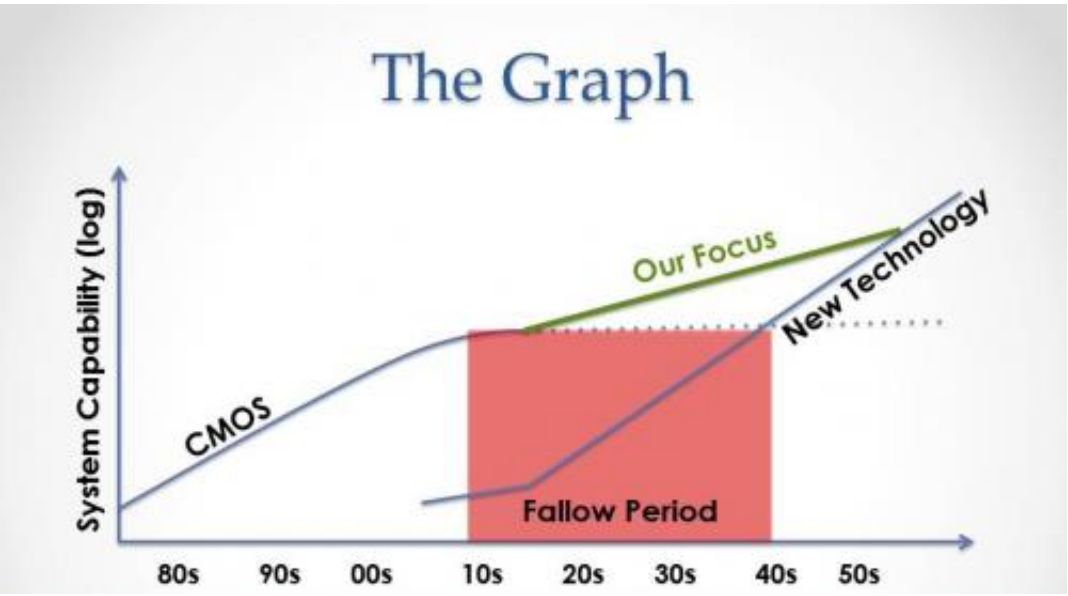
Researcher finds hidden efficiencies in computer architecture

April 18 2014, by Aaron Dubrow



Key question: How to advance computer performance without significant technological progress? Hill and others are working to harvest new gains during the "fallow" period of the near future. Credit: Advancing Computer Systems without Technology Progress, ISAT Outbrief, Mark D. Hill and Christos Kozyrakis, DARPA/ISAT Workshop, March 26-27, 2012

The computer is one of the most complex machines ever devised and most of us only ever interact with its simplest features. For each keystroke and web-click, thousands of instructions must be communicated in diverse machine languages and millions of calculations computed.

Mark Hill knows more about the inner workings of computer hardware than most. As Amdahl Professor of Computer Science at the University of Wisconsin, he studies the way computers transform 0s and 1s into social networks or eBay purchases, following the chain reaction from personal computer to processor to network hub to cloud and back again.

The layered intricacy of computers is intentionally hidden from those who use—and even those who design, build and program—computers. Machine languages, compilers and network protocols handle much of the messy interactions between various levels within and among computers.

"Our computers are very complicated and it's our job to hide most of this complexity most of the time because if you had to face it all of the time, then you couldn't get done what you want to get done, whether it was solving a problem or providing entertainment," Hill said.

During the last four decades of the 20th century, as computers grew faster and faster, it was advantageous to keep this complexity hidden. However, in the past decade, the linear speed-up in processing power that we'd grown used to (often referred to as "Moore's law") has started to level off. It is no longer possible to double computer processing power every two years just by making transistors smaller and packing more of them on a chip.

In response, researchers like Hill and his peers in industry are reexamining the hidden layers of computing architecture and the interfaces between them in order to wring out more processing power for the same cost.

