A Study of Caching and Tiering Approaches

Hyojun Kim, Sangeetha Seshadri, Clement L. Dickey, Lawrence Chiu
IBM Almaden Research

Abstract

Storage systems based on Phase Change Memory (PCM) devices are beginning to generate considerable attention in both industry and academic communities. But whether the technology in its current state will be a commercially and technically viable alternative to entrenched technologies such as flash-based SSDs remains undecided. To address this it is important to consider PCM SSD devices not just from a device standpoint, but also from a holistic perspective.

This paper presents the results of our performance study of a recent all-PCM SSD prototype. The average latency for a 4 KiB random read is 6.7 µs, which is about 16× faster than a comparable eMLC flash SSD. The distribution of I/O response times is also much narrower than flash SSD for both reads and writes. Based on the performance measurements and real-world workload traces, we explore two typical storage use-cases: tiering and caching. For tiering, we model a hypothetical storage system that consists of flash, HDD, and PCM to identify the combinations of device types that offer the best performance within cost constraints. For caching, we study whether PCM can improve performance compared to flash in terms of aggregate I/O time and read latency. We report that the IOPS/$ of a tiered storage system can be improved by 12–66% and the aggregate elapsed time of a server-side caching solution can be improved by up to 35% by adding PCM.

Our results show that – even at current price points – PCM storage devices show promising performance as a new component in enterprise storage systems.

1 Introduction

In the last decade, solid-state storage technology has dramatically changed the architecture of enterprise storage systems. Flash memory based solid state drives (SSDs) outperform hard disk drives (HDDs) along a number of dimensions. When compared to HDDs, SSDs have higher storage density, lower power consumption, a smaller thermal footprint and orders of magnitude lower latency. Flash storage has been deployed at various levels in enterprise storage architecture ranging from a storage tier in a multi-tiered environment (e.g., IBM Easy Tier [15], EMC FAST [9]) to a caching layer within the storage server (e.g., IBM XIV SSD cache [17]), to an application server-side cache (e.g., IBM Easy Tier Server [16], EMC XtreamSW Cache [10], NetApp Flash Accel [24], FusionIO ioTurbine [11]). More recently, several all-flash storage systems that completely eliminate HDDs (e.g., IBM FlashSystem 820 [14], Pure Storage [25]) have also been developed. However, flash memory based SSDs come with their own set of concerns such as durability and high-latency erase operations.

Several non-volatile memory technologies are being considered as successors to flash. Magneto-resistive Random Access Memory (MRAM [2]) promises even lower latency than DRAM, but it requires improvements to solve its density issues; the current MRAM designs do not come close to flash in terms of cell size. Ferroelectric Random Access Memory (FeRAM [13]) also promises better performance characteristics than flash, but lower storage density, capacity limitations, and higher cost issues remain to be addressed. On the other hand, Phase Change Memory (PCM [29]) is a more imminent technology that has reached a level of maturity that permits deployment at commercial scale. Micron announced mass production of a 128 Mbit PCM device in 2008 while Samsung announced the mass production of 512 Mbit PCM device follow-on in 2009. In 2012, Micron also announced in volume production of a 1 Gbit PCM device.

PCM technology stores data bits by alternating the phase of material between *crystalline* and *amorphous*. The crystalline state represents a logical 1 while the amorphous state represents a logical 0. The phase is alternated by applying varying length current pulses depending upon the phase to be achieved, representing the write operation. Read operations involve applying a small current and measuring the resistance of the material.

Flash and DRAM technologies represent data by storing electric charge. Hence these technologies have difficulty scaling down to thinner manufacturing processes, which may result in bit errors. On the other hand, PCM technology is based on the phase of material rather than electric charge and has therefore been regarded as more scalable and durable than flash memory [28].

In order to evaluate the feasibility and benefits of PCM technologies from a systems perspective, access to accurate *system-level* device performance characteristics is essential. Extrapolating *material-level* characteristics to a *system-level* without careful consideration may result in inaccuracies. For instance, a previously published paper states that PCM write performance is only 12× slower than DRAM based on the 150 ns *set operation time* reported in [4]. However, the reported write throughput from the referred publication [4] is only 2.5 MiB/s, and thus the statement that PCM write performance is only 12× slower is misleading. The missing link is that only *two bits* can be written during 200 µs on the PCM chip because of circuit delay and power consumption issues [4]. While we may conclude that *PCM write operations* are 12× slower than *DRAM write operations*, it is incorrect to conclude that a *PCM device* is only 12× slower than a *DRAM device* for writes. This reinforces the need to consider PCM performance characteristics from a system perspective based on independent measurement in the right setting as opposed to simply re-using device level performance characteristics.

Our first contribution is the result of our *system-level* performance study based on a real prototype all-PCM SSD from Micron. In order to conduct this study, we have developed a framework that can measure I/O latencies at nanosecond granularity for read and write operations. Measured over five million random 4 KiB read requests, the PCM SSD device achieves an average latency of 6.7 µs. Over one million random 4 KiB write requests, the average latency of a PCM SSD device is about 128.3 µs. We compared the performance of the PCM SSD with an *Enterprise Multi-Level Cell (eMLC)* flash based SSD. The results show that in comparison to eMLC SSD, read latency is about 16× shorter, but write latency is 3.5× longer on the PCM SSD device.

Our second contribution is an evaluation of the feasibility and benefits of including a PCM SSD device as a tier within a multi-tier enterprise storage system. Based on the conclusions of our performance study, reads are faster but writes are slower on PCM SSDs when compared to flash SSDs, and at present PCM SSDs are priced higher than flash SSD ($ / GB). Does a system built with a PCM SSD offer any advantage over one without PCM SSDs? We approach this issue by modeling a hypothetical storage system that consists of three device types: PCM SSDs, flash SSDs, and HDDs. We evaluate this storage system using several real-world traces to identify optimal configurations for each workload. Our results show that PCM SSDs can remarkably improve the performance of a tiered storage system. For instance, for a one week retail workload trace, 30% PCM + 67% flash + 3% HDD combination has about 81% increased IOPS/$ from the best configuration without PCM, 94% flash + 6% HDD even when we assume that PCM SSD devices are four times more expensive than flash SSDs.

Our third contribution is an evaluation of the feasibility and benefits of using a PCM SSD device as an application server-side cache *instead of* or *in combination with* flash. Today flash SSD based server-side caching solutions are appearing in the industry [10, 11, 16, 24] and also gaining attention in academia [12, 20]. What is the impact of using the 16× faster (for reads) PCM SSD instead of flash SSD as a server-side caching device? We run cache simulations with real-world workload traces from enterprise storage systems to evaluate this. According to our observations, a combination of flash and PCM SSDs can provide better aggregate I/O time and read latency than a flash only configuration.

The rest of the paper is structured as follows: Section 2 provides a brief background and discusses related work. We present our measurement study on a real all-PCM prototype SSD in Section 3. Section 4 describes our model and analysis for a hypothetical tiered storage system with PCM, flash, and HDD devices. Section 5 covers the use-case for server-side caching with PCM. We present a discussion of the observations in Section 6 and conclude in Section 7.

2 Background and related work

There are two possible approaches to using PCM devices in systems: as storage or as memory. The storage approach is a natural option considering the non-volatile characteristics of PCM, and there are several very interesting studies based on real PCM devices.

In 2008, Kim, et al. proposed a hybrid Flash Translation Layer (FTL) architecture, and conducted experiments with a real 64 MiB PCM device (KPS1215EZM) [19]. We believe that the PCM chip was based on 90 nm technology, published in early 2007 [22]. The paper reported 80 ns and 10 µs as word (16 bits) access time for read and write, respectively. Better write performance numbers are found in Samsung's 2007 90 nm PCM paper [22]: 0.58 MB/s in ×2 division-write mode, 4.64 MB/s in ×16 accelerated write mode.

