Software Sea Change: Multi-Core Processing Opens Innovative Business Possibilities

This article appears in its entirety in *Intel® Software Insight* magazine, September 2006.

Table of Contents

(Click on page number to jump to sections)

Software Sea Change: Multi-Core Processing Opens Innovative Business Possibilities

This article appears in its entirety in *Intel® Software Insight* magazine, September 2006.

A Profound Transformation in Programming Paradigms

Every so often a technology change sweeps through the industry and shakes the software world to its roots. Parallel computing, revitalized by newly introduced multi-core processors, is such a change. The shift to parallel computing is turning conventional linear programming paradigms upside down and producing a renaissance of development tools, fresh programming concepts, new models for multitasking, and a slate of opportunities for solution developers and system architects to break new ground and create trend-setting applications. Businesses can look to a future where energy-efficient performance and future-proofing of applications spur innovation, extend product life spans, and offer competitive advantages to those who embrace the technology now.

Amidst the turmoil and restructuring that accompanies any technology change, companies have an ideal opportunity to catch this wave as it crests, adapting applications to take full advantage of multithreading and the energy-efficient performance built into Intel® Core™ microarchitectures. Intel stands behind companies that make the leap with a vast array of enabling tools and technologies, ranging from university-level education and training to multi-threading building blocks to accelerate development. This article discusses the direction and potential of multi-threaded application development, emerging usage models, and the promise of future-proofing application design by aiming beyond dual-core and quad-core processors to the many-core processors that are in the planning stages today.

Innovative Uses Flourish

By the end of 2007, a substantial majority of the new computers sold will contain Intel® dual-core or quad-core technology. The widespread adoption of multi-core processing technology is inspiring a fresh look at computing possibilities and triggering a creative outburst of innovative solution designs. Developers creating client applications can explore new ways to employ multitasking that have been impractical or burdensome in the past. For example, useful tasks can be running all the time—the virus scanner operating continuously and proactively, an automatic backup utility ensuring no work files are ever lost, an intelligent workflow monitor anticipating user requirements and delivering real-time information on cue. The variety of practical applications that will be running in the background on an ongoing basis will continue to grow as multi-core processing becomes more prevalent on client computers.

With additional processing capabilities coming online, development opportunities will move beyond simple performance-related improvements. Voice-activated commands, voice and video over IP, new information-assistance and real-time access capabilities, enhanced IP management, and search and retrieval functions on multiple layers are just some of the capabilities enabling richer and more responsive computing environments for users. The software ecosystem is ripe for additional innovative uses that expand this kind of thinking into entirely new realms.

Intel is committed to helping developers use new development techniques and exploit new processor capabilities through training, education, and new software development products. One goal is to get the next generation of developers to “think parallel.” This change will be a gradual evolution; it is not something that is going to happen overnight. As designers, software engineers, system architects, and solution developers grasp the concept of being able to run multiple threads simultaneously, it opens up whole new worlds in terms of what computers can do.

Likewise, threaded application design expands possibilities for the software development community and innovative uses are already appearing that hint at the potential. A researcher at the University of California at San Diego recently demonstrated a threaded application with life-saving potential. Volcanic eruptions on the island of Java in Indonesia periodically threaten people living in one of the most densely populated areas in the world. To assess threat levels and help establish evacuation plans, the researcher designed a threaded application that concurrently retrieves topographical

details, satellite maps of the region, earthquake activity data, and other information. By combining and layering the images and data, he created a photorealistic 3-D map that clearly depicts fault lines, threatened areas, and potential risks for population centers. Geologists, city planners, emergency officials, news outlets, and others in the Java region now have a threaded tool that can help save lives and assess property risk. The application data is being updated regularly, based on volcanic occurrences. Armed with this information, the government can make arrangements for people who are likely to lose their homes.

