'Not so fast, supercomputers,' say software programmers

May 22 2007

The fastest of the fastest computers - supercomputers used at national research centers, research universities and major corporations - will soon gain even more performance by taking advantage of multicore computing.

Despite the promise of almost unimagined computing power, however, even computing experts wonder whether this time the hardware developers have raced too far ahead of many programmers' ability to create software.

Faisal Saied, senior research scientist for Information Technology at Purdue, says that parallel computing has been an esoteric skill limited to people involved with high-performance supercomputing. That is changing now that desktop computers and even laptops are multicore.

"High-performance computing experts have learned to deal with this, but they are a fraction of the programmers," Saied says. "In the future you won't be able to get a computer that's not multicore, and as multicore chips become ubiquitous, all programmers will have to learn new tricks."

Even in high-performance computing there are areas that aren't yet ready for the new multicore machines.

"In industry, much of their high-performance code is not parallel," Saied says. "These corporations have a lot of time and money invested in their software, and they are rightly worried about having to re-engineer that code base."

Multicore computers have more than one processing unit, or CPU, on a computer chip, each, in essence, a separate PC. In the next few years new high-performance computers will have dozens or hundreds of PCs on a chip, offering vast improvements in computing performance over current top machines.

Multicore computers are required if computers are going to continue to increase in computing performance as they have over previous decades. This increased performance is needed for a variety of high-tech tasks, such as climate modeling, military weapons design, drug discovery and improving manufacturing.

But multicore computers require parallel computer programs because each PC, or core, must get its own set of instructions. Meanwhile, much of the currently available software is not written to take advantage of multicore computing.

For all of the amazing things that computers do, they do it all one thing at a time. The instructions are delivered in single file, as if passing through a single door. Parallel processing opens more doors, but also creates challenges because of the multiple instruction threads required.

"Imagine you had four golf balls and needed to hit four targets. If you had four people each throwing a ball at the same time, they could do it faster than one person alone. That's the advantage of multicore computing. Multiple PCs, all on the same chip, are working on multiple tasks. The difficulty comes in breaking the task into multiple components," Saied says.

Steve Kirsch, an engineering fellow for Raytheon Systems Co., says that multicore computing presents both the dream of infinite computing power and the nightmare of programming.

"The real lesson here is that the hardware and software industries have to pay attention to each other," Kirsch says. "Their futures are tied together in a way that they haven't been in recent memory, and that will change the way both businesses will operate."

Gordon Moore, retired chairman and CEO of Intel Corp., famously observed that the number of components on an integrated chip would double every 24 months (often stated as every 18 months), and Moore's Law has served as both a prescient prediction and an engineering goal for the information technology industry.

But about five years ago, researchers began seeing a discrepancy between the predicted performance from circuits and the actual computing capability in high-performance computing. Although the number of transistors on the circuits continued to increase, as predicted by Moore's Law, actual performance remained about the same because of power and heat issues. This has become known as "Moore's Gap."

The problem with current chips is that the transistors leak power even when they are doing nothing, and with sequential processing there are a lot of processors waiting their turn, says Tilak Agerwala, vice president of systems at IBM Research.

"Currently, transistor performance is limited by power constraints causing microprocessor clock speeds to saturate and high-performance microprocessor cores to dissipate more power than simpler alternatives," Agerwala says. "As the performance of a single thread of computation flattens out, multicore processing will become critical to system performance growth."

Computation is at a point where multithread programming is the only way to accelerate innovation and discovery, Agerwala says.

"We will deliver the levels of computing capability required to advance the fields of science and engineering," Agerwala says. "Future supercomputers will effectively utilize many cores per chip and perhaps a million cores per system on multithreaded, parallelized applications."

Agerwala notes that IBM's Blue Gene/L, the world's most powerful supercomputer, already exploits 131,072 processor cores all working in parallel, using two cores per chip.

Chip makers Intel, IBM, AMD and Sun have all announced that they will soon begin producing multicore chips. In February, Intel released research details about a chip with 80 cores, a fingernail sized chip that has the same processing power that in 1996 required a supercomputer with a 2,000-square-foot footprint and using 1,000 times the electrical power.

Kirsch says despite the promise of powerful new supercomputers, multicore computing presents a problem for companies and researchers who depend on previously written software that has been steadily improving and evolving over the past few decades. "Our legacy software is a real concern to us," he says.

Kirsch said that parallel programming for multicore computers may even require new computer languages.

"Today we program in sequential languages," he says. "Do we need to express our algorithms at a higher level of abstraction? Research into these areas is critical to our success."

Researchers at Purdue, working closely with industrial collaborators, are developing new programming models and tools that simplify the task of writing programs for a multicore platform, says Susanne Hambrusch, professor and head of Purdue's Computer Science Department.

"Our programming languages researchers are exploring new programming paradigms and models," Hambrusch says. "Our course on multicore architectures is also preparing students for future software development positions. Purdue is clearly playing a defining role in this critical technology."

Multicore computers are beginning to appear for consumers, and computer scientists say this commodity approach to parallel computing will benefit consumers as well as users of high-performance computing.

Suresh Jagannathan, an associate professor of computer science at Purdue, has a positive outlook on the future of parallel programming in computing.

"There's a thin line between pessimism and opportunity," he says. "This is a definite opportunity. There is notable work here at Purdue to develop new programming languages, abstractions and implementations to harness parallelism and bring it into the mainstream."

Jagannathan says the transition for desktop computers will take a different path than that of high-performance computing.

"HPC systems are on the bleeding edge of technology," he says. "But it's not the case that we've hit a brick wall. There are approaches available to us. In the long run, we will harness multicore technology, and more programs will take advantage of it. When multicore is ubiquitous, we expect the entire software stack - from applications all the way down to operating system kernels - to take advantage of the parallelism afforded by these architectures."

Saied says that although the difficulty of writing parallel software is currently an issue in high-performance computing, that issue will reach desktop computing soon enough.

"In five or six years, laptop computers will have the same capabilities, and face the same obstacles, as today's supercomputers," Saied says. "This challenge will face people who program for desktop computers, too. People who think they have nothing to do with supercomputers and parallel processing will find out that they need these skills, too."

Source: Purdue University

Citation: 'Not so fast, supercomputers,' say software programmers (2007, May 22) retrieved 10 April 2024 from https://phys.org/news/2007-05-fast-supercomputers-software-programmers.html