Table 1: A PCM SSD prototype: *Micron built an all-PCM SSD prototype with their newest 45 nm PCM chips.*

Usable Capacity	64 GiB
System Interface	PCIe gen2 x8
Minimum Access Size	4 KiB
Seq. Read BW. (128 KiB)	2.6 GiB/s
Seq. Write BW. (128 KiB)	100-300 MiB/s

In 2011, a prototype all-PCM 10 GB SSD was built by researchers from the University of California, San Diego [1]. This SSD, named *Onyx*, was based on Micron's first-generation P8P 16 MiB PCM chips (NP8P128A13B1760E). On the chip, a read operation for 16 bytes takes 314 ns (48.6 MB/s), and a write operation for 64 bytes requires 120 µs (0.5 MB/s). Onyx drives many PCM chips concurrently, and provides 38 µs and 179 µs for 4 KiB read and write latencies, respectively. The Onyx design corroborates the potential of PCM as a storage device which allows massive parallelization to improve the limited write throughput of today's PCM chips. In 2012, another paper was published based on a different prototype PCM SSD built by Micron [3], using the same Micron 90 nm PCM chip used in Onyx. This prototype PCM SSD provides 12 GB capacity, and takes 20 µs and 250 µs for 4 KiB read and write, respectively, excluding software overhead. This device shows better read performance and worse write performance than the one presented in Oynx. The authors compare the PCM SSD with Fusion IO's Single-Level Cell (SLC) flash SSD, and point out that PCM SSD is about 2× faster for read, and 1.6× slower for write than the compared flash SSD.

Alternatively, PCM devices can be used as memory [18, 21, 23, 26, 27]. The main challenge in using PCM devices as a memory device is that writes are too slow. In PCM technology, high heat (over 600°C) is applied to a storage cell to change the phase to store data. The combination of quick heating and cooling results in the amorphous phase, and this operation is referred to as a *reset operation*. The *set operation* requires a longer cooling time to switch to the crystalline phase, and write performance is determined by the time required for a set operation. In several papers, PCM's set operation time is used as an approximation for the write performance for a simulated PCM device. However, care needs to be taken to differentiate among material, chip-level and device level performance. Set and reset operation times describe material level performance, which is often very different from chip level performance. For example, in Bedeschi et al. [4], the set operation time is 150 ns, but reported write throughput is only 2.5 MB/s because only two bits can be written concurrently, and there is an additional circuit delay of 50 ns. Similarly, the chip level performance differs from the device level (SSD) performance. In the rest of the paper, our performance measurements address *device level* performance based on a recent PCM SSD prototype device based on newer 45 nm chips from Micron.

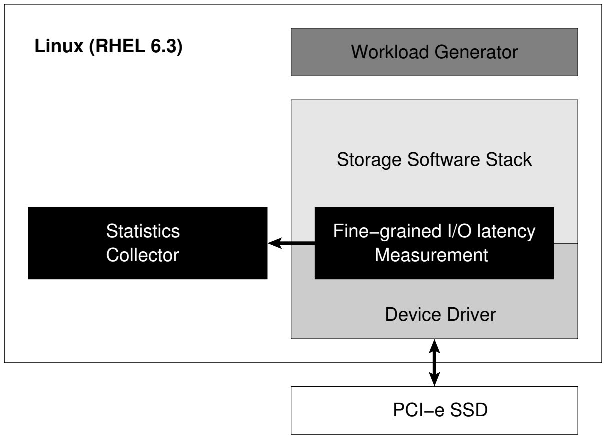


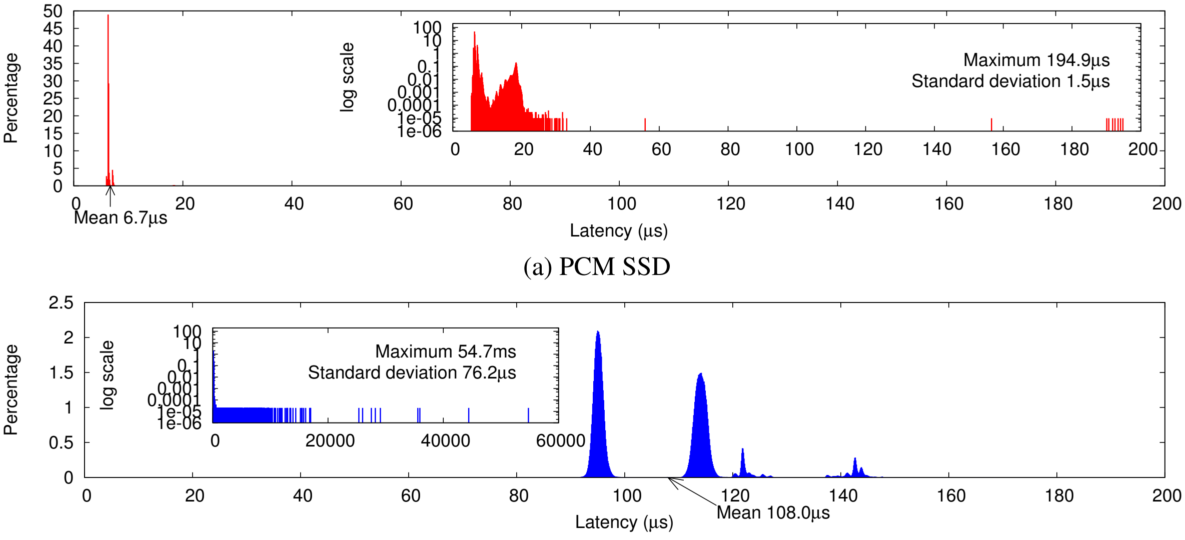
Figure 1: Measurement framework: *we modified both the Linux kernel and the device driver to collect I/O latencies in nanosecond units. We also use an in-house workload generator and a statistics collector.*

3 PCM SSD performance

In this section we describe our methodology and results for the characterization of system-level performance of a PCM SSD device. Table 1 summarizes the main features of the prototype PCM SSD device used for this study.

In order to collect fine-grained I/O latency measurements, we have patched the kernel of Red Hat Enterprise Linux 6.3. Our kernel patch enables measurement of I/O response times at nanosecond granularity. We have also modified the drivers of the SSD devices to measure the elapsed time from the arrival of an I/O request at the SSD to its completion (at the SSD). Therefore, the I/O latency measured by our method includes minimal software overhead.

Figure 1 shows our measurement framework. The system consists of a workload generator, a modified storage stack within the Linux kernel that can measure I/O latencies at nanosecond granularity, a statistics collector, and a modified device driver that measures the elapsed time for an I/O request. For each I/O request generated by the workload generator, the device driver measures the time required to service the request and passes that information back to the Linux kernel. The modified Linux kernel keeps the data in two different forms: a histogram (for long term statistics) and a fixed length log (for precise



(a) PCM SSD

(b) eMLC SSD

Figure 2: **4 KiB random read latencies for five million samples**: *PCM SSD shows about 16× faster average, much smaller maximum, and also much narrower distribution than eMLC SSD.*

data collection). Periodically, the collected information is passed to an external *statistics collector*, which stores the data in a file.

For the purpose of comparison, we use an eMLC flash-based PCI-e SSD providing 1.8 TiB user capacity. To capture the performance characteristics at extreme conditions, we precondition both the PCM and the eMLC flash SSDs using the following steps: 1) Perform raw formatting using tools provided by SSD vendors. 2) Fill the whole device (usable capacity) with random data, sequentially. 3) Run full random, 20% write, 80% read I/O requests with 256 concurrent streams for one hour.

