Seminar Report

Non-Volatile Storage

Mathias Meinschad, Sebastian Waldhart

1 June 2016

Abstract

The bottleneck in most modern computers is the storage (often called the "I/O gap"). With the rise of Storage Class Memory (SCM) this condition may be completely inverted. In this paper we will discuss the article "Non-Volatile Storage" [3] which is about the benefits of SCM and also explains some difficulties that arise with this new technology.

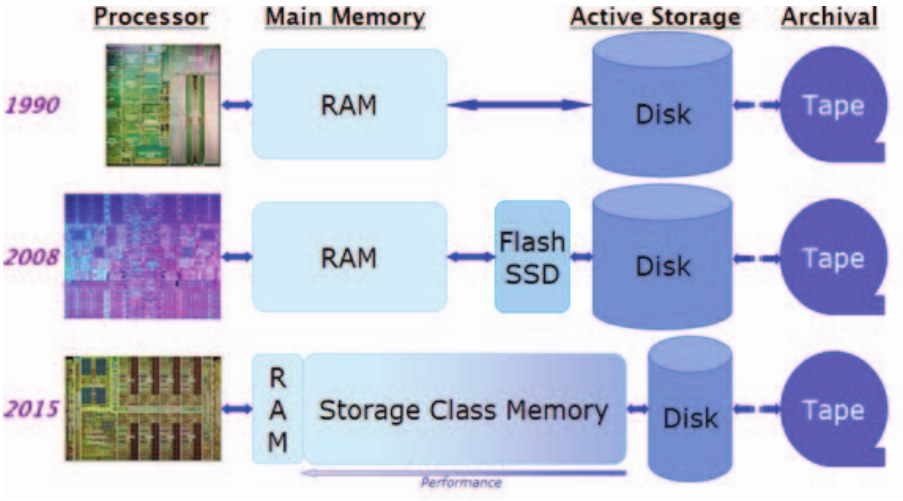

Figure 1: Evolution of Memory Hierarchy. This image shows where SCM may fit into computer hardware. [2]

1 Introduction

1.1 What is Storage Class Memory?

> "The arrival of high-speed, non-volatile storage devices, typically referred to as storage class memories (SCM), is likely the most significant architectural change datacenter and software designers will face in the foreseeable future." [3]

In modern computers there is a distinction between storage (non-volatile, slow, cheap) and memory (volatile, fast, expensive) devices. As the term already indicates, SCM describes storage class devices with some benefits of memory. In other words: SCM blurs the line between those two device categories.

The most visible type nowadays are PCIe SSDs (which already show the impact of the topics we'll discuss in this paper), but there are other technologies, which belong to the category of SCM and will be available in near future. More Information about these technologies can be found at *Huang* [1] and *Greengard* [2].

Unfortunately, current PCIe SSDs are not as cheap as storage and cost $ 3 000 - $ 5 000. This easily outweighs the cost of a CPU. Furthermore, it is important to note the price being 25 times higher than for traditional spinning disks. This means underutilizing a SCM can also been seen as a waste of money.

1.2 The I/O Gap

While the computational power of processors have improved as predicted by Moore's Law, storage performance has remained nearly unchanged. This has happened because of physical properties like the rotational velocity of the disks.

Due that reason the I/O gap widened throughout the 1990s and early 2000s. To overcome this gap hardware designers started to use caching. As the gap

widened every year, caching started to extend across all layers (processors cache the contents of RAM, operating systems cache entire disk sectors, etc.).

There are some other techniques that trade CPU time for better disk performance, for example zRam.

Spinning disks can perform about 100 IOPS (I/O operations per second) while SCMs deliver an astounding performance of 100.000 IOPS. Thus, they have the potential to totally shift the I/O gap. As a consequence the CPU itself will be the bottleneck of the system.

