Q&A: NI LabVIEW 8.5 Simplifies Multicore Development

Dr. James Truchard, NI president, CEO and co-founder, discusses the benefits of multicore processing and how National Instruments LabVIEW graphical system design software takes advantage of the next generation of processors.

Q: How does the shift to multicore architectures improve system performance?

Dr. Truchard: Historically, as processor speeds increased over time, applications automatically ran faster by taking advantage of newer processors. Faster clock speeds simply meant faster execution of the instructions in a particular application. Now that processors are moving toward a multicore architecture, the performance improvement comes instead from balancing more work in the same amount of time. Now, programmers must determine how to spread their applications across the processor cores available in these systems. The notion of multiprocessor systems has been around for a long time, so the challenge is not new. The difference is that this multiprocessor, or multicore processor to be more precise, is now the primary way that PCs are advancing in terms of performance. Multicore processing is here on the desktop today, so programmers have to learn new tricks if they are going to take advantage of this new technology.

Q: What technical applications can benefit specifically from multicore systems?

Dr. Truchard: You can make just about any engineering application more parallel to take advantage of the parallel processing architectures found in multicore PCs. A good example is with data streaming. Any application in which you acquire and process or save large amounts of data at high speeds, such as communications protocol design and testing or audio/video processing, can benefit from multicore technology. When systems must acquire and analyze a stream of data, many times the data processing can be a bottleneck. By splitting the application into two parallel loops (acquisition and analysis) on a dual-core system, or multiple loops (acquisition, analysis and data storage) on a multicore system, you can achieve faster throughput. This is known as pipelining. By splitting the load between processors, you can acquire and process data in parallel faster than you can in a series.

Another example is large data set analysis. In any application in which you analyze huge data sets, such as image processing or matrix operations, you can split the work across processors. Rather than pipelining the algorithm as described earlier, you can split up the data into four equal sections and perform the same analysis algorithm in parallel on these subsets of the original data set.

Finally, time-critical systems especially benefit from multicore processing. In some applications, such as high-speed control systems, you may need to separate the time-critical code from other less-important communication or file IO code. Using a multicore system, you can isolate the time-critical portions of your code on one processing core away from the rest of your application for greater reliability and performance.

Q: How do engineers and scientists take advantage of this new multicore architecture?

Dr. Truchard: Multicore systems present a new programming challenge for the software development community in general. Programmers must learn how to split up their applications into parallel sections, or threads, that they can balance across multicore processing cores in their systems. Multithreaded programming is a very complex task that challenges even the most experienced professional programmers. Engineers and scientists building PC-based measurement and control systems must be able to take advantage of the latest PC technology, but they often do not know how to master the

complexities of multithreaded programming. The LabVIEW block diagram approach to programming is designed for engineers and scientists. By designing their solutions as block diagrams, they naturally develop parallel programs. The LabVIEW compiler automatically breaks up these parallel programs into multiple threads for the user and passes these threads to the operating system for assignment to multiple processing cores. This way, engineers and scientists can focus on their solutions without getting bogged down in the details of multithreaded programming and still gain a performance advantage from the latest PC technology.

Q: Will I automatically observe performance improvements when I move to multicore systems?

Dr. Truchard: There is no guarantee that your application will run faster when you move it to a multicore system. It all depends on how parallel your program runs. For example, if you have written your application in a traditional, text-based language such as C or Basic, you probably have not broken up the program into parallel sections with threads. Therefore, if your program runs in a sequential fashion using a single thread, the operating system cannot easily balance the workload across multiple processing cores – meaning your program will probably not run any faster on a new PC, no matter how many cores it has. With LabVIEW, you can naturally build parallel solutions in block diagrams. Furthermore, LabVIEW users automatically take advantage of this parallelism when they develop their applications without even thinking about it. Therefore, most LabVIEW users can expect some performance improvements when they move their applications to multicore machines.

Q: How much performance improvement can I expect when I move to a multicore system?