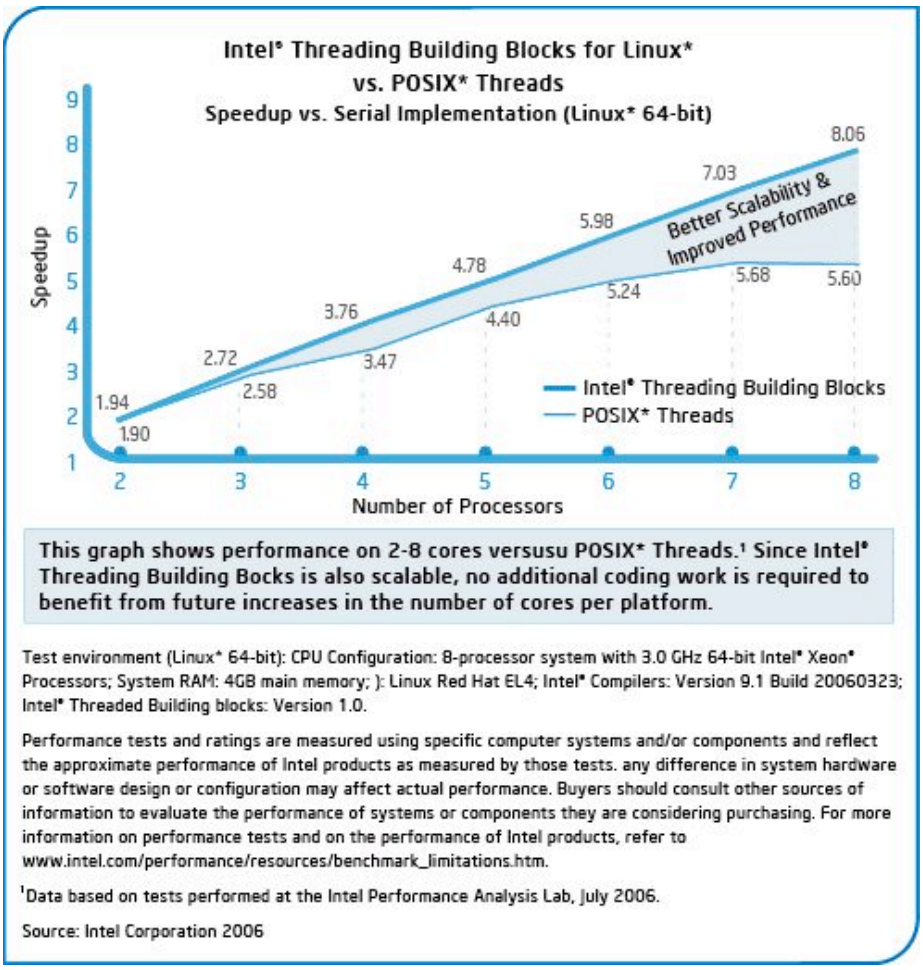
> "Just think: For the first time in the history of computing, mainstream computers will no longer be von Neumann machines—they will be parallel. Given that our applications are going to run on parallel machines, this is a time of enormous opportunity, along with a great deal of work. Sure, concurrency has been done before; parallel computing was researched by some of the very people already mentioned, and companies like Cray have been doing it for years. But the mainstream programmer and mainstream environments have most certainly not been doing it routinely, and we have only now just begun the process of bringing concurrency and parallel programming to the mainstream."
>
> —Dr. Dobb's Journal, August 2006

Future-Proofing Application Design

Threads can take advantage of existing and future processor designs in ways that developers are only now beginning to understand. For example, if an application has been set up for hyper-threading, it can deliver immediate benefits on a dual-core system because it was simply written to use two threads. Hopefully, the application is written to create more than two threads so it can use more than a hyper-threaded processor or a dual-core processor. Ideally, a program is written and prepared to create at least as many threads as processors. Having a program ready to use additional threads, whether from cores or hyper-threading, allows the application performance to scale with the number of processors.

When architecting applications for dual- and quad-core processors, developers have a tendency, because of the way platforms have worked in the past, to think in terms of utilizing two threads or four threads. However, if developers architect programs for as many threads as possible within the program, as additional cores become available, the application will run faster and more efficiently. The practice of threading beyond the immediate number of available cores essentially future-proofs application design and delivers returns on the investment of time and effort that can extend years beyond the application release.

Optimally, developers and software designers should not think in terms of whether they need dual- or quad-core. Instead, it is practical and effective when architecting applications to consider how many tasks can be performed in parallel.



If developers architect for as many threads as possible initially, they gain the benefit of Intel's roadmap for years to come. If you architect to only parallelize two tasks (thinking that dual-core is the target), you miss out on the benefit of the next round of processor advances. As the newer, more capable platforms are released, applications designed for as many threads as possible will be able to run faster.

Software developers will be challenged to create scalable applications, deal with new debugging challenges, and do this while creating maintainable code. In general, software development will need to rely more on abstractions to avoid explicit thread management. Most tool developers need to devote more attention to this area. Abstractions such as libraries, OpenMP*, and Intel® Threading Building Blocks are critical to help realize effective scaling in practice. Finally, having confidence in deploying an application requires dealing with the threat of deadlock and race conditions (common parallel programming bugs). The Intel® Thread Checker is able to help contend with these threats. On a positive note, the tools are improving significantly and, over time, vendors will find ways to further enhance parallel programming. Developers need to understand scalability, debugging, and maintainable code issues while depending on software development tools to simplify programming and ensure correctness when transforming applications to parallelism.

