Onyx: A Protoype Phase Change Memory Storage Array

Ameen Akel Adrian M. Caulfield Todor I. Mollov
Rajesh K. Gupta Steven Swanson
Computer Science and Engineering
University of California, San Diego

Abstract

We describe a prototype high-performance solid-state drive based on first-generation phase-change memory (PCM) devices called *Onyx*. Onyx has a capacity of 10 GB and connects to the host system via PCIe. We describe the internal architecture of Onyx including the PCM memory modules we constructed and the FPGA-based controller that manages them. Onyx can perform a 4 KB random read in 38 μs and sustain 191K 4 KB read IO operations per second. A 4 KB write requires 179 μs. We describe our experience tuning the Onyx system to reduce the cost of wear-leveling and increase performance. We find that Onyx out-performs a state-of-the-art flash-based SSD for small writes ($<$ 2 KB) by between 72 and 120% and for reads of all sizes. In addition, Onyx incurs 20-51% less CPU overhead per IOP for small requests. Combined, our results demonstrate that even first-generation PCM SSDs can out-perform flash-based arrays for the irregular (and frequently read-dominated) access patterns that define many of today's "killer" storage applications. Next generation PCM devices will widen the performance gap further and set the stage for PCM becoming a serious flash competitor in many applications.

1 Introduction

Storage devices based on non-volatile, solid-state memories are rewriting the rules governing the relationship between storage devices and the rest of computer systems. Flash-based SSDs are in the vanguard of this change, but faster, more reliable, and less idiosyncratic technologies are on the horizon. Of these advanced non-volatile memories, phase-change memory (PCM) is the closest to seeing use in real storage products.

PCM promises increased speed, better scaling, and, eventually, better density than flash memory. Most importantly, it does not suffer from flash's crippling inability to perform in-place updates of data that, in turn, necessitate complex management and wear-leveling systems that increase latency for small requests and, for high-end PCIe-attached SSDs, increase CPU and power overheads. The improved performance and reduced complexity that PCM provides will make it a potent competitor to flash memory in the coming years.

However, PCM has its own idiosyncrasies and designing a PCM-based storage array will present its own set of challenges. We have constructed a first-generation PCM-based SSD called *Onyx* that allows us to grapple with these issues first hand. Onyx attaches to the host system via PCIe and applications access it via a highly-optimized block driver that eliminates most software overheads and allows for high concurrency among accesses. Onyx has a usable capacity of 8 GB with 2 GB of storage for error correction or other meta data (2 bytes per 8 byte word).

This paper describes Onyx's architecture and the changes we have made to its design to improve performance. Onyx can sustain just over 1.1 GB/s for reads, and its write performance is 34% better than expected based on the specifications of the PCM components it contains. Onyx uses start-gap wear leveling [4] and we explore the effect of different start-gap parameters on performance. Our results show that Onyx is faster for small write requests than a state-of-the-art flash-based SSD, but that the poor write performance of currently-available PCM limits throughput for large writes. For reads, Onyx outperforms the flash-based SSD by between 11 and 430%, depending on access size.

We also demonstrate that PCM reduces CPU overheads compared to a flash-based storage array, because PCM's simpler interface allows for a simpler driver. This frees the CPUs to handle more useful computation and reduces the power overheads of IO as well.

These results show that (assuming PCM scaling projections hold) PCM-based storage array architectures will be a competitive alternative to flash-based SSDs. They will be of particular value in applications, like high-performance caching systems and key-value stores, that require high performance reads and small writes.

The remainder of this paper is organized as follows. Section 2 describes the Onyx hardware and its software

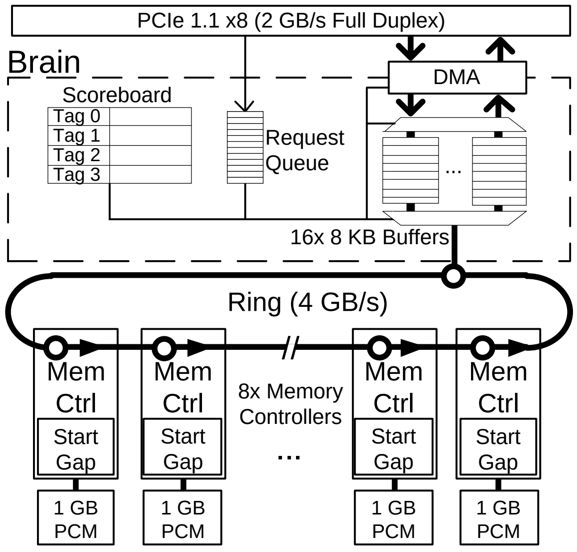

Figure 1: **Onyx's high-level architecture** The Onyx main controller, or *brain*, allows it track up to 64 in-flight requests at once. Onyx stripes large requests across multiple controllers in 4 KB slices.

stack. Section 3 describes how we refined Onyx's PCM controller to improve performance and compares Onyx to an existing PCIe-attached SSD. Finally, Section 4 presents our conclusions.