Dr. Truchard: This is very difficult to answer. Obviously, the theoretical limit on improvement when moving from a single- to dual-core PC is two times, or a 100 percent performance improvement. However, this is almost impossible to achieve because the simple process of swapping between threads introduces overhead in the system. For some very basic LabVIEW applications, we have seen performance improvement upward of 25 to 35 percent when going from single- to dual-core machines with no code modifications, just by relying on the built-in, automatic multithreading that LabVIEW provides. Even with very little parallelism in your LabVIEW block diagram, you can expect some improvement because the LabVIEW compiler automatically splits your application into two threads – one for the user interface, which can be very processor-intensive, and one for the program code.

By employing some of the parallel programming techniques previously described, we have seen simple algorithms implemented in LabVIEW experience 60 to 70 percent performance improvement when they are moved to multicore systems. Again, the key to achieving better performance is the ability to separate your application into parallel threads. With LabVIEW, this task is much more intuitive than learning the intricacies of multithreading in a traditional programming language.

Q: Will multicore systems have an impact beyond the desktop PC?

Dr. Truchard: Yes. Already, many of the real-time and embedded operating system vendors are struggling with the challenge of multicore development. On the desktop, Windows automatically maps threads onto available processing cores. This is a very complex and difficult task to add to an operating system. In the embedded space, where applications also must execute in a deterministic fashion, this is an even bigger challenge. In the latest version of LabVIEW, LabVIEW 8.5, we have managed to port the automatic thread-scheduling from the desktop into the LabVIEW Real-Time operating system. Now, developers of real-time systems can simply develop their applications in a parallel approach with LabVIEW, and LabVIEW Real-Time will automatically schedule, or balance, the workload across multiple processor cores if they are available. In addition, for an extra level of fine-tuning and reliability, developers

can manually assign specific portions of their application to run on a specific processor. By doing this, they can isolate time-critical sections of their code to a single processing core.

Conceptually, the intuitive dataflow approach of LabVIEW for programming multicore systems applies equally well to FPGA development. Because FPGAs are simply huge fields of programmable gates that can be programmed into many parallel hardware paths, the notion of a naturally parallel block diagramming language such as LabVIEW can provide a very intuitive approach for FPGAs. With the LabVIEW FPGA Module, programmers can automatically map their parallel solutions directly to FPGAs, and once again bypass the more tedious, traditional HDL languages for implementing a parallel solution.

In fact, in LabVIEW 8.5, we have added new multichannel filters and PID control algorithms to maximize the use of FPGA resources for industrial machine control applications. Using these multichannel algorithms, we have increased the number of PID channels that a 1M gate FPGA can handle – from eight to 256 channels. In addition, the PID algorithms now execute three times faster in LabVIEW 8.5, or more than 30 kHz loop rates for 256 PID channels.

Clearly, the concepts of parallel programming that are now so critical for taking advantage of new multicore processors found in desktop PCs have equal benefits for optimizing embedded systems based on real-time operating systems and FPGAs.

Q: Is LabVIEW a programming language or a measurement tool?

Dr. Truchard: This is an interesting question posed to us by many prospective users. In fact, LabVIEW is neither a programming language nor a measurement tool per se. LabVIEW is a graphical system design platform that ties high-level design concepts with different hardware platforms. Using LabVIEW, you can quickly design an algorithm using a variety of programming approaches – statecharts, dynamic system modeling tools, textual math – and run these algorithms with real-world I/O to quickly prototype their operations in worldwide measurement or control scenarios. The core technology powering the platform is the graphical programming language. Using the dataflow language, you can map your algorithms directly to hardware, providing a streamlined design-prototype-deploy experience. With multicore technology entering the mainstream, it was critical that we update the language to handle this new technology. Now, LabVIEW 8.5 brings you the power of multicore processors without imposing entirely new programming structures or concepts. As processor architectures continue to expand to more and more cores, we are confident that the LabVIEW platform will continue to scale with your needs.

NATIONAL INSTRUMENTS™
11500 N Mopac Expwy • Austin, TX 78759-3504 USA
Tel: (800) 433-3488 • Fax: (800) 683-9300
info@ni.com • ni.com