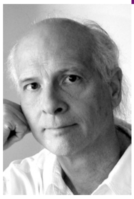
RIK FARROW

musings



Rik is the Editor of *;login:*.

rik@usenix.org

AS I HIKED DOWN PANORAMIC HILL, shortly after the HotPar workshop had completed in Berkeley, I marveled at the mounds of dirt pushed up by gophers. Were these the same species of gophers that made similar mounds in my yard? At home, the gophers come and go, apparently limited by effective predation. There is a natural balance between prey and predators, just as there is a balance that occurs in the acceptance and use of related types of technologies.

HotPar had been preceded by FAST, and you will find the FAST '09 summaries in this issue of *;login:*. Both HotPar and FAST sessions explored the edges of research in these two fields. And both left me with the impression that there are real trade-offs in computer technology, just like the natural balancing acts I can see when I am hiking.

Storage

I arrived too late to catch any of the morning FAST tutorials, but I did get to listen in on the Storage Class Memory (SCM) Technology tutorial [1] given by a group of IBM Almaden researchers. Flash-based Solid State Disks (SSDs) immediately come to mind when considering new types of storage. But these IBM researchers point out that flash does not rise to the level of what they consider SCM.

They expect that SCM will become a routine part of server-class systems in the future—that is, by 2013. SCM will sit between DRAM and disk, but mainly as a DRAM replacement, instead of replacing or supplanting the familiar hard drives of today. I considered this strange, as SSDs have replaced disks in laptops already, and Sun has products in the works for placing SSDs in storage systems. Besides, the current disadvantages found in flash seem to preclude flash from ever replacing random access memory. But the IBM guys are experts in their field, and this was their prediction.

Flash really shines at random read operations. But 7200 rpm consumer hard drives have the best cost/performance ratio for every other class of storage operations, something that surprised me (but I trust the Carnegie Mellon researchers who produced this result [2]). Flash has a big problem with writes. Flash, like any hard drive, always writes data in blocks. But flash works differently from disk, in that a block has to be erased before it can be written. Typical flash blocks are 128KB, so writing a single byte of data means finding a previously erased block, reading the current data block into an on-device buffer, modifying the block, and writing it out. On top of this, flash technology is limited to 100,000 writes per block, implying that long-term endurance of a flash device could be measured in seconds.

Flash devices handle endurance with the Flash Translation Layer (FTL), which uses write-leveling to distribute erasures and writes across the entire media. Some flash uses, particularly those typical of desktops and laptops, are predominantly random read, so flash can function well here. The higher cost of SSD implies that flash will appear first in higher-cost devices, and today you only see SSD being sold with laptops and netbooks—a natural fit.

During an SSD BoF at FAST, Milo Polte of Carnegie Mellon pointed out that we still don't know where SSDs best fit into system architecture. SSDs are currently used as disk replacements, but he and others present at the BoF questioned if this is the right place for them. SSD vendors make flash appear to be block-oriented, but future uses may be more like what the IBM researchers envision, with SCM acting as a main memory replacement and not as block-oriented storage devices.

Like hard drive manufacturers, SSD vendors are hiding the internal details and functioning of their products [3]. Intel's pricy X25E 32GB SSD, according to Polte, starts off with great write performance, but this plummets because the device hides the overhead of block erasure during its initial operation. Once erased blocks become scarce, the thread controlling block erasure, a slow process (2ms), cuts heavily into write performance.

We are watching the evolution of a new class of hardware, the SCM. Just like the gophers and their predators, this new class will eventually find its way into its proper niche. Unlike nature, human researchers and designers have control over the design of flash and SCM, and that will have a lot to do with where SCM will fit into the architecture of future systems.

The Pack Against the Three-Headed Hound

During the late 1980s, I had the privilege of getting to review new workstation hardware for *UnixWorld* magazine. I didn't even own a computer, as I always had at least one brand-new UNIX-based workstation at home. I studied the systems architecture, as well as the user manuals for CPUs (MIPS, SPARC, PowerPC, Alpha), trying to understand why certain workstations did better at different benchmarks. Floating-point performance depended largely on processor support, but general performance relied on good bus design.