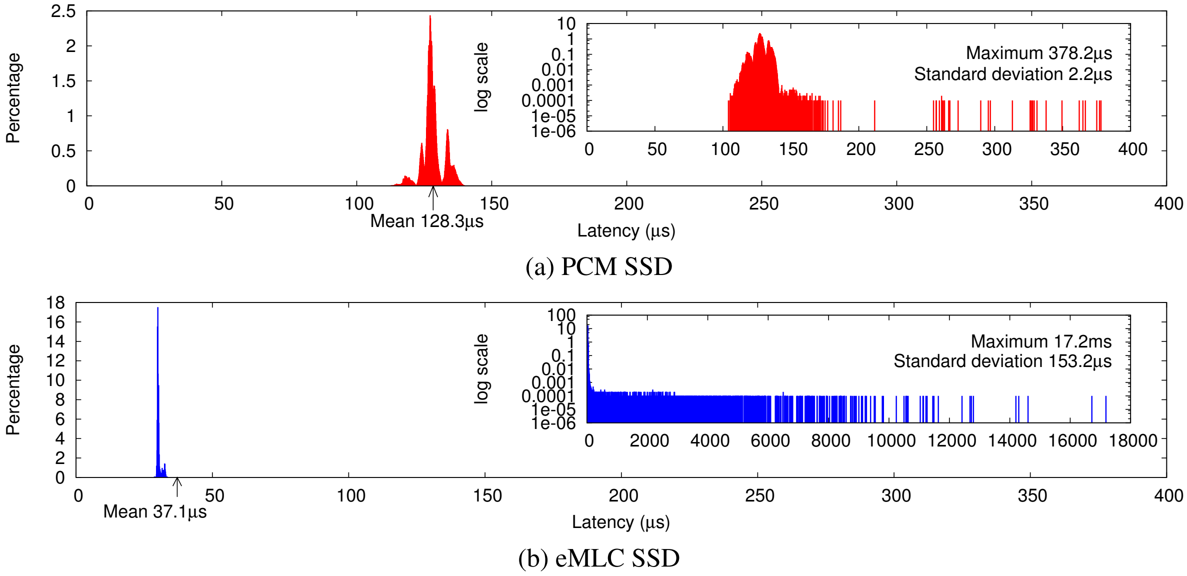
3.1 I/O Latency

Immediately after the preconditioning is complete we set the workload generator to issue one million 4 KiB sized random write requests with a single thread. We collect write latency for each request and the collected data is periodically retrieved and written to a performance log file. After one million writes complete, we set the workload generator to issue five million 4 KiB sized random read requests by using a single thread. Read latencies are collected using the same method.

Figure 2 shows the distributions of collected read latencies for the PCM SSD (Figure 2(a)) and the eMLC SSD (Figure 2(b)). The X-axis represents the measured read latency, and the Y-axis represents the percentage of data samples. Each graph has a smaller graph embedded, which presents the whole data range with a log scaled Y-axis.

Several important results can be observed from the graphs. First, the average latency of the PCM SSD device is only 6.7 µs, which is about 16× faster than the eMLC flash SSD's average read latency of 108.0 µs. This number is much improved from the prior PCM SSD prototypes (Onyx: 38 µs [1], 90 nm Micron: 20 µs [3]). Second, the PCM SSD latency measurements show much smaller standard deviation (1.5 µs, 22% of mean) than the eMLC flash SSD's measurements (76.2 µs, 71% of average). Finally, the maximum latency is also much smaller on the PCM SSD (194.9 µs) than on the eMLC flash SSD (54.7 ms).

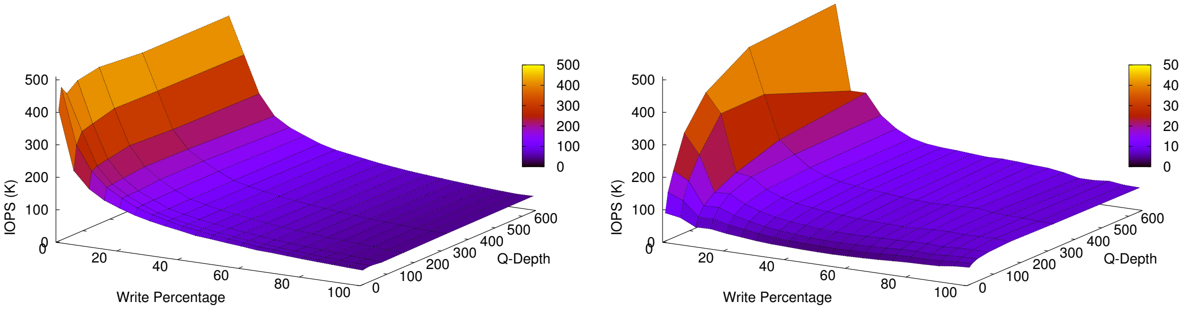
Figure 3 shows the latency distribution graphs for 4 KiB random writes. Interestingly, eMLC flash SSD (Figure 3(b)) shows a very short average write response time of only 37.1 µs. We believe that this is due to the RAM buffer within the eMLC flash SSD. Note that over 240 µs latency was measured for 4 KiB random writes even on Fusion IO's SLC flash SSD [3]. According to our investigation, the PCM SSD prototype does not implement RAM based write buffering, and the measured write latency is 128.3 µs (Figure 3(a)). Even though this latency number is about 3.5× longer than the eMLC SSD's average, it is still much better than the performance measurements from previous PCM prototypes. Previous measurements reported for 4 KiB write latencies are 179 µs and 250 µs in Onyx [1] and 90 nm PCM SSDs [3], respectively. As in the case of reads, for standard deviation and maximum value measurements the PCM SSD outperforms the eMLC SSD; the PCM SSD's standard deviation is only 2% of the average and the



(a) PCM SSD

(b) eMLC SSD

Figure 3: **4 KiB random write latencies for one million samples**: *PCM SSD shows about 3.5× slower mean, but its maximum and distribution are smaller and narrower than eMLC SSD.*



(a) PCM SSD

(b) eMLC SSD

Figure 4: Asynchronous IOPS: *I/O request handling capability for different read and write ratios and for different degree of parallelism.*

maximum latency is 378.2 µs while the eMLC flash SSD shows 153.2 µs standard deviation (413% of the average) and 17.2 ms maximum latency value. These results lead us to conclude that the PCM SSD performance is more consistent and hence predictable than that of the eMLC flash SSD.

Micron provided this feedback on our measurements: this prototype SSD uses a PCM chip architecture that was designed for code storage applications, and thus has limited write bandwidth. Micron expects future devices targeted at this application to have lower write latency. Furthermore, the write performance measured in the drive is not the full capability of PCM technology. Additional work is ongoing to improve the write characteristics of PCM.

3.2 Asynchronous I/O

In this test, we observe the number of I/Os per second (IOPS) while varying the read and write ratio and the degree of parallelism. In Figure 4, two 3-dimensional graphs show the measured results. The X-axis represents the percentage of writes, the Y-axis represents the queue depth (i.e. number of concurrent IO requests issued), and the Z-axis represents the IOPS measured. The most obvious difference between the two graphs occurs when the queue depth is low and all requests are reads (lower left corner of the graphs). At this point, the PCM SSD shows much higher IOPS than the eMLC flash SSD. For the PCM SSD, performance does not vary much with variation in queue depth. However, on the eMLC SSD, IOPS increases with increase in queue depth. In general, the

Table 2: The parameters for tiering simulation

	PCM	eMLC	15K HDD
4 KiB R. Lat.	6.7 µs	108.0 µs	5 ms
4 KiB W. Lat.	128.3 µs	37.1 µs	5 ms
Norm. Cost	24	6	1

PCM SSD shows smoother surfaces when varying the read / write ratio. It again supports our finding that the PCM SSD is more predictable than the eMLC flash SSD.