1.3 Age Old Assumptions

> "Processing power is in fact so far ahead of disk latencies that prefetching has to work multiple blocks ahead to keep the processor supplied with data. [...] Fortunately, modern machines have sufficient spare cycles to support more computationally demanding predictors than anyone has yet proposed." [4]

Most software developers learn that I/O operations are slow and the CPU is fast. This assumption leads to many design decisions. Even in CPU scheduling mechanisms of operating systems, these assumptions play a central role. However, this won't be true anymore, once SCM gets more common. Therefore it is required to rethink most of our software, otherwise it will become very inefficient.

2 Utilizing SCMs

When using SCM you might stumble over some issues. *Nanavati* [3] describes the problems he experienced in four years while building scalable enterprise storage systems using SCMs.

2.1 Everything Will Just Get Faster, Right?

One may think of SCMs as a fast storage, so they could simply replace the spinning disks.

The first problem that comes across are the slow SATA drivers, which are typically used for storage nowadays. The solution is simple: The device must be moved to the PCIe bus.

A single SCM needs 4 to 8 lanes on the PCIe bus, so the number of parallel devices is limited already by the hardware. While this might not be a problem for most consumers, it is definitely a concern for servers and data-centers.

As stated above, the next problem is that most software is based on the assumption that storage is slow. This will lead to trouble when a big collection of data is processed. For such a task the data is loaded from the disk into a buffer and then gets fetched by a thread. As long as the computation needs less time than loading the data this approach works perfectly well, but with SCMs this won't work any longer. An additional consequence might be that the operating system will start swapping the data back to storage as the RAM gets too occupied.

2.2 I/O-Centric Scheduling

> "For a long time, interrupt-driven I/O has been the model of choice for CPU-disk interaction. This was a direct consequence of the mismatch in their speeds: for a core running at a few gigahertz, servicing an interrupt every few milliseconds is fairly easy. A single core can service tens or hundreds of disks without getting overwhelmed and missing deadlines." [3]

The interrupt-driven approach may slow the system down. I/O will be much faster, thus the number of interrupts will grow. Therefore the interrupt comes with some overhead for the CPU caused by the context switch.

I/O scheduling can be improved by switching to polling mode when the system is under high load. This is also implemented by some network devices. Polling however has it's own difficulties, for example choosing the right frequency.

2.3 Horizontal Scaling and Placement Awareness

> "Enterprise datacenter storage is frequently consolidated into a single server with many disks, colloquially called a JBOD (Just a Bunch Of Disks). JBODs typically contain 70 – 80 spinning disks and are controlled by a single controller or head, and provide a high-capacity, low-performance storage server to the rest of the datacenter." [3]

In speed comparison a single SCM can outperform a whole JBOD, but the capacity will not be satisfiable enough. With a JBOD of SCMs this problem will be solved, though others will arise. For example the SCM JBOD requires 350G - 400G of bandwidth and this will draw 3.000 W. This is obviously impractical. Another big problem is to guarantee high performance across clustered machines. Synchronized file system metadata and a lot of communication are needed to satisfy this.

2.4 Workload-aware Storage Tiering

Another important point in using SCMs is to maintain a high percentage of hot data in the storage. It would be extremely inefficient to store data which will be seldom accessed into an high speed storage device. This concept is similar to caching, because in caches the hot data is clustered too. The best solution would be a hybrid system with SCMs and normal hard drives. Therefore, fast non-volatile storage could be seen as a kind of cache for slower disks. Unfortunately, this idea is not perfect either and causes some complications with granularity access and different storage tiers.

3 Conclusion

The advantages of fast non-volatile storage is kind of clear. However, integrating SCMs in current systems is a really tough task. There are many solutions which

do not fully satisfy the SCM or only fit in certain situations. Though, we are just at the beginning of a completely new era of memory and can expect much more in the near future.

References

[1] Chiao-Ying Huang. Storage class memory.

[2] C. H. Lam. Storage class memory. In *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE International Conference on*, pages 1080–1083, Nov 2010.

[3] Mihir Nanavati, Malte Schwarzkopf, Jake Wires, and Andrew Warfield. Non-volatile storage. *Commun. ACM*, 59(1):56–63, December 2015.

[4] Athanasios E Papathanasiou and Michael L Scott. Aggressive prefetching: An idea whose time has come. In *HotOS*, 2005.