My workstation focus left me largely unaware of the revolution in multiprocessing that was happening at the same time. There were good reasons for this—for example, I couldn't have my own Connection Machine [4] for review. Not only were delivery, setup, power, and cooling serious issues, but I really had no idea how to review such massive and specialized systems.

Other people within the design industry were paying close attention to various Symmetric MultiProcessors (SMPs), because they saw these as potential game changers in the high-end server market. Gregory Pfister of IBM in Austin had collected a lot of research which eventually became a book comparing various forms of SMP to clusters [5]. Ric Wheeler of Red Hat recommended that I read this book when I mentioned my interest in earlier SMP technologies.

I actually had owned this book, but gave it away without reading it. The cover depicts a three-headed dog fighting off a pack of dogs. I recall puzzling over this weird cover before packing the book up to send to a university library, something I did with many of the books publishers sent me out of the blue. Now I needed to buy a copy of the book I had given away many years ago.

Pfister is an entertaining writer who is also a master of his topic. The cover that puzzled me symbolizes SMPs as multi-headed beasts and a cluster as a pack of dogs. He explains that while an SMP has multiple "heads," the processors, it has to share the rest of the "body." The pack of dogs represents the complete computer systems in a cluster all working together to attack a problem. The SMP not only must share system resources, such as memory and I/O, but also has less resiliency, as there is only one system image running. The cluster can lose one or more of its members and continue running, as each member is a compete system.

The big trade-off between SMP designs and clusters has to do with memory latency. SMPs share memory, so they can communicate quickly when running parallel programs. Cluster members each have their own memory and rely on message-passing to synchronize while running similar programs. Message-passing latency, usually over commodity networks, is many times slower than interprocess communication in SMP systems.

NUMA, Non-Uniform Memory Access systems, or Cache Coherent NUMA as Pfister calls them, sit somewhere in between when it comes to communications latency. These systems have high-speed interconnects, like the AMD Barcelona, so shared memory can be close (controlled by the on-chip memory manager) or far (accessed via another CPU's memory manager). Cache coherency means that changes to values in one processor's memory will be visible to other connected processors, but only after a significant delay.

You may be wondering what SMPs of the past have to do with the systems of today and the near future. The connection is obvious once you consider that multicore chips are like SMPs; as the number of cores grows larger over time, CPU designers are facing the same types of issues faced by the much earlier SMP system designers, with access to memory being the most significant one.

Cluster systems have become very popular when it comes to applications that can have lots of data and can be partitioned easily. Google's search application should come to mind, as it involves storing immense amounts of data using a cluster file system (GFS) and manipulating that data with MapReduce operations. MapReduce allows data to be partitioned, with each map or reduce operating on data that is local. Hadoop, an open source project sponsored by Yahoo!, works in a similar fashion and has been adopted by companies working outside of the Internet search business, but with lots of data and tasks that are easy to partition.

Clusters certainly have their place today and in the future of large systems. Multicore systems will dominate both the server and the desktop, including each head/node in any cluster. This is why I wanted to read Pfister's book, which despite being over 10 years old is still very relevant (and still in print, which also says a lot).

The Line-up

Nothing like thinking about multiple racks of high-powered servers used by clusters to get one thinking about energy. Alva Couch, professor at Tufts University and a USENIX Board member deeply interested in Green IT, writes about the Greening of IT. I really like how Alva describes the two types of Green that IT cares about and how this impacts buying behavior.

Next up, Hasan et al. have written a nice article, based on their FAST '09 paper, about provenance. Margo Seltzer drew my attention to their paper as a potential *;login:* article, and I agreed it would work well. Hasan explains what provenance is and why it is important and becoming even more important, before describing their own work on turning provenance on shared data into something that can be trusted.