4 Workload simulation for storage tiering

The results of our measurements on PCM SSD device performance show that the PCM SSD improves read performance by 16×, but shows about 3.5× slower write performance than eMLC flash SSD. Will such a storage device be useful for building enterprise storage systems? Current flash SSD and HDD tiered storage systems maximize *performance per dollar* (price-performance ratio) by placing hot data on faster flash SSD storage and cold data on cheaper HDD devices. Based on PCM SSD device performance, an obvious approach is to place hot, read intensive data on PCM devices; hot, write intensive data on flash SSD devices; and cold data on HDD to maximize performance per dollar. But do real-world workloads demonstrate such workload distribution characteristics? In order to address this question, we first model a hypothetical tiered storage system consisting of PCM SSD, flash SSD and HDD devices. Next we apply to our model several real-world workload traces collected from enterprise tiered storage systems consisting of flash SSD and HDD devices. Our goal is to understand whether there is any advantage to using PCM SSD devices based on the characteristics exhibited by real workload traces.

Table 2 shows the parameters used for our modeling. For PCM and flash SSDs, we use the data collected from our measurements. For the HDD device we use 5 ms for both 4 KiB random read and write latencies [7]. We compare the various alternative configurations using performance per dollar as a metric. In order to use this metric, we need price estimates for the storage devices. We assume that a PCM device is 4× more expensive than eMLC flash, and eMLC flash is 6× more expensive than 15 K RPM HDD. The flash-HDD price assumption is based on today's (June 2013) market prices from Dell's web page [6, 8]. We prefer the Dell's prices to Newegg's or Amazon's because we want to use prices for enterprise class devices. The PCM-flash price assumption is based on an opinion from an expert who prefers to remain anonymous; it is our best effort considering that the 45 nm PCM device is not available in the market yet.

We present two methodologies for evaluating PCM capabilities for a tiering approach: static optimal tiering and dynamic tiering. Static optimal tiering assumes static and optimal data placement based on complete knowledge about a given workload. While this methodology provides a simple back-of-the-envelope calculation to evaluate the effectiveness of PCM, we acknowledge that this assumption may be unrealistic and that data placements need to adapt dynamically to runtime changes in workload characteristics.

Accordingly, our second evaluation methodology is a simulation-based technique to evaluate PCM deployments in a dynamic tiered setting. Dynamic tiering assumes that data migrations are reactive and dynamic in nature and in response to changes in workload characteristics and system conditions. The simulated system begins with no prior knowledge about the workload. The simulation algorithm then periodically gathers I/O statistics, learns workload behavior and migrates data to appropriate locations in response to workload characteristics.

4.1 Evaluation metric

For a given workload observation window and a hypothetical storage composed of *X%* of PCM, *Y%* of flash, and *Z%* of HDD, we calculate the IOPS/$ metric using the following steps:

Step 1. From a given workload during the observation window, aggregate the total amount of read and write I/O traffic at an extent (1 GiB) granularity. An extent is the unit of data migration in tiered storage environment. In our analysis, the extent size is set to 1 GiB accordingly to the configuration of the real-world tiered storage systems from which our workload traces were collected.

Step 2. Let $ReadLat._{HDD}$, $ReadLat._{Flash}$ and $ReadLat._{PCM}$ represent the read latencies of HDD, flash and PCM devices respectively. Similarly, let $WriteLat._{HDD}$, $WriteLat._{Flash}$ and $WriteLat._{PCM}$ represent the write latencies. Let $ReadAmount_{Extent}$ and $WriteAmount_{Extent}$ represent the amount of read and write traffic given to the extent under consideration. For each extent, calculate $Score_{Extent}$ using the following equations:

$$\begin{aligned}Score_{PCM} &= (ReadLat._{HDD} - ReadLat._{PCM}) \times ReadAmount_{Extent} + \\ &\quad (WriteLat._{HDD} - WriteLat._{PCM}) \times WriteAmount_{Extent} \\ Score_{Flash} &= (ReadLat._{HDD} - ReadLat._{Flash}) \times ReadAmount_{Extent} + \\ &\quad (WriteLat._{HDD} - WriteLat._{Flash}) \times WriteAmount_{Extent} \\ Score_{Extent} &= MAX(Score_{PCM}, Score_{Flash})\end{aligned}$$

Step 3. Sort extents by $Score_{Extent}$ in descending order.

Step 4. Assign a tier for each extent based on Algorithm 1. This algorithm can fail if either (1) HDD is the best choice, or (2) we run out of HDD space, but that will never happen with our configuration parameters.

Algorithm 1 Data placement algorithm

```
for $e$ in $SortedExtentsByScore$ do
    $tgtTier \leftarrow (e.scorePCM > e.scoreFlash)?PCM : FLASH$
    if $(tgtTier.freeExt > 0)$ then
        $e.tier \leftarrow tgtTier$
        $tgtTier.freeExt \leftarrow tgtTier.freeExt - 1$
    else
        $tgtTier \leftarrow (tgtTier == PCM)?FLASH : PCM$
        if $(tgtTier.freeExt > 0)$ then
            $e.tier \leftarrow tgtTier$
            $tgtTier.freeExt \leftarrow tgtTier.freeExt - 1$
        else
            $e.tier \leftarrow HDD$
        end if
    end if
end for
```

Step 5. Aggregate the amount of read and write I/O traffic for PCM, flash, and HDD tiers based on the data placement.

Step 6. Calculate *expected average latency* based on the amount of read and write traffic received by each storage media type and the parameters in Table 2.

Step 7. Calculate *expected average IOPS* as 1 / *expected average latency*.

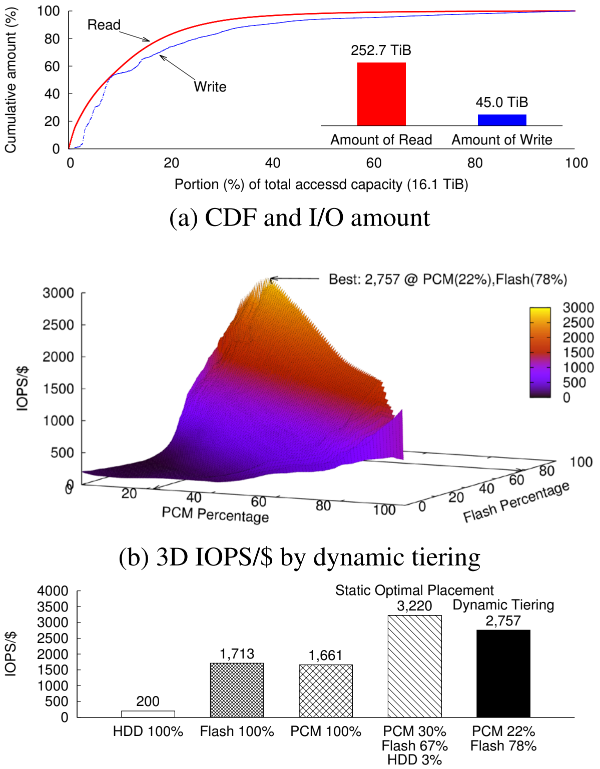
Step 8. Calculate *normalized cost* based on the percentage of storage: for example, the *normalized cost* for an all-HDD configuration is 1, and the *normalized cost* for a 50% PCM + 50% flash configuration is $(24 \times 0.5) + (6 \times 0.5) = 15$.

Step 9. Calculate *performance-price ratio = IOPS/$* as *expected average IOPS* (from Step 7) / *normalized cost* (from Step 8).

The value obtained from Step 9 represents the IOPS per normalized cost – a higher value implies better performance per dollar. We repeat this calculation for every possible combination of PCM, flash, and HDD to find the most desirable combination for a given workload.

4.2 Simulation methodology

In the case of the static optimal placement methodology, the entire workload duration is treated as a single observation window and we assume unlimited migration bandwidth. The dynamic tiering methodology uses a two-hour workload observation window before making migration decisions and assumes a migration bandwidth of 41 MiB/s according to the configurations of real-world tiered storage systems from which we collected workload traces. Our experimental evaluation shows that utilizing PCM can result in a significant performance improvement. We compare the results from the static optimal methodology and the dynamic tiering methodology using the evaluation metric described in Section 4.1.



