ABSTRACT

Solid State Drives (SSDs) are an important component of secondary storage systems. While Hard Disk Drives (HDDs) are cheaper per gigabyte, SSDs are cheaper per random I/O per second. A variety of solid-state technologies are being developed with NAND flash being the most mature, and Phase Change Memory (PCM) beginning to enter the marketplace. Compared with flash, PCM has finer-grained addressability and higher write endurance. PCM is also expected to offer lower read and write response times.

In this work we study the use of solid-state storage in latency-bound applications, a type of workload that can benefit from the characteristics of Flash and PCM technologies. We identify graph processing and Resource Description Framework (RDF) query processing as candidate applications. Using an early PCM prototype device, we demonstrate the benefits of PCM for this workload and compare with a flash device. Moreover, we describe Pythia, a prototype RDF repository designed for Solid State Storage. Using Pythia we investigate whether the predicted benefits for this type of workload can be achieved in a properly designed RDF repository.

1. INTRODUCTION

Flash devices have superior random read performance compared to magnetic hard-drives but suffer from several limitations. First, there is a significant asymmetry in read and write performance. Second, only a limited number of updates can be applied on a flash device before it becomes unusable; this number is decreasing with newer generations of flash [1]. Third, writing on flash not only is much slower than reading and destructive of the device, but it has proven to interfere with the redirection software layers, known as Flash Translation Layers (FTL) [6].

PCM addresses some of these challenges. The endurance of PCM cells is significantly higher than NAND flash [2], although still not close to that of DRAM. Unlike NAND flash, PCM does not require the bulk erasure of large memory units before it can be rewritten. Moreover, while cost is still uncertain, for our purposes, we assume normal cell size competitiveness and standard volume economics will apply to this technology as it ramps into high volume.

The most pronounced benefit of solid state storage over hard disks is the difference in response time for random accesses. Hence, we identify dependent reads as an access pattern that has the potential for significant performance gains. Latency-bound applications like path processing [7] in the context of graph processing, or RDF-data processing are typical examples of applications with such access patterns. The Resource Description Framework (RDF) [3] data model is widely adopted for several on-line, scientific or knowledge-based datasets because of its simplicity in modeling and the variety of information it can represent.

We find that PCM-based storage is an important step towards better latency guarantees with minor bandwidth penalties for random I/O, and we identify a concurrency trade-off between maximizing bandwidth and minimizing latency. In order to measure the performance benefit (decrease in response time) for long path queries we implement a simple benchmark and we compare the response times when using Flash and PCM. We observe that PCM can yield 1.5x to 2.5x smaller response times for any bandwidth utilization without any graph-aware optimizations. We take this observation one step further and we design a new data layout suitable for RDF data optimized for a solid state storage layer. The proposed layout increases the locality of related information and decreases the cost of graph traversals by storing more linkage information (i.e., metadata about how to navigate faster when a graph is traversed).

In this short paper, which is based on recently published work [5], we show the benefits of path processing applications over data that is resident in solid state storage. First, we present a custom graph benchmark that is used to highlight the differences between two solid state technologies through two representative devices: a state-of-the-art flash device and an enterprise-level PCM prototype provided to us by Micron. Second, we develop a prototype RDF repository to show the benefits that RDF processing can have if it adopts PCM storage.

2. PATH PROCESSING

Current PCM prototypes behave as a “better” flash [9], in the sense that they have faster and more stable reads. We argue that the best way to make the most of this behavior is in the domain of applications with dependent reads. Hence, we create a simple benchmark that performs path traversals over randomly created graphs to showcase the potential benefits using PCM as secondary storage for such applications.

We create a benchmark that models path traversal queries by graph traversal over a custom build graph. The graph is stored in
RDF-tuples to be stored as a tuple with variable length in a database page containing many such tuples. Figure 2 shows how an RDF-tuple is laid out within a page.

![Figure 2: RDF-tuple layout](image)

We experiment with Pythia and we corroborate the performance benefit achieved in the simple benchmark. Pythia for both devices (Flash and PCM) scales close to linearly until 4 threads (7KQ/s for PCM and 4.5KQ/s for flash leading to 1.56x speedup), while for higher number of threads the PCM device is more stable showing speedup from 1.8x to 2.6x. Last but not least, we compare the performance of Pythia repository against the state of the art RDF-3X repository in a limited set of tests and we find that Pythia is competitive with RDF-3X.

4. CONCLUSIONS

We have described our experiences and observations of a real PCM prototype. At least for the near future, PCM-based devices are going to be “better” flash devices, rather than a new incomparably fast and more efficient storage technology. The limits of interconnecting interfaces, and the overhead of the software stack used to access the devices, are significant determinants of overall system performance. As new generations of PCM become available, and as new software technologies are developed to minimize overheads, PCM device performance and access granularity constraints are likely to improve.

5. ACKNOWLEDGMENTS

The authors would like to thank Michele Tsao of IBM T.J. Watson Research for his help in setting up the machine used in the experiments, and Micron Inc. for providing access to an early PCM prototype.

6. REFERENCES