Interview with Darrell Long

RIK FARROW

Dr. Darrell D. E. Long is Professor of Computer Science at the University of California, Santa Cruz. He holds the Kumar Malavalli Endowed Chair of Storage Systems Research and is Director of the Storage Systems Research Center. His broad research interests include many areas of mathematics and science, and in the area of computer science include data storage systems, operating systems, distributed computing, reliability and fault tolerance, and computer security. He is currently Editor-in-Chief of *ACM Transactions on Storage*.
darrell@soe.ucsc.edu

Rik Farrow is the editor of *;login:*.
rik@usenix.org

The Intel/Micron announcement of XPoint 3D in July 2015 really got my attention [1]: finally, a vendor will start shipping a form of non-volatile memory (NVM) that's not NAND flash. XPoint 3D promises to be byte addressable, faster, more durable, and require lower power than any form of flash today. The downsides (there are always downsides) will be that XPoint 3D will be more expensive and have less storage capacity when it appears in early 2016.

Having byte-addressable NVM will have impacts on the way computers are designed and operating systems and software are written. If this technology proves to be everything that Intel and Micron are promising, it might change everything about the systems we are familiar with. At the very least, XPoint 3D would become a new tier in the storage hierarchy.

I asked around, trying to find someone I knew in our community who could address this topic from a file system and storage perspective. The timing was terrible, as all of the people I asked (who responded) were busy preparing FAST '16 papers for submission, and with two deadlines at about the same time, you can guess which one is the more important.

Darrell Long, a professor at UCSC, took up my challenge, even though he too was busy on an overseas trip, as well as supervising papers to be submitted to FAST '16. Long has experience in both storage systems and operating systems, and seemed like the right person to talk to about this development.

Rik: I recently heard about a new type of NVM developed by Intel and Micron and in production. I've been hearing talks about how a technology like this could result in large changes in the designs of systems and operating systems for many years.

Darrell: I don't know a lot about the details of the technology, and was waiting to get home from travel to sit down with Intel for a technical briefing on the details, but I agree with you 100% that this may in fact be a game-changer. Ethan and I wrote a paper on this about 14 years ago, when there was hope for MRAM (before physics got in the way) [2].

One of the key points is that unlike flash, this *will* be on the memory bus. We will have persisted memory that is lower power, denser, and unfortunately slower than DRAM. But it will be byte addressable. That means all the tricks we have developed over the years for packing data structures into blocks go away—at least at that level.

My belief is that files do not go away, they are simply too useful. I think it would be a mistake to tie an object, say a photograph, to an application (as Apple loves to do with the iPad). Objects may become first-class entities, and you can then think of applications as operators that perform transformations on them. There will be a lot of them, so we still need naming and protection, and we still need long-term storage—so I do not see spinning rust disappearing, at least not for quite a while, but it may move back into "archive" where we keep our named objects. And as before with disk, we will still need to worry about mapping memory addresses to some (in this case) higher level representation; unlike Word in the old days, you can't just dump persistent memory to disk and load the image.

Rik: MRAM never happened, and XPoint 3D will be slower than DRAM, initially by a factor of 100,000, although the claim is that this will come down to 1000 times slower than DRAM.

Darrell: Physics got in the way of MRAM, and phase change memories were not yet a thing when we wrote the paper [2]. So MRAM will never happen, except for low-density stuff that needs to be radiation hard (but phase change is also radiation hard).

I think the key issues for this new technology are byte-addressability and its speed relative to the others. For high performance computing, power consumption will also be huge. Most people do not realize it, but the vast majority of energy comes from moving the bits, not the computation. If you look at a processor die, it's all cache and bit movement. The ALU and FPU are tiny parts of it these days.

Having byte-addressable persistent memory on the memory bus is, I think, a game changer. I am not sure how it will all shake out. But the usual ideas of never having to fully reboot (but we need to retain that ability due to the crappy software that gets written) will certainly come up.

I think the key things that we need to think about are how we manage our data: how do we name it, find it, and protect it? The file model works pretty well, and we may not want to just throw it away. But we can lose a lot of its strictures.

Rik: I agree, moving bits is expensive, and have been looking at dies (well, masks for dies) since the early '80s. Now, it's mostly cache and the parts that determine whether the right line is present or not.

As for crappy software, even geniuses can't write perfect software, and most people who write software aren't geniuses. Someday, perhaps software systems will write software, but even then there might be memory leaks, accidental corruption of data structures, and so on.

And will all of the strictures on file system design go away? I think file systems will still use locks, write metadata before data is written, and so on. NVM file systems will need to be different, or should be different, for handling media that is byte rather than block addressable. That might be the interesting bit, given examples like NTFS, where MS systems programmers decided to have irregularly sized data structures, as they did in their in-memory logging system, and had terrible trouble with reliability and in the performance and reliability of their logging system.

Darrell: I haven't completely thought it through yet, but I think that the role of file systems will change. And I think that consequently they will get simpler. Consider the object abstraction we have been pushing for about 20 years (Swift [3] was 1991), and it is finally getting traction through Ceph and Seagate's Kinetics. A lot of the low-level stuff you can just push off onto the device, and let the higher-level file system worry about naming, load balancing and distribution, and protection (though some of that must be at the object level too).

Now look at the persistent memory. If the density is high enough, we really don't need the low-level parts of the file system on, say, your laptop. We already have flash, although that pretty much pretends to be a fast disk. But we will still need the naming and protection; when we want to back up our system, we will still want to use something like files. We will be taking byte-addressable memory and mapping it to block storage, be it flash, disk, tape, or the long-promised holostores.

So the locking and so forth will still be there, but it will change. It will be more like shared memory, and in the beginning before programming models really change I would expect a lot of mmap() kinds of things to happen. The back-end file systems may get a lot simpler, since they will probably not need to support range locking or update in most cases. Remember Tanenbaum's "Bullet" file system? [4] They could very well end up like that, with immutable files that get written in one fell swoop.

References

[1] Intel Micron XPoint memory announcement: http://newsroom.intel.com/community/intel_newsroom/blog/2015/07/28/intel-and-micron-produce-breakthrough-memory-technology; analysis by Anandtech: http://www.anandtech.com/show/9470/intel-and-micron-announce-3d-xpoint-nonvolatile-memory-technology-1000x-higher-performance-endurance-than-nand.

[2] Ethan L. Miller, Scott A. Brandt, and Darrell D. E. Long, "HeRMES: High-Performance Reliable MRAM-Enabled Storage," *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (HotOS-VIII),* Elmau, Germany: IEEE, May 2001, pp. 83–87: ftp://ftp.soe.ucsc.edu/pub/darrell/HotOS-Miller-01.pdf.

[3] Luis-Felipe Cabrera and Darrell D. E. Long, "Swift: Using Distributed Disk Striping to Provide High I/O Data Rates," *Computing Systems,* vol. 4, no. 4 (1991), pp. 405–436: https://www.usenix.org/publications/compsystems/1991/fall_cabrera.pdf.

[4] Robbert van Renesse, Andrew S. Tanenbaum, Annita Wilschut, "The Design of a High-Performance File Server," *9th International Conference on Distributed Computing Systems,* 1989, pp. 22–27: https://static.aminer.org/pdf/PDF/000/297/552/the_design_of_a_high_performance_file_server.pdf.