(a) CDF and I/O amount

(b) 3D IOPS/$ by dynamic tiering

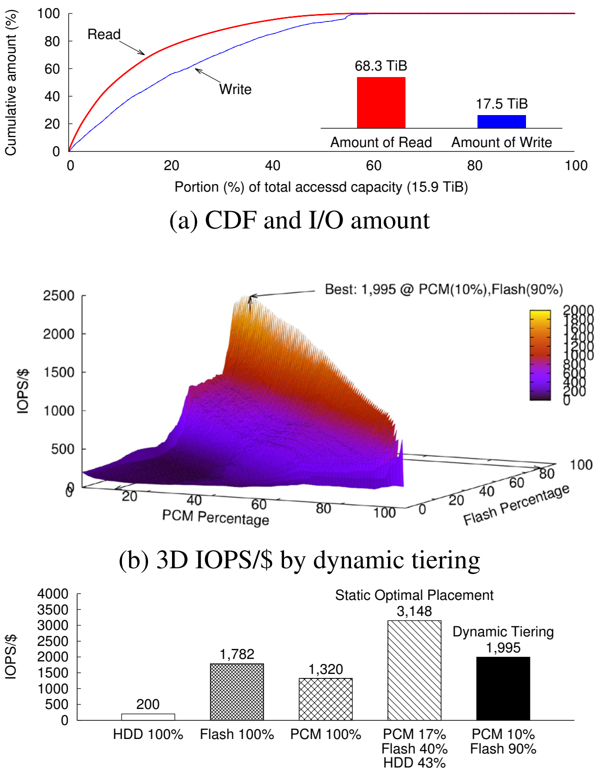
(c) IOPS/$ for key configuration points

Figure 5: Simulation result for the retail store trace: *this workload is very friendly for PCM; read dominant and highly skewed spatially – PCM (22%) + flash (78%) configuration can make the best IOPS/$ value (2,757) in dynamic tiering simulation.*

4.3 Result 1: Retail store

The first trace is a one week trace collected from an enterprise storage system used for online transactions at a retail store. Figure 5(a) shows the cumulative distribution as well as the total amount of read and write I/O traffic: the total storage capacity accessed during this duration is 16.1 TiB, the total amount of read traffic is 252.7 TiB, and the total amount of write traffic is 45.0 TiB. As can be seen from the distribution, the workload is heavily skewed, with 20% of the storage capacity receiving 83% of the read traffic and 74% of the write traffic. The distribution also exhibits a heavy skew toward reads, with nearly six times more reads than writes.

Figures 5 (b) and (c) show the modeling results. Graph (b) represents performance price ratios obtained by dynamic tiering simulation on a 3-dimensional surface, and graph (c) shows the same performance–price values (IOPS/$) for several important data points: all-HDD, all-flash, all-PCM, the best configuration for static optimal data placement, and the best configuration for



(a) CDF and I/O amount

(b) 3D IOPS/$ by dynamic tiering

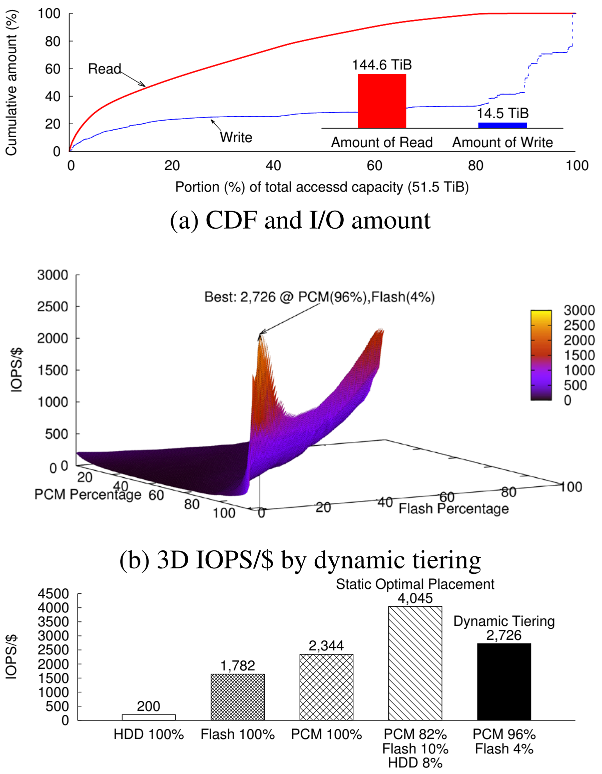
(c) IOPS/$ for key configuration points

Figure 6: Simulation result for the bank trace: *this workload is less friendly for PCM than the retail workload – PCM (10%) + flash (90%) configuration can make the best IOPS/$ value (1,995) in dynamic tiering simulation.*

dynamic tiering. Note that for the first three homogeneous storage configurations, there is no difference between static and dynamic simulation results. The best combination using static data placement consists of PCM (30%) + flash (67%) + HDD (3%), and the calculated IOPS/$ value is 3,220, which is about 81% higher than the best combination without PCM: 94% flash + 6% HDD yielding 1,777 IOPS/$; the best combination from dynamic tiering simulation consists of PCM (22%) + flash (78%), and the obtained IOPS/$ value is 2,757. This value is about 61% higher than the best combination without PCM: 100% flash yielding 1,713 IOPS/$.

4.4 Result 2: Bank

The second trace is a one week trace from a bank. The total storage capacity accessed is 15.9 TiB, the total amount of read traffic is 68.3 TiB, and the total amount of write traffic is 17.5 TiB as shown in Figure 6(a). Read to write ratio is 3.9 : 1, and the degree of skew toward reads is less than the previous retail store trace (Figure 5(a)). Approximately 20% of the storage capacity receives about 76% of the read traffic and 56% of the write traffic.



(a) CDF and I/O amount

(b) 3D IOPS/$ by dynamic tiering

(c) IOPS/$ for key configuration points

Figure 7: Simulation result for the telecommunication company trace: *this workload is less spatially skewed, but the amount of read is about 10× of the amount of write – PCM (96%) + flash (4%) configuration can make the best IOPS/$ value (2,726) in dynamic tiering simulation.*

Figures 6(b) and (c) show the modeling results. The best combination using static data placement consists of PCM (17%) + flash (40%) + HDD (43%), and the calculated IOPS/$ value is 3,148, which is about 14% higher than the best combination without PCM: 57% flash + 43% HDD yielding 2,772; the best combination from dynamic tiering simulation consists of PCM (10%) + flash (90%), and the obtained IOPS/$ value is 1,995. This value is about 12% higher than the best combination without PCM: 100% flash yielding 1,782 IOPS/$.

4.5 Result 3: Telecommunication company

The last trace is a one week trace from a telecommunication provider. The total accessed storage capacity is 51.5 TiB, the total amount of read traffic is 144.6 TiB, and the total amount of write traffic is about 14.5 TiB. As shown in Figure 7(a), this workload is less spatially

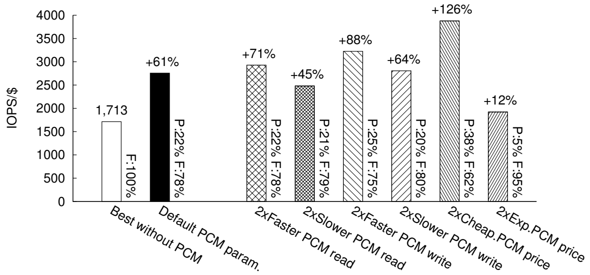


Figure 8: The best IOPS/$ for Retail store workload with varied PCM parameters

skewed than the retail and bank workloads; approximately 20% of the storage capacity receives about 52% of the read traffic and 23% of the write traffic. But read to write ratio is about 10 : 1, which is the most read dominant among the three workloads.

