CSE 551
Design Exercise #1
File Systems on Phase Change Memory

First draft due: noon, Thursday, November 5, 2009
Final draft: 4:30pm, Thursday, November 12, 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 bridge this gap.

PCM's relevant characteristics:

- read and write latency/bandwidth within a factor of 2-5 of normal RAM
- word addressible – there is no need to read or write an entire sector as with mechanical disks
- density comparable to RAM (can replace memory slots with PCM slots)
- much lower energy costs than either RAM or mechanical disks. RAM requires power to refresh memory, and add seconds to restart times if RAM is checkpointed to disk to save power. Spinning disks draw significant power, while quiescent disks use little power but take seconds to power up, limiting responsiveness. (Note that even for servers, power usage is a large factor in total cost over the lifetime of the device.)
- high cost per unit of storage, but costs are likely to decrease over time and with volume.
- writes stop working after approx 10^12 lifetime writes to a particular memory location (longer than flash, shorter than disk)

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 should target one of the following scenarios for PCM deployment. Either the PCM is a level in the storage hierarchy between RAM and disk (e.g., systems needing a lot of storage will continue to have disk for capacity, but most reads and writes will be able to be satisfied out of PCM), or the PCM completely replaces disk and/or RAM. (Also be clear if your design is for portables or server environments.) Please consider, in addition to cost, performance, and energy use: atomicity, durability, file system metadata structures, and the API – will a file system read/write interface still make sense with PCM?

In addition, we would like you to describe how you would quantitatively evaluate your design. What principal experiment would you run, and what would you expect it to show?

We are interested in a sketch of your general approach, rather than details -- e.g., try to keep the writeup to 1-2 pages for the first draft, 2-3 pages for the final draft.