

CSE 551
Design Exercise #2
File Systems on Phase Change Memory

First draft due: noon, Thursday, April 30, 2009

Final draft: 4:30pm, Thursday, May 7, 2009

The widespread deployment of lightweight portable devices has led to increasing interest in hardware technologies for chip-based persistent storage. Non-volatile, or flash, RAM is an early example of this technology, but flash is both relatively slow (compared to RAM) and is severely limited in terms of how many times it can be written during its lifetime. While many hand-held devices and even a few high-end portables (such as the MacBook Air) use flash RAM, most portables and workstations continue to use mechanical disk, with its cheap per byte storage, high transfer rates, and exceedingly long seek latencies. The next generation non-volatile chip technology, called “phase change memory” or PCM, attempts to address these limits with flash RAM. PCM offers read and write latency within an order of magnitude of normal RAM, and nearly unlimited storage cycles. Like flash, PCM is word addressible – there is no need to read or write an entire sector as with mechanical disks. Perhaps most importantly, PCM offers much lower energy costs relative to mechanical disks. Spinning disks draw significant power, while quiescent disks take seconds to power up, limiting responsiveness. At present, PCM is prohibitively expensive per unit of storage, but costs may decrease with volume, making them a contender for completely replacing disk storage over time, especially for portables (now the largest segment of the computer market).

The assignment is to design a file system specifically for PCM. The UNIX file system and its derivatives were designed for an environment where the unit of persistent storage is relatively large and relatively slow. How would PCM change file system design? You can target one or both of the following scenarios for PCM deployment. One deployment scenario has PCM as a level in the storage hierarchy in front of disk (e.g., systems needing a lot of storage will continue to have disk for capacity, but most reads and writes will go to PCM). The other has PCM as the only storage on the system, completely replacing disk. Please consider: atomicity, durability, file system metadata structures, and the API – will a file system read/write interface still make sense with PCM?

As before, we are most interested in your general approach – e.g., try to keep the writeup to 1-2 pages.