According to Figures 7(b) and (c), the best combination from static data placement consists of PCM (82%) + flash (10%) + HDD (8%), and calculated IOPS/$ value is 4,045, which is about 2.2× better than the best combination without PCM: 84% flash + 16% HDD yielding 1,853; the best combination from dynamic tiering simulation consists of PCM (96%) + flash (4%), and the obtained IOPS/$ value is 2,726. This value is about 66% higher than the best combination without PCM: 100% flash yielding 1,641 IOPS/$.

4.6 Sensitivity analysis for tiering

The simulation parameters are based on our best effort estimation of market price and the current state of PCM technologies, or based on discussions with experts. However, PCM technology and its markets are still evolving, and there are uncertainties about its characteristics and pricing. To understand the sensitivity of our simulation results to PCM parameters, we tried six variations of PCM parameters in three aspects: read performance, write performance, and price. For each aspect, we tried half-size and double-size values. For instance, we tested 4.35 µs and 13.4 µs instead of the original 6.7 µs for PCM 4 KiB read latency.

Figure 8 shows the highest IOPS/$ value for varying PCM parameters. We observe that our IOPS/$ measure is most sensitive to PCM price. If PCM is only twice as expensive as flash while maintaining its read and write performance, the PCM (38%) + flash (62%) configuration can yield about 126% higher IOPS/$ (3,878); if PCM is 8× more expensive than flash, PCM (5%) + flash (95%) configuration yields 1,921, which is 12% higher than the IOPS/$ value from the best configuration without PCM.

Interestingly, the configuration with twice slower PCM write latency yields an IOPS/$ of 2,806, which is slightly higher than the baseline value (2,757). That may happen because the dynamic tiering algorithm is not perfect. With the static optimal placement method, 2× longer PCM write latency results in 3,216, which is lower than the original value of 3,220.

4.7 Summary of tiering simulation

Based on the results above, we observe that PCM can increase IOPS/$ value by 12% (bank) to 66% (telecommunication company) even assuming that PCM is 4× more expensive than flash. These results suggest that PCM has high potential as a new component for enterprise storage systems in a multi-tiered environment.

5 Workload simulation for server caching

Server-side caching is gaining popularity in enterprise storage systems today [5, 10, 11, 12, 16, 20, 24]. By placing frequently accessed data close to the application on a locally attached (flash) cache, network latencies are eliminated and speedup is achieved. The remote storage node benefits from decreased contention and the overall system throughput increases.

At first glance PCM SSD seems to be promising for server-side caching, considering the 16× faster read time compared to eMLC flash SSD. But given that PCM is more expensive and slower for write than flash, will PCM be a cost effective alternative? To address this question we use a second set of real-world traces to simulate caching performance. The prior set of traces used for tiered storage simulation could not be used to evaluate cache performance since the traces were summarized spatially and temporally at a coarse granularity. Three new IO-by-IO traces are used: 1) a 24 hour trace from a manufacturing company, 2) a 36 hours trace from a media company, and 3) a 24 hour trace from a medical service company. We chose three cache friendly workloads – highly skewed and read intensive – since our goal was to compare PCM and flash for server-side caching scenarios.

5.1 Cache simulation

We built a cache simulator using an LRU cache replacement scheme, 4 KiB page size, and write-through policy, which are the typical choices for enterprise server-side caching solutions. The simulator supports both single tier and hybrid (i.e. multi-tier) cache devices to test a configuration using PCM as a first level cache and flash as a second level cache. Our measurements (Table 2) are used for PCM and flash SSDs, and for networked storage

Table 3: Networked storage related parameters from [12]

Network base latency	8.2 µs / packet
Network data latency	1 ns / bit
File server fast read	92 µs / 4 KiB
File server slow read	7,952 µs / 4 KiB
File server write	92 µs / 4 KiB
File server fast read rate	90%

Table 4: Cache simulation parameters

	PCM	**eMLC**	**Net. Storage**
4 KiB R. Lat.	6.7 µs	108.0 µs	919.0 µs
4 KiB W. Lat.	128.3 µs	37.1 µs	133.0 µs
Norm. Cost	4	1	–

we use 919 µs and 133 µs for 4 KiB read and write, respectively. These numbers are based on the timing model parameters (Table 3) from previous work [12]; network overhead for 4 KiB is calculated as 41.0 µs (8.2 µs base latency + (4,096 × 8) bits × 1 ns), write time is 133 µs (write time 92 µs + network overhead 41 µs), and read time is 919 µs (90% × fast read time 92 µs + 10% × slow read time 7,952 µs + network overhead 41 µs).

The simulator captures the total number of read and write I/Os to the caching device and the networked storage separately, and then calculates average read latency as our evaluation metric; with write-through policy, write latency cannot be improved.

We vary the cache size from 64 GiB to a size that is large enough to hold the entire dataset. We then calculate the average read latency for all-flash and all-PCM configurations.

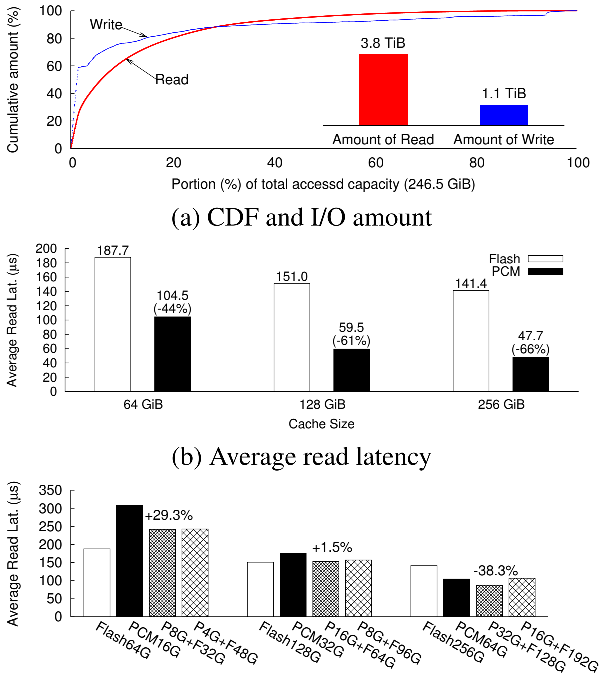
Next, we compare the cache performance for all-PCM, all-flash, and PCM and flash hybrid combinations having the same cost.

5.2 Result 1: Manufacturing company

The first trace is from the storage server of a manufacturing company, running an On-Line Transaction Processing (OLTP) database on a ZFS file system.

Figure 9(a) shows the cumulative distribution as well as the total amount of read and write I/O traffic for this workload. The total accessed capacity (during 24 hours) is 246.5 GiB, the total amount of read traffic is 3.8 TiB, and the total amount of write traffic is 1.1 TiB. The workload exhibits strong skew: 20% of the storage capacity receives 80% of the read traffic and 84% of the write traffic.

Figure 9(b) shows the average read latency (Y-axis) for flash and PCM with different cache sizes. From the results, we see that PCM can provide an improvement of 44–66% over flash. Note that this figure assumes equal amount of PCM and flash and hence the PCM caching solution results in 4 times higher cost than an all-flash setup (Table 4).