Ready, set... compute

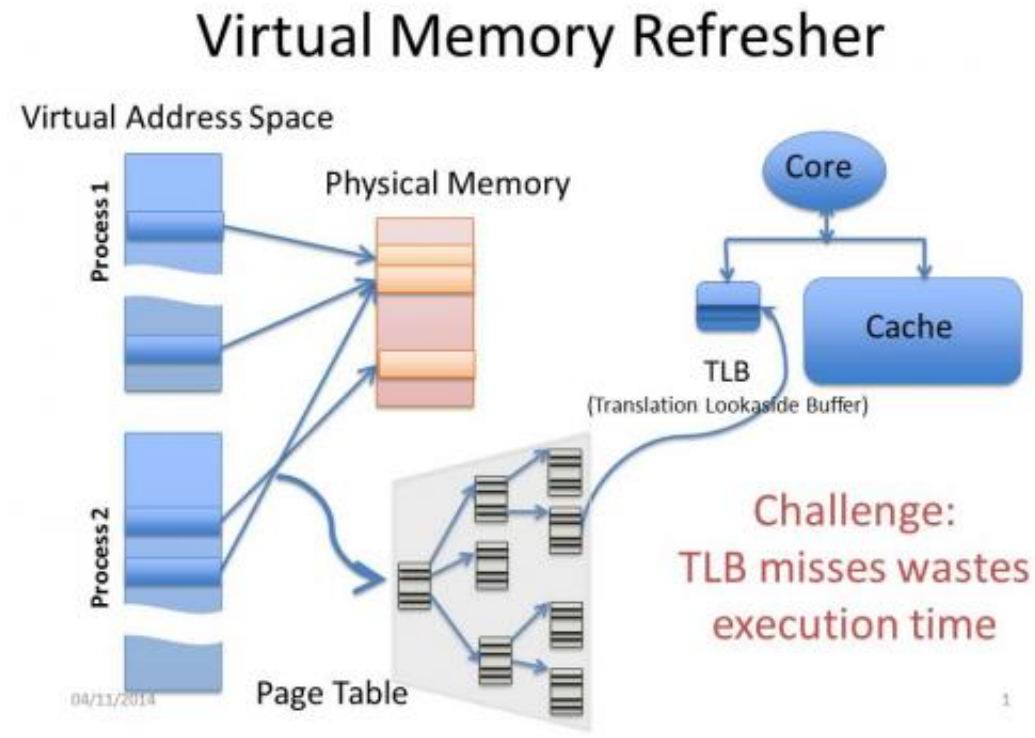
One of the main ways that Hill and others do this is by analyzing the performance of computer tasks. Like a coach with a stopwatch, Hill

times how long it takes an ordinary processor to, say, analyze a query from Facebook or perform a web search. He's not only interested in the overall speed of the action, but how long each step in the process takes.

Through careful analysis, Hill uncovers inefficiencies, sometimes major ones, in the workflows by which computers operate. Recently, he investigated inefficiencies in the way that computers implement virtual memory and determined that these operations can waste up to 50 percent of a computer's execution cycles. (Virtual memory is a memory management technique that maps memory addresses used by a program, called virtual addresses, to physical addresses in computer memory, in part, so that every program can seem to run as if is alone on a computer.)

The inefficiencies he found were due to the way computers had evolved over time. Memory had grown a million times bigger since the 1980s, but the way it was used had barely changed at all. A legacy method called paging, that was created when memory was far smaller, was preventing processors from achieving their peak potential.

Hill designed a solution that uses paging selectively, adopting a simpler address translation method for key parts of important applications. This reduced the problem, bringing cache misses down to less than 1 percent. In the age of the nanosecond, fixing such inefficiencies pays dividends. For instance, with such a fix in place, Facebook could buy far fewer computers to do the same workload, saving millions.



Software generates virtual addresses as it accesses memory. Each process has its own virtual address space. This virtual address gets mapped to a physical address at the granulairity of the page. The mapping information is stored in a hierarchal page table. Since each memory access needs a translation, processors used a hardware cache called the translation look aside buffer (TLB). Hits to the TLB are fast but a miss causes a delay of several cycles. The goal is thus to reduce the TLB misses. While such a TLB design remained unchanged for several decades memory usage has changed significantly. Hill designed a solution that uses paging selectively, adopting a simpler address translation method for key parts of important applications. This reduced the problem, bringing cache misses down to less than 1 percent. In the age of the nanosecond, fixing such inefficiencies pays dividends. For instance, with such a fix in place, Facebook could buy far fewer computers to do the same workload. Credit: Mark D. Hill

"A small change to the operating system and hardware can bring big benefits," he said.