"Taking advantage of Intel's broad range of software development tools for threading and their extensive expertise, we were able to see a 70 percent performance increase when our mp3 and mp3 SURROUND codecs ran on Intel® Centrino® Duo mobile technology. This results in a greater listening experience for consumers."

—Rocky Caldwell, General Manager of mp3 Licensing, Thomson

Introducing Intel® Threading Building Blocks

The available Intel Software Development Products that support parallelism include libraries, thread checkers, compilers with OpenMP* extensions, debuggers, and performance analyzers. Complementing this capable collection of products, Intel just introduced Intel® Threading Building Blocks, a cross-platform C++ template-based runtime library. It focuses on bridging the gap between complex, lower-level multithreading development tools and higher-level, ready-to-use algorithms. Using these templates, developers can immediately build tested, scalable parallel constructs into existing code. Ease of programming, an emphasis on the correctness of threaded algorithms, and provisions for scalability characterize the Intel Threading Building Blocks. These characteristics and capabilities should attract C++ developers who are less experienced with threading concepts and quickly bring them up to speed creating thread-safe, high-concurrency applications.

With support for Microsoft Windows*, Linux*, and Mac OS X* on 32- and 64-bit platforms, these building blocks will likely become a much-used tool for forward-looking development groups targeting multi-core processor implementations with an interest in streamlining deployment. The appeal of this product is also broadened through compatibility with Intel, Microsoft, and GNU compilers. Intel threading tools equip software engineers with a fast path through the complexities of application threading, an advantage that can save companies time and money and keep software projects focused and on schedule. For additional details, visit: www.intel.com/software/products/tbb

Energy Efficiency Comes with the Package

In the past, advances in processor performance generated higher power consumption and higher operating temperatures. Intel Core microarchitecture breaks that mold by delivering vastly improved performance-per-watt, creating an environment for a new generation of computing solutions:

- Compact high-performance devices that run cooler
- Server rooms that consume less power and need less air conditioning
- Mobility solutions that extend battery life
- Consumer and business solutions that were physically impossible with past-generation processors

Virtually all platforms benefit from the energy efficiency underlying the architecture transition to multi-core processing.

Coping with the Change

Looking to the future, James Reinders, director of marketing and business for the Intel® Software Development Products division, says, "It strikes me that in terms of future development, the magnitude of the change that software developers are going to experience will be substantial. A decade from now, we'll be looking back and thinking how much differently we approach writing program code. Parallelism, for everyone, is going to be ubiquitous. And, truthfully, this offers significant opportunities for companies and new possibilities for every single software developer, including tools developers such as our team in the Intel Software Development Group. Right now we have an excellent opportunity to rethink and plan programming strategies for the next decade and find ways to fully exploit the potential of multi-core processor architectures."

Many of the advantages of multiprocessor architectures are available immediately, without requiring any modification of application designs. Taking full advantage of threading possibilities opens additional capabilities and helps achieve optimal use of all available processing resources. As developers embrace the paradigm shift that is unfolding, the nature of multi-core processing redefines the performance map for the future. Fortunately, a wealth of educational resources, training opportunities, and technical expertise exists and can be tapped to lessen the impact of this change.

Take Action

Actions for IT Groups

- Architect your server deployments for cooler-running, higher-performing operation with systems based on the latest multi-core processors from Intel.
- Identify enterprise application bottlenecks and performance-limiting factors using software development tools from Intel.
- Make business processes more efficient through effective use of multi-threaded application design.
- Improve the productivity and effectiveness of staff members through better use of multitasking in business suites and applications.

Actions for Software Developers

- Incorporate threading into new and existing applications to support maximum processor cores and threads.
- Educate and train your staff members to exploit multi-core processor features and future-proof application designs.
- Identify and remove I/O memory bottlenecks.
- Validate your applications using the latest Intel® Software Development Products.
- Understand where your operating system of choice is threaded and take full advantage.
- Deliver products that enhance end user experiences through richer media processing, liberal multitasking, and multi-threading performance advantages.

Summary

This is the time for ISVs and solution developers to become fully engaged in the transition and begin forging a path to help ensure the emerging performance and multitasking capabilities will be well supported in their application designs.

The parallel computing transformation is underway and those businesses that rise to the occasion have an excellent opportunity to seize a competitive edge, future-proof their applications for greater revenue and increasing performance as many-core processors become available, and redesign their applications with enhanced capabilities that will excite customers, deliver improved responsiveness, and chart new innovative paths in the software world. Companies that wait may be left behind.

More Info

You can discover much more about Intel's software technologies and parallel computing efforts by visiting the following areas of the Intel Web site:

Intel® Software
Intel® Threading Tools
Energy-Efficient Performance
The Evolution of Parallel Computing

—End of Technology@Intel Magazine Article—