(a) CDF and I/O amount

(b) Average read latency

(c) Average read latency for even cost configurations

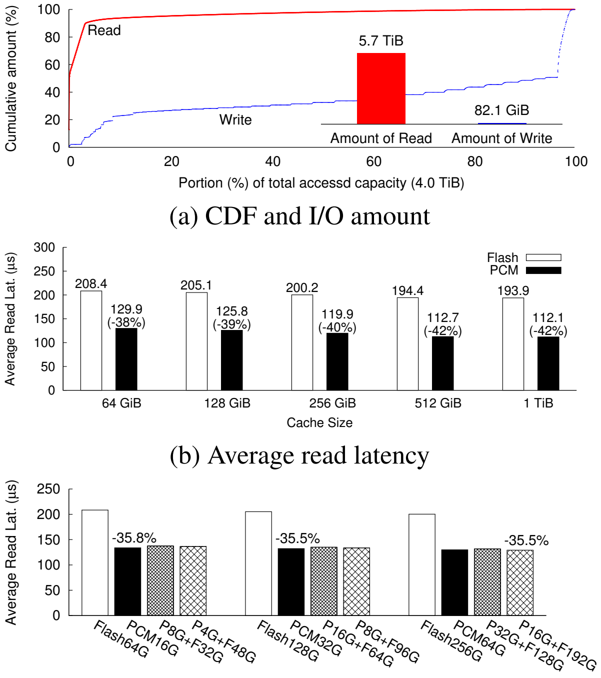
Figure 9: Cache simulation result for manufacturing company trace

Next, Figures 9(c) shows average read latency for cost-aware configurations. The results are divided into three groups. Within each group, we vary the ratio of PCM and flash while keeping the cost constant. For the first two groups, all-flash configurations (64 GiB, 128 GiB flash) show superior results to any configuration with PCM. For the third group (256 GiB flash), the 32 $GiBPCM + 128\ GiBflash$ combination shows about 38% shorter average read latency than an all-flash configuration.

5.3 Result 2: Media company

The second trace is from the storage server of a media company, also running an OLTP database.

The cumulative distribution and the total amount of read and write I/O traffic are shown in Figure 10(a). The total accessed storage capacity is 4.0 TiB, the total amount of read traffic is 5.7 TiB, and the total amount of write traffic is 82.1 GiB. This workload is highly skewed and read intensive. Compared to other workloads, this workload has a larger working set size and a longer tail,



(a) CDF and I/O amount

(b) Average read latency

(c) Average read latency for even cost configurations

Figure 10: Cache simulation result for media company trace

which results in a higher proportion of cold misses.

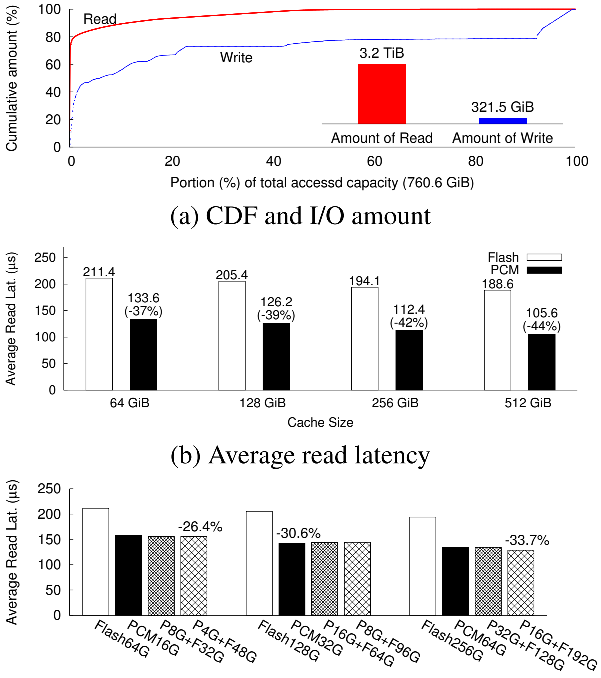
Figure 10(b) shows average read latency (Y-axis) for different cache configurations ranging from 64 GiB to 1 TiB. Because of the large number of cold misses, the improvements are less then those observed for the first workload: 38–42% shorter read latency than flash.

Figures 10(c) shows the simulation results for cost-aware configurations. Again, the results are divided into three groups. Within each group, we vary the ratio of PCM and flash while keeping the cost constant. Unlike the previous workload (manufacturing company), PCM reduces read latency in all three groups by about 35% compared to flash.

5.4 Result 3: Medical database

The last trace was captured from a front-line patient management system. Traces were captured over a period of 24 hours, and in total 760.6 GiB of storage space was touched. The amount of read traffic (3.2 TiB) is about 10× more than the amount of write traffic (321.5 GiB), and read requests are highly skewed as shown in Figure 11(a).

Figure 11(b) shows the aggregate I/O time (Y-axis) with 64 GiB to 512 GiB cache sizes. We observe that PCM can provide 37–44% shorter read latency than flash.



(a) CDF and I/O amount

(b) Average read latency

(c) Average read latency for even cost configurations

Figure 11: Cache simulation result for medical database trace

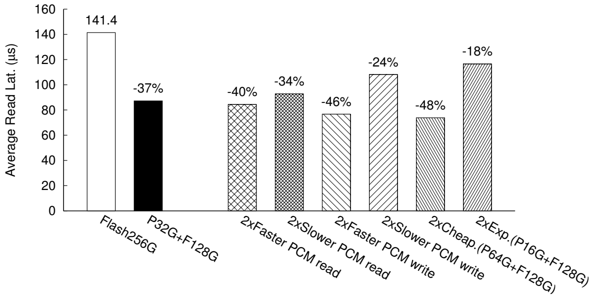


Figure 12: The average read latency for manufacturing company trace with varied PCM parameters

For the cost-aware configurations, PCM can improve read latency by 26.4–33.7% (Figure 11(d)) compared to configurations without PCM.

5.5 Sensitivity analysis for caching

Similar to the study of tiering in Section 4.6, we run sensitivity analysis for server caching as well. We test six variations of PCM parameters: (1) 2× shorter PCM read latency (4.35 µs), (2) 2× longer PCM read latency (13.4 µs), (3) 2× shorter PCM write latency (64.15 µs), (4) 2× longer PCM write latency (256.6 µs), (5) 2× cheaper normalized PCM cost (12), and finally (6) 2× more expensive normalized PCM cost (48). We pick the manufacturing company trace and its best configuration

(PCM 32 GiB + flash 128 GiB).

Figure 12 shows the simulated average read latencies for varied configurations. The same trend is shown as observed from the result for tiering (Figure 8); price creates the biggest impacts; even when performing half as well as our measured device, PCM still achieves 18–34% shorter average read latencies than all flash configuration.

5.6 Summary of caching simulation

Our cache simulation study with real-world storage access traces has demonstrated that PCM can improve aggregate I/O time by up to 66% (manufacturing company trace) compared to a configuration that uses the same size of flash. With cost-aware configurations, we show that PCM can improve average read latency up to 38% (again, manufacturing company trace) compared to the flash only configuration.

From our results, we observe that the result from the first workload (manufacturing) is different from the results of the second (media) and third (medical). While configurations with PCM offer significant performance improvement over any combination without PCM in the second and third workloads, we observe that that is true only for larger cache sizes in the first workload (i.e. Figures 9(c). This can be attributed to the varying degrees of skewing in the workloads. The first workload exhibits less skew (for read I/Os) than the second and third workloads and hence has a larger working-set size. As a result, by increasing the cache size to capture the entire working set for the first workload (data point PCM 32 GiB + flash 128 GiB), we are eventually able to achieve a configuration that captures the active working-set.

These results point to the fact that PCM-based caching options are a viable, cost-effective option to flash-based server-side caches, given a fitting workload profile. Consequently, analysis of workload characteristics is required to identify critical parameters such as proportion of writes, skew and working set size.

6 Limitations and discussion

Our study into the applicability of PCM devices in realistic enterprise storage settings has provided several insights. But we acknowledge that our analysis does have several limitations: First, since our evaluation is based on a simulation, it may not accurately represent system conditions. Second, from our asynchronous I/O test (see section 3.2), we observe that the prototype PCM device does not exploit I/O parallelism much, unlike the eMLC flash SSD. This means that it may not be fair to say that the PCM SSD is 16× faster than the eMLC SSD for read, because the eMLC SSD can handle multiple read I/O requests concurrently. It is a fair concern if we ignore the capacity of the SSDs. The eMLC flash SSD has 1.8 TiB capacity while the PCM SSD has only 64 GiB capacity. We assume that as the capacity of PCM SSD increases, its parallel I/O handling capability will increase as well. Finally, in order to understand long-term architectural implications, longer evaluation runs may be required for performance characterization.