2 Onyx

This section describes the Onyx storage array. We begin by briefly describing Onyx's high-level architecture. Then, we describe the PCM DIMM memory modules and the controller that provides access to them.

2.1 System overview

Figure 1 shows the high-level organization of Onyx. The design is based on the Moneta [3] SSD that used DRAM to emulate next-generation non-volatile memories. Onyx replaces the DRAM with real PCM, but retains Moneta's highly-optimized software stack to minimize latency and maximize concurrency. The hardware includes a "brain" which handles high-level scheduling and communicates with the banks of PCM via a ring network. The brain contains a scoreboard for tracking outstanding accesses, a DMA controller, transfer buffers, and an interface to the ring. The ring connects to eight, 1.25 GB banks of PCM memory.

Onyx connects to the host system via an 8-lane PCIe 1.1 interface that provides a 2 GB/s full-duplex connection (4 GB/s total). The baseline design supports 64 concurrent, outstanding requests, each identified by a unique tag. The prototype is implemented on a BEE3 FPGA prototyping system [2] developed as part of the RAMP project [6]. The system contains four FPGAs connected in a ring. Each FPGA has two banks of two DDR2 DIMM slots (four DIMMs per FPGA). Onyx runs at 250 MHz. More information about the Onyx system architecture can be found in [3].

2.2 The PCM Module

The PCM devices in Onyx connect to the FPGA via the DDR2 DIMM sockets and a custom-built memory module.

The PCM devices

Onyx uses Micron's first-generation "P8P" 16 MB PCM devices (part # NP8P128A13B1760E). The electrical interface and command set they provide is similar to a NOR flash device with the important difference that it supports writes of arbitrary data at the byte level as opposed to separate erase and program operations with different granularities. Each chip has 16 data lines and 23 address lines, in addition to 5 other signaling pins (e.g., Write Enable, Output Enable, and Chip Enable).

To perform a read, the controller must place the PCM devices into Read Array mode and set the address lines to those of the requested data. After a fixed delay, the data appears in the chip's internal buffers. The controller then clocks out the data onto the data lines.

Onyx uses high-bandwidth Buffered Write operations for writing. These require the controller to fill the PCM's internal write buffer. The controller then instructs the PCM to commit the data to the non-volatile array. The controller detects the successful completion of the write by polling the PCM's status register.

Based on datasheet timing, a maximum size, 16 byte, read requires 314 ns to complete. A maximum size write (64 bytes) takes 120 μs for arbitrary data (writes of all zeros are faster). This gives theoretical peak bandwidths for reads and writes (of random data) of 48.6 MB/s and 0.5 MB/s, respectively, per chip.

Wear out is a concern for PCM devices, and the datasheet for the P8P devices gives a lifetime of 1 million programs per cell. Discussions with the manufacturer suggest that this value is not directly comparable to the lifetime values given for flash devices. The PCM lifetime estimate is the number of programs per cell before the first bit error appears in a large population of devices without ECC. By contrast, flash memory durability ratings are usually given as the number of program/erase cycles until a given ECC scheme will no longer be able to correct the errors that appear.

The PCM DIMM

We use the P8P devices to build PCM memory modules, or *PCM DIMMs* (Figure 2). The PCM DIMMs fit into standard DDR2 DRAM DIMM slots, but they are

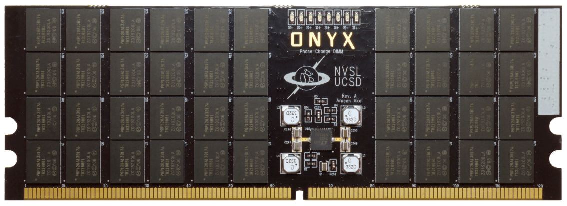

Figure 2: **The Onyx PCM DIMM** The PCM DIMM contains eight indentical ranks of five chips. Ranks share data lines, and all chips share address lines.

slightly taller. While they are mechanically compatible with DDR2 DIMM sockets, and the power and ground pins are in the same locations, the signaling interface, and pin assignments are completely different to accommodate the P8P's NOR-like interface and slower speed.