Andrew Leung, working with NetApp engineers and Ethan Miller, writes about Spyglass. Based on a related FAST '09 paper, Leung points out that metadata searches for all the files owned by a particular user greater than 100KB and modified during the past week, for example, can take too long to accomplish when storage systems become truly large. The UNIX `find` command doesn't scale well when used on petabyte systems (unless you plan on reading a nice book like Pfister's while you wait for a response). Leung et al. came up with a solution that uses a separate store of metadata to speed up searches by a couple of orders of magnitude.

Weihang Jiang et al. examined the role of system logs when troubleshooting storage system problems. Like past FAST research sponsored by NetApp, this research also relies on the vast storehouse of data collected by NetApp as customers use their storage devices. In this article, Jiang explains the relationship between various classes of storage failures, from the failure of a disk to that of software, and how logs can help in analyzing these failures. I've spent a fair amount of time looking at logs in my life, as I suspect is true of most USENIX members. The results of Jiang's research will, I expect, match your own experience.

Rudi van Drunen continues his series on hardware by explaining signals. Unlike power and voltage, signals need to deliver data reliability, and Rudi explains how factors such as distance and the design of cables affect data delivery.

David Blank-Edelman continues on his theme of Web-page scraping by taking on the challenge of extracting data from a messy page. Peter Galvin describes advances in installing the LAMP/SAMP stacks when using Solaris and OpenSolaris. Dave Josephsen tells us about what happened when the company he works for suffered from the TV version of slashdotting, an interesting tale. Robert Ferrell then regales us with his thoughts on social networking.

Elizabeth Zwicky and Sam Stover have written book reviews for this issue. Jason Dusek has vowed to return in the next issue with more reviews about programming books.

We finish up with FAST '09 and TaPP '09 conference reports.

Wrap-up

I still am astounded by the diversity I see in nature. Once I learned how to pay attention to my surroundings, I started to notice how the vegetation changed between the different faces of a hill, related to both sun and water, and how different critters exist in different environments.

As I walked down Panoramic Hill, I not only noticed the gopher holes but wondered if there were any pesky pack rats in the Bay Area. Pack rats build messy nests of gathered materials, and will even do so on car engines using your wiring harness as part of their nest. You really don't want pack rats in your area, believe me. Their range appears limited to desert areas, including most of Arizona, and some of the southern California deserts. Pack rats have evolved to fit a particular niche, and our technology just happened to get involved in very recent times.

I expect that new technologies such as SSDs, SCM, and multicore processors will also fit into their own niches. Computer scientists and technology companies invent devices, but these devices will fit naturally into whatever environment that matches them the best.

REFERENCES

[1] Wilcke, "Flash and Storage Class Memories" (similar to a portion of the FAST '09 tutorial): institute.lanl.gov/hec-fsio/workshops/2008/presentations/day3/Wilcke-PanelTalkFlashSCM_fD.pdf.

[2] Simsa, Polte, and Gibson, "Comparing Performance of Solid State Devices and Mechanical Disks," Parallel Data Laboratory, Carnegie Mellon University, 2008: http://www.pdsi-scidac.org/events/PDSW08/resources/slides/simsa_PDSW.pdf.

[3] Farrow, "Musings," June 2007: http://www.usenix.org/publications/login/2007-06/openpdfs/musings.pdf.

[4] Thinking Machines: http://en.wikipedia.org/wiki/Thinking_Machines.

[5] Pfister, *In Search of Clusters,* 2nd ed. (Prentice Hall, 1998), ISBN 0-13-899709-8.

PACKRATS OFTEN CONSTRUCT NESTS OF STICKS AND DEBRIS AROUND LARGE PRICKLY PEAR CACTUSES. THIS NEST IS SIX FEET ACROSS.

PACKRATS WILL USE HUMAN-PROVIDED STRUCTURES, SUCH AS WOODPILES OR CARS, WHEN BUILDING NESTS. WHEN USING A CAR, SUCH AS MY SUBURU, SHOWN HERE, ONLY THE INTERIOR LAYER, SHREDDED JUNIPER BARK FOR BEDDING, IS REQUIRED.