

Multi-core! What Is It Good For?

By Andy Patrizio
June 22, 2007

At the turn of this century, semiconductor makers were facing a conundrum. Traditionally, they would simply bump up the clock speed to get more performance out of their processors.

After doing this so effectively for two decades, the gimmick was running out of gas. More clock speed meant more heat, and processors have gone from no cooling required to a heat sink that dissipated heat, to heat sinks with large fans that were the size of a Rubik's Cube.

"As AMD and Intel have learned, if you stick with one big core you get one big heating problem," Tony Massimini, chief of technology for Semico Research, told *internetnews.com*.

The solution became multi-core processing. Multi-core is no different from the old days of multiprocessor computers, only instead of two physical chips, there are two cores acting like two CPUs. Windows sees two CPUs, just as it would if there were two physical, single core chips. Two dual-core chips means the system sees four CPUs, and so on.

"Multi-core gives you the ability to divvy up the work for a complex app and be able to then run these cores at lower frequencies," said Massimini. "You are looking at efficient use of power to handle the power dissipation."

AMD released its first dual-core Opteron processors in April 2005 and its first Athlon 64 dual-core processors one month later. Intel went dual core in May 2005 with the Pentium Extreme Edition.

These processors ran in the 2.0GHz to 2.6GHz range, which is slower than the 3.8GHz of the Pentium IV. But the two cores made them execute faster under a workload because of the two cores. They were also cooler, running at around 80 to 90 degrees Fahrenheit on average, as opposed to the 100 to 110 degrees or more of a Pentium IV.

Two cores eventually turned to four for Intel, with its Kentsfield line, released as the Core 2 Extreme line, in November 2006. Clovertown was the first quad-core Xeon processor for servers, also released in November 2006.

AMD will go quad with Phenom and Barcelona on the desktop and server, respectively, later this year.

It's built, but will they come?

So, if Intel and AMD built all of this technology, will the apps come? Not for a while, it seems. As to what applications parallelization will lend itself to, that's more open to debate.

"There is a class of applications that parallelize very well," Jerry Bautista, director of technology management in the microprocessor research lab at Intel, told *internetnews.com*. "They span from cinematic content creation like Pixar and DreamWorks through home video, gaming and even in financial analytics. These are all a broad class of apps that we typify as taking advantage of model-based computing."

Margaret Lewis, director of commercial solutions for AMD, sees things differently. "The killer app for multi-core is virtualization," she said. "For the desktop it's going to be a little harder for it to take off. In the desktop world, you are one user to a machine. The server is beautiful for multi-core because the server is multiple users, and multiple users means multiple threads."

Virtualization can only happen in the 64-bit world because processors are now free from the restrictions of the 4GB of addressable memory space in a 32-bit processor. A 64-bit chip can, in theory, access 16 exabytes of memory, although hardware vendors are for now sticking to terabytes as the theoretical limit for memory.

"This could only happen with the advent of 64-bits," said Lewis. "The beauty of what's happening is all of these [technologies] are starting to come together to enable virtualization."

Gartner and IDC trends for the last two months reflect this, with fewer machines sold than in the past. But the machines are considerably more "decked out" with much more memory.

One dual- or quad-processor system with 16GB or 32GB of memory is ideal for running dozens of servers in a virtualized environment.

To parallelize an application

The next trick, then, becomes parallelizing an application so it can march in two or more rows rather than a single file. Some processes can never be parallelized, but rather have to be done sequentially, such as applications where one step is determined by the results of the previous step.

In other cases, though, you just can't make the whole application parallel, said Lewis. "You may go to one part of the code that can be highly parallelized, then you go to another part of the code and it's highly serialized and can't be made parallel."

Intel's Bautista agreed. "The major challenges are on the software side. Do we have an appropriate programming environment, benchmarks, and optimizers? That's an issue. A lot of the research is around those areas today."

The problem facing hardware and software vendors alike is that parallel programming is a rare skill and extremely hard to master. Parallel processing has

been around for decades, but programming effectively for multiple processors is hardly a commodity skill like Java or C++ programming.

Massimini said there has to be a breakthrough in that area just as there has been in every other section of computing.

"Someone's going to have to crack [parallel programming]. Otherwise you will never have a better game or computer or app because we will hit a wall. Saying we can't do it is not the answer people want to hear."

The solution has been to put parallel code, libraries and intelligence into compilers to detect segments of code that parallelize well.

"We've got high-level languages today, where I don't think anyone who programs in C thinks about [assembly language]," said Massimini. "You're going to have to develop that underlying layer in the software where someone can program in a higher level of code that translates it back into the op code to provide that parallelism."

Intel has announced a new set of tools to do just that, as has Sun. Intel's new C compiler looks for code that could operate in parallel and is "parallelized," as Bautista put it. "Would it be as good as a programmer skilled at parallel processing? No, but it can come close."

Lewis agreed this is the best short-term solution. "Long-term, we all need to look at what are some different methods for parallelization," she said. "But for now, the things we need to do are shielding the developer from having to understand some of the intricacies of parallelization."

The trick then, is for programmers to catch up to the hardware. Intel is quad-core now. AMD will be when Phenom and Barcelona ship later this year. Then Intel goes to eight cores, and the race continues.

Massimini is the most optimistic that the industry will make full use of every core.

"I like to say software is like a gas. It will expand or contract to fill its available volume. If the hardware community gives them more power, it will suck it up like a leech and take more power."