Hill and his colleagues reported the results of their research in the International Symposium on Computer Architecture in June 2013.

Computer companies like Google and Intel are among the richest in the world, with billions in their coffers. So why, one might ask, should university researchers, supported by the National Science Foundation (NSF), have to solve problems with existing hardware?

"Companies can't do this kind of research by themselves, especially the cross-cutting work that goes across many corporations," said Hill. "For those working in the field, if you can cross layers and optimize, I think there's a lot of opportunity to make computer systems better. This creates value in the U.S. for both the economy and all of us who use computers."

"The National Science Foundation is committed to supporting research that makes today's computers more productive in terms of performance, energy-efficiency and helping solve problems arising from the entire spectrum of application domains, while also studying the technologies that will form the basis for tomorrow's computers," said Hong Jiang, a program director in the Computer Information Science and Engineering directorate at NSF.

"In the process of expanding the limits of computation, it's extremely important to find both near-term and long-term solutions to improve performance, power efficiency and resiliency. Professor Mark Hill's pioneering research in computer memory systems is an excellent example of such efforts."

The "divide and conquer" approach to computer architecture design, which kept the various computing layers separate, helped accelerate the industry, while minimizing errors and confusion in an era when faster

speeds seemed inevitable. But Hill believes it may be time to break through the layers and create a more integrated framework for computation.

"In the last decade, hardware improvements have slowed tremendously and it remains to be seen what's going to happen," Hill said. "I think we're going to wring out a lot of inefficiencies and still get gains. They're not going to be like the large ones that you've seen before, but I hope that they're sufficient that we can still enable new creations, which is really what this is about."

Most recently, Hill has been exploring how graphic processing units (GPUs), which have become common in personal and cloud computing, can process big memory tasks more efficiently.

Writing for the proceedings of the International Symposium on High-Performance Computer Architecture, Hill, along with Jason Power and David Wood (also from the University of Wisconsin), showed that it is possible to design virtual memory protocols that are easier to program without slowing down overall performance significantly. This opens the door to the use of GPU-accelerated systems that can compute faster than those with only traditional computer processing units.

Accelerating during a slow-down

Improvements to virtual memory and GPU performance are a few examples of places where cross-layer thinking has improved computer hardware performance, but they are also emblematic of a wholesale transformation in the way researchers are thinking about computer architecture in the early 21st century.

Hill led the creation of a white paper, authored by dozens of top U.S. computer scientists, that outlined some of the paradigm-shifts facing

computing.

"The 21st century is going to be different from the 20th century in several ways," Hill explained. "In the 20th century, we focused on a generic computer. That's not appropriate anymore. You definitely have to consider where that computer sits. Is it in a piece of smart dust? Is it in your cellphone, or in your laptop or in the cloud? There are different constraints."

Among the other key findings of the report: a shift in focus from the single computer to the network or datacenter; the growing importance of communications in today's workflows, especially relating to Big Data; the growth of energy consumption as a first-order concern in chip and computer design; and the emergence of new, unpredictable technologies that could prove disruptive.

These disruptive technologies are still decades away, however. In the meantime, it's up to computer scientists to rethink what can be done to optimize existing hardware and software. For Hill, this effort is akin to detective work, where the speed of a process serves as a clue to what's happening underneath the cover of a laptop.

"It's all about problem solving," Hill said. "People focus on the end of it, which is like finishing he puzzle, but really it's the creative part of defining what the puzzle is. Then it's the satisfaction that you have created something new, something that has never existed before. It may be a small thing that's not well known to everybody, but you know it's new and I just find great satisfaction in that."

More information: The white paper titled "21st Century Computer Architecture" is available online: www.cra.org/ccc/files/docs/ini … ecturewhitepaper.pdf

Provided by National Science Foundation