Each PCM DIMM contains 40 PCM devices arranged into eight ranks of five chips. The chips within each rank act in concert (much as DRAM chips do in a normal DIMM) to provide an 80 bit wide interface. Sixty-four of the bits are data, the remaining 16 are available for ECC or other meta data. The aggregate capacity of a single PCM DIMM is 640 MB (512 MB without the meta data section).

2.3 The PCM DIMM controller

Onyx contains eight PCM DIMM controllers, and each manages a pair of PCM DIMMs. From the controller's perspective the two PCM DIMMs appear as 16 independent ranks.

Figure 3 shows the internal architecture of the PCM DIMM controller. Requests arrive from the ring interface and pass into the wear-leveling module (see below). A scoreboard tracks outstanding requests, an active list tracks request ordering, and a request scheduler maps requests to available ranks to maximize parallelism. Once the scheduler assigns the request to a rank, the PCM control unit issues low-level commands to the rank, sends completion notifications back to the request scheduler, and forwards data (for reads) back to the ring. The request completion module sends request completion notifications back to the "brain."

The PCM DIMM controller can signal that a request is complete at two points: *Late completion* occurs when write to the PCM DIMM is completely finished. *Early completion* occurs when all the store data has arrived at the buffers in the memory controller. Early completion allows Onyx to hide most of the write latency, but it raises the possibility of data loss if power fails before the write to the PCM completes. To guarantee durability, the PCM DIMMs include large capacitors that will supply enough energy to complete the operation. State-

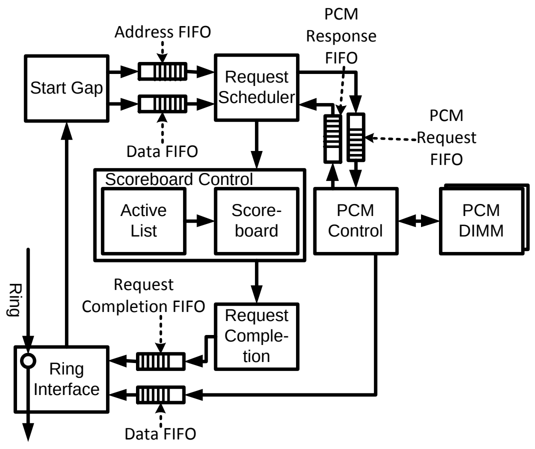

Figure 3: **The PCM DIMM controller** Requests originate in the Request Scheduler. The Request Completion module generates completion signals.

of-the-art flash-based SSDs routinely use early completion to reduce write latency. With early completion, the peak bandwidth per PCM DIMM pair is 156 MB/s for reads and 47.1 MB/s for writes.

To avoid uneven wear, the Onyx controller incorporates the first real-system implementation of the start-gap wear leveling [4] scheme. Start-gap works by slowly rotating the mapping between storage addresses and physical 4 KB rows of PCM memory. The key parameter in the start-gap scheme is the "gap write interval," G. If the PCM memory contains R rows, then after $R \times G$ writes, start-gap will have shifted all the addresses by 1 row (i.e., if address a initially corresponded to physical memory row p, a will now refer to physical row $p + 1$). By default, we use a gap write interval of 128. We discuss tuning this parameter in Section 3.

3 Onyx performance

This section evaluates the raw performance of Onyx using a combination of microbenchmarks and simple database workloads. We also compare its performance to a state-of-the-art flash-based SSD from FusionIO and the original Moneta storage array.

3.1 Raw performance

Figure 4 measures the bandwidth for random reads, writes, and a combination of 50% reads/writes for a range of access sizes. We collected the data with XDD [7] (a flexible IO workload generator) using 16 threads. The data show that for large requests, Onyx can sustain over 1.1 GB/s for reads and 470 MB/s for writes with early completion notifications. The read bandwidth matches projections using the datasheet latencies in Sec-