In this study, we approach PCM as storage rather than memory, and our evaluation is focused on average performance improvements. However, we believe that the PCM technology may be capable of much more. As shown in our I/O latency measurement study, PCM can provide well-bounded I/O response times. These performance characteristics will prove to be very useful to provide Quality of Service (QoS) and multi-tenancy features. We leave exploration of these directions to future work.

7 Conclusion

Emerging workloads seem to have an ever-increasing appetite for storage performance. Today, enterprise storage systems are actively adopting flash technology. However, we must continue to explore the possibilities of next generation non-volatile memory technologies to address increasing application demands as well as to enable new applications. As PCM technology matures and production at scale begins, it is important to understand its capabilities, limitations and applicability.

In this study, we explore the opportunities for PCM technology within enterprise storage systems. We compare the latest PCM SSD prototype to an eMLC flash SSD to understand the performance characteristics of the PCM SSD as another storage tier, given the right workload mixture. We conduct a modeling study to analyze the feasibility of PCM devices in a tiered storage environment.

8 Acknowledgments

We first thank our shepherd Steven Hand and anonymous reviewers. We appreciate Micron for providing their PCM prototype hardware for our evaluation study and answering our questions. We also thank Hillery Hunter, Michael Tsao, and Luis Lastras for helping our experiments, and Paul Muench, Ohad Rodeh, Aayush Gupta, Maohua Lu, Richard Freitas, Yang Liu for their valuable comments and help.

References

[1] AKEL, A., CAULFIELD, A. M., MOLLOV, T. I., GUPTA, R. K., AND SWANSON, S. Onyx: a protoype phase change memory storage array. In *Proceedings of the 3rd USENIX conference on Hot topics in storage and file systems* (Berkeley, CA, USA, 2011), HotStorage'11, USENIX Association, pp. 2–2.

[2] AKERMAN, J. Toward a universal memory. *Science 308*, 5721 (2005), 508–510.

[3] ATHANASSOULIS, M., BHATTACHARJEE, B., CANIM, M., AND ROSS, K. A. Path Processing using Solid State Storage. In *Proceedings of the 3rd International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures (ADMS 2012)* (2012).

[4] BEDESCHI, F., RESTA, C., ET AL. An 8mb demonstrator for high-density 1.8v phase-change memories. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on* (2004), pp. 442–445.

[5] BYAN, S., LENTINI, J., MADAN, A., PABON, L., CONDICT, M., KIMMEL, J., KLEIMAN, S., SMALL, C., AND STORER, M. Mercury: Host-side flash caching for the data center. In *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on* (2012), pp. 1–12.

[6] DELL. 300 gb 15,000 rpm serial attached scsi hotplug hard drive for select dell poweredge servers / powervault storage.

[7] DELL. Dell Enterprise Hard Drive and Solid-State Drive Specifications. `http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/enterprise-hdd-sdd-specification.pdf`.

[8] DELL. LSI Logic Nytro WrapDrive BLP4-1600 - Solid State Drive -1.6 TB - Internal. `http://accessories.us.dell.com/sna/productdetail.aspx?sku=A6423584`.

[9] EMC. FAST: Fully Automated Storage Tiering. `http://www.emc.com/storage/symmetrix-vmax/fast.htm`.

[10] EMC. XtreamSW Cache: Intelligent caching software that leverages server-based flash technology and write-through caching for accelerated application performance with data protection. `http://www.emc.com/storage/xtrem/xtremsw-cache.htm`.

[11] FUSION-IO. ioTurbine: Turbo Boost Virtualization. `http://www.fusionio.com/products/ioturbine`.

[12] HOLLAND, D. A., ANGELINO, E., WALD, G., AND SELTZER, M. I. Flash caching on the storage client. In *Proceedings of the 11th USENIX conference on USENIX annual technical conference* (2013), USENIXATC'13, USENIX Association.

[13] HOYA, K., TAKASHIMA, D., ET AL. A 64mb chain feram with quad-bl architecture and 200mb/s burst mode. In *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International* (2006), pp. 459–466.

[14] IBM. IBM FlashSystem 820 and IBM FlashSystem 720. `http://www.ibm.com/systems/storage/flash/720-820`.

[15] IBM. IBM System Storage DS8000 Easy Tier. `http://www.redbooks.ibm.com/abstracts/redp4667.html`.

[16] IBM. IBM System Storage DS8000 Easy Tier Server. `http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp5013.html`.

[17] IBM. IBM XIV Storage System. `http://www.ibm.com/systems/storage/disk/xiv`.

[18] KIM, D., LEE, S., CHUNG, J., KIM, D. H., WOO, D. H., YOO, S., AND LEE, S. Hybrid dram/pram-based main memory for single-chip cpu/gpu. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE* (2012), pp. 888–896.

[19] KIM, J. K., LEE, H. G., CHOI, S., AND BAHNG, K. I. A pram and nand flash hybrid architecture for high-performance embedded storage subsystems. In *Proceedings of the 8th ACM international conference on Embedded software* (New York, NY, USA, 2008), EMSOFT '08, ACM, pp. 31–40.

[20] KOLLER, R., MARMOL, L., SUNDARARAMAN, S., TALAGALA, N., AND ZHAO, M. Write policies for host-side flash caches. In *Proceedings of the 11th USENIX conference on File and Storage Technologies* (2013), FAST'13, USENIX Association.

[21] LEE, B. C., IPEK, E., MUTLU, O., AND BURGER, D. Architecting phase change memory as a scalable dram alternative. In *Proceedings of the 36th annual international symposium on Computer architecture* (New York, NY, USA, 2009), ISCA '09, ACM, pp. 2–13.

[22] LEE, K.-J., ET AL. A 90nm 1.8v 512mb diode-switch pram with 266mb/s read throughput. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International* (2007), pp. 472–616.

[23] MOGUL, J. C., ARGOLLO, E., SHAH, M., AND FARABOSCHI, P. Operating system support for nvm+dram hybrid main memory. In *Proceedings of the 12th conference on Hot topics in operating systems* (Berkeley, CA, USA, 2009), HotOS'09, USENIX Association, pp. 14–14.

[24] NETAPP. Flash Accel software improves application performance by extending NetApp Virtual Storage Tier to enterprise servers. `http://www.netapp.com/us/products/storage-systems/flash-accel`.

[25] PURESTORAGE. FlashArray, Meet the new 3rd-generation FlashArray. `http://www.purestorage.com/flash-array/`.

[26] QURESHI, M. K., FRANCESCHINI, M. M., JAGMOHAN, A., AND LASTRAS, L. A. Preset: improving performance of phase change memories by exploiting asymmetry in write times. In *Proceedings of the 39th Annual International Symposium on Computer Architecture* (Washington, DC, USA, 2012), ISCA '12, IEEE Computer Society, pp. 380–391.

[27] QURESHI, M. K., SRINIVASAN, V., AND RIVERS, J. A. Scalable high performance main memory system using phase-change memory technology. In *Proceedings of the 36th annual international symposium on Computer architecture* (New York, NY, USA, 2009), ISCA '09, ACM, pp. 24–33.

[28] RAOUX, S., BURR, G., BREITWISCH, M., RETTNER, C., CHEN, Y., SHELBY, R., SALINGA, M., KREBS, D., CHEN, S.-H., LUNG, H. L., AND LAM, C. Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development 52*, 4.5 (2008), 465–479.

[29] SIE, C. *Memory Cell Using Bistable Resistivity in Amorphous As-Te-Ge- Film.* Iowa State University, 1969.