



July 10, 2008

## **The Cell Processor Builds Its Mojo**

With the arrival of the new QS22 blade, IBM and its partners are pushing hard to flesh out the software ecosystem for the Cell Broadband Engine. The QS22 contains the double-precision enhanced version of the processor, and as such, represents the first mainstream Cell platform for HPC. Since the QS22 is also at the heart of the new Roadrunner petaflop machine for Los Alamos, there's been even more interest in exploring the application possibilities for the Cell.

This week Georgia Tech announced they are renewing the Sony/Toshiba/IBM (STI) Center of Competence effort that was initiated last year. The center is working with the three STI partners plus a number of other firms and academic groups to help develop libraries, application kernels, and productivity tools for the Cell architecture. The upcoming work will include a new set of application research projects.

One of the more interesting projects will involve using the processor to perform real-time monitoring of an aircraft's structural components while in flight. The computer system would be hooked up to a network of sensors distributed around the aircraft to measure vibrations at key locations. The Cell processor would be called upon to perform the heavy-duty FFTs required to process the structural fatigue that is taking place across the fuselage, wings, and engine assemblies. The application is meant to act as an early warning system for pilots, giving them enough time to take the appropriate action (most likely, landing the plane) before a catastrophic failure occurs. The system is being considered for both military and commercial aircraft.

To help develop this application, Georgia Tech is working with engineers at Pratt & Whitney and researchers at the University of Connecticut. David Bader, HPC director at Georgia Tech, says that because the processor has cores to spare it would also be possible to task a Cell-based system to simultaneously provide processing for both an in-flight entertainment system and the structural diagnostics function, allowing commercial airlines to get double-duty from the Cell hardware.

"This is one example of the vision of the STI Center at Georgia Tech," Bader told me. "We're really looking to make innovations at the application level."

Other application areas being looked at include data/file compression codes, financial services apps, encryption libraries, multimedia encoders/decoders, and bioinformatics codes. The STI Center is also working on software productivity enhancement tools that involve a cross-platform profiler, performance estimation and tuning system with IDE type features. In

addition, they will be developing an automatic translator that takes a C/C++ application and spits out the appropriate code for the Cell's PowerPC and SPU (Synergistic Processing Unit) cores.

Even though some commercial partners are involved in these efforts, all software produced will be open source, with some of the libraries and tools likely ending up in IBM's own Cell SDK. At this point in the processor's life, this is the only rational model, since new architectures like Cell will need a heavy infusion of software to become a popular platform for system integrators and third-party software developers. It's the typical "chicken-and-egg" problem all new architectures must confront.

One of the elements that is being talked about in the nascent Cell community is a standardized development environment. Today the different form factors -- IBM QS2X blades, the PlayStation game console, Toshiba's Cell Reference Set, and the various platforms from Mercury Computer Systems -- use different compiler/runtime systems. A standard compiler and runtime environment would help create a richer environment for developers and allow users to share code more easily. Bader says there is some consensus now to come up with a common runtime environment that would be compatible across all the major form factors.

But the big knock on Cell is that it's just hard to program. The processor implements a rather unique heterogeneous architecture -- one general-purpose PowerPC core that manages 8 SIMD SPU cores. The SPU cores use a local store in place of cache and the SPUs talk with main memory via DMA. At this point, it's not possible to just recompile an existing application for this processor. Developers need to break apart the program and rejigger it for the new architecture.

Bader says they're trying to dispell the myth that it's hard to program. Rather, he thinks it's "a processor ahead of its time." He claims that the Cell is not really difficult to program directly, once the underlying architecture is understood. Once that point is reached, Bader says programmers comment on how flexible the architecture is. If you're not disposed to code to the bare metal, there are a few higher level environments available, including an OpenMP compiler from IBM, the RapidMind development platform, and the Gadae compiler. As time goes on, Bader believes "we'll see a lot more of these frameworks that are portable across the Cell and other processors."

**Find this article at:**

<http://www.hpcwire.com/blogs/24391682.html>

Check the box to include the list of links referenced in the article.

Copyright © 1994-2008 Tabor Communications, Inc. All Rights Reserved.