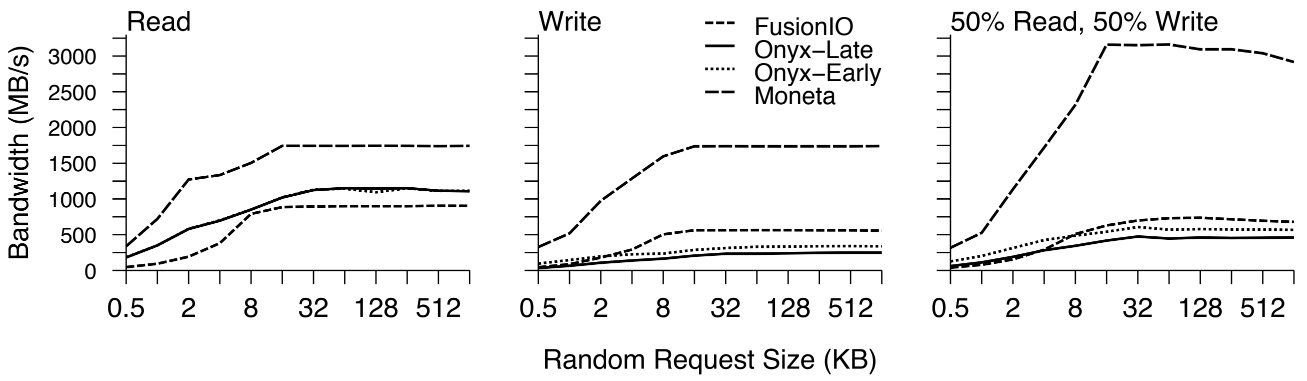

Figure 4: **Onyx basic IO performance** For reads and small writes, Onyx consistently out-performs FusionIO, but for large writes FusionIO's greater internal bandwidth leads to better performance. With early write completion, Onyx's write performance improves for both small and large requests.

tion 2.2, but the write bandwidth exceeds these projections by 34%.

The graphs compare the performance differences between early and late write completion for both write-only and mixed workloads. Early completion improves write performance by between 32% for large requests and 174% for small requests. We use early completion in the remainder of the paper.

The graphs also compare Onyx's performance to that of an 80 GB FusionIO ioDrive [1] and the Moneta SSD. For reads, Onyx consistently outperforms the ioDrive, and the gap is especially wide for small requests: For 512 byte requests – Onyx can sustain 478K IOPS compared to the ioDrive's 90K IOPS. We believe the improved performance is due in large part to absence of a complex flash translation layer (FTL) in Onyx. The FTL adds software overhead to each access to the ioDrive, limiting throughput on small requests.

For writes, Onyx outperforms the ioDrive small requests (because of smaller per-operation overheads) but is slower for large requests. The ioDrive's aggregate write bandwidth is higher than Onyx's: A single flash die can sustain programming rates of between 5 and 10 MB/s. Analyzing our ioDrive and its datasheet suggests that it contains 96 flash dies for a total of at least 480 MB/s. Each of the eight PCM controllers in Onyx can sustain no more than 47.1 MB/s, or 340 MB/s in aggregate. Next generation PCM devices will sustain up to 3.76 MB/s per chip, and should allow an Onyx-like system to outperform the ioDrive. Increasing the number of PCM DIMM controllers in Onyx would likely also improve performance.

The results for Moneta provide a projection for Onyx's performance with future-generations of PCM. Faster PCM devices will roughly double read performance and increase write performance by between 5 and $6\times$.

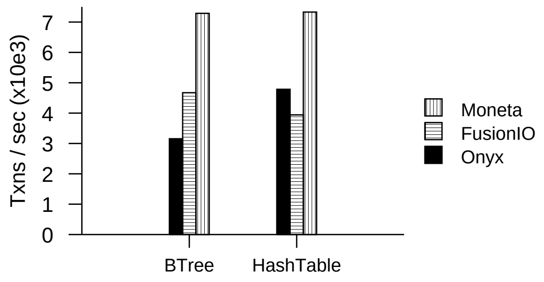

Figure 5: **BDB Benchmark Results** Onyx outperforms the ioDrive for the hash table version of BerkeleyDB but under-performs for the b-tree.

We use BerkeleyDB to compare application-level performance on Onyx and the ioDrive. Figure 5 contains results for a pair of BerkeleyDB benchmarks that use a hash table or B-tree to store the database tables. Our BerkeleyDB configuration includes full transaction support and performs synchronous IO. The workload transactionally swaps the values associated with two keys in the database. For each storage device we use the number of threads that maximizes throughput (one for FusionIO, four for Onyx, and two for Moneta). The performance for Onyx compared to the ioDrive is mixed: For the hash table, Onyx outperforms the ioDrive by 21%, but for B-tree, the ioDrive delivers 48% greater operations per second. We are still investigating the reason for this variability.

3.2 CPU overhead

A less obvious advantage of Onyx over the ioDrive is the reduction in CPU overhead. The ioDrive's complex driver plays an important role in managing the flash array, and that requires substantial CPU resources for each IO operation. For small requests, this means that a system with Onyx spends between 20-51% less CPU time performing IO operations. This has two effects: First, it frees up CPUs for other, more useful work. Second, it reduces the overall energy requirement of the storage system, improving efficiency and/or offsetting the increased energy cost of writing to PCM memory. More broadly, reduced CPU overhead and energy consumption combined with increased performance mean that Onyx-like arrays will significantly alter what constitutes a balanced computing system at both the server and cluster levels.

3.3 Wear-leveling

The start-gap wear leveling scheme that Onyx uses requires the memory controller to periodically copy a row of memory from one location to another. Recent work [5] has defined the "line vulnerability factor" as the number of writes that might go to an address before start-gap remaps it. The factor is the product R$\times$G described in Section 2. Smaller gap write intervals result in lower vulnerability factors, but also introduce overhead in the form of the extra PCM accesses required to rotate the data to match the new addresses

Figure 6 measures these effects. The horizontal axis varies the gap write interval. The left hand vertical axis measures sustained bandwidth for 4 KB writes, while the right hand vertical axis measures write latency. The impact on latency is smaller than on bandwidth because it is usually possible to hide the latency of shifting the gap.

By default, Onyx uses a gap write interval of 128, which gives a line vulnerability factor of 32 million. The manufacture rates our devices for 1 million writes before the first error occurs, so it is possible that a pathological access pattern could cause significant damage to one row. However, recent work [5] describes how to vary the interval dynamically to prevent this. In addition, breaking each bank of PCM into multiple start-gap domains would further reduce the vulnerability factor. We are considering these and other changes to improve wear-leveling in a future version of Onyx.

4 Conclusion

Onyx provides a glimpse into the future of solid-state drives and identifies some of challenges designers will face as they incorporate novel, non-volatile memories into storage systems. Our experience designing and using Onyx shows that PCM can provide good performance, but that several challenges remain. In particular, the poor write performance of individual PCM devices limits overall write performance for Moneta. As manufacturers work toward solving that problem, we will also need to refine the PCM controller design to exploit more parallelism.

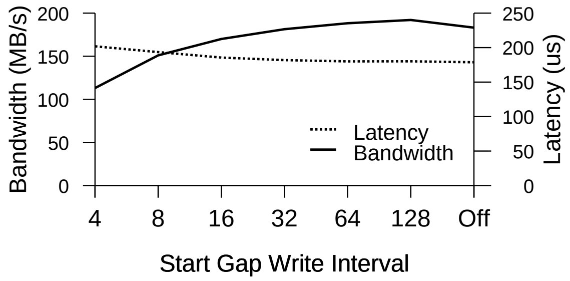

Figure 6: **Start-gap's impact** As the gap write intervals grows, wear leveling overhead drops and bandwidth increases.

Despite these challenges, Onyx shows that phase-change memory has a potentially bright future as a storage technology. For emerging data-intensive applications that require large amount of irregular IO (e.g., large graph computations), PCM-based SSDs can already outperform their flash-based counterparts. As performance and density improves, the benefits they can offer will only increase.

Acknowledgements

This work was supported, in part, by financial and equipment donations from Micron Inc. In particular, we would like to thank Clifford Smith, Mark Leinwander, Sean Eilert, and Brian Bradford for their technical guidance in designing the PCM DIMM.

References

[1] http://www.fusionio.com/.

[2] http://www.beecube.com/platform.html.

[3] A. M. Caulfield, A. De, J. Coburn, T. Mollov, R. Gupta, and S. Swanson. Moneta: A high-performance storage array architecture for next-generation, non-volatile memories. In *Proceedings of The 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, 2010.

[4] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 14–23, New York, NY, USA, 2009. ACM.

[5] M. K. Qureshi, A. Seznec, L. A. Lastras, and M. M. Franceschini. Practical and secure pcm systems by online detection of malicious write streams. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 478 –489, feb. 2011.

[6] The ramp project. http://ramp.eecs.berkeley.edu/index.php.

[7] Xdd version 6.5. http://www.ioperformance.com/.