Dynamic Kernel Instrumentation on Clusters

Nathan DeBardeleben, William Rust, HPC-4

High-performance computing clusters are increasingly complex systems that bring together disparate subsystems of CPUs, memory, I/O, networks, and file systems. All of these subsystems can affect the performance and stability of the others, leading to a complex balance that must be maintained. As we move to new processors such as the IBM Cell Broadband Engine™ (Cell BE) (Fig. 1) in the Los Alamos National Laboratory Roadrunner supercomputer, we can expect these effects to get worse, not better. The emphasis on correctly tuning applications, system software, and operating system kernels will grow to be even more important for overall system performance.

We have been studying the technology of dynamic kernel instrumentation[2, 3] using SystemTap [1] on a small testbed cluster designed to mimic Lightning, a large machine used at LANL for the Advanced Simulation and Computing program. We have already used this technology to instrument the virtual file system layer to look at potential parallel file system problems. Additionally we have written probes to trace kernel calls, monitor specific files, and examine network activity and assorted other hot spots in the kernel.

With the increasing emphasis on multicore systems and, in particular, the Cell BE processor for Roadrunner, we extended this dynamic probing technology to the Cell BE kernel. The kernel modifications for the Cell BE are vast and rapidly changing. As with any new technology put into the kernel, there are problems that take time to work out. Furthermore, the complexity of the Cell processor is revealing many nuances about how to correctly program applications for it when one looks at how they affect the kernel. By probing a running Cell kernel, we are able to debug and study the complex interaction between the main processor and the Synergistic Processing Units. With this technique we can look at how a system is behaving without the need to build a custom kernel with debugging features always enabled. This ability to attach to a kernel and then detach at a later point is important in production machines because we do not want to be running with debug kernels waiting for a problem to arise. Furthermore, with the complexities of these systems growing, it is often impossible to predict what a system analyst will need to examine. Therefore, having the ability to apply a dynamic kernel modification is important for looking at a problem as it arises.

Conclusion

As we experiment with this developing kernel, we are finding many interesting places to glean detailed information about a running system. We believe carefully placed kernel probes can be used to allow the kernel to self-diagnose problems and report them to a system analyst. It may even be possible to build an expert system that attempts to tune itself when it recognizes problems.

For more information contact Nathan DeBardeleben at ndebard@lanl.gov.

[1] Redhat Linux, *Systemtap Open-Source Project,* http://sources.redhat.com/systemtap/
[2] V. Prasad, et al., "Locating System Problems Using Dynamic Instrumentation," *Proceedings of the Linux Symposium* **2**, 49 (July 2005).
[3] A. Tamches and B. P. Miller, "Using Dynamic Kernel Instrumentation for Kernel and Application Tuning," *Intl. J. High Perf. Comput. Appl.* **13**, 263 (1999).

Funding Acknowledgements
This research was supported by the NNSA tri-Lab Advanced Simulation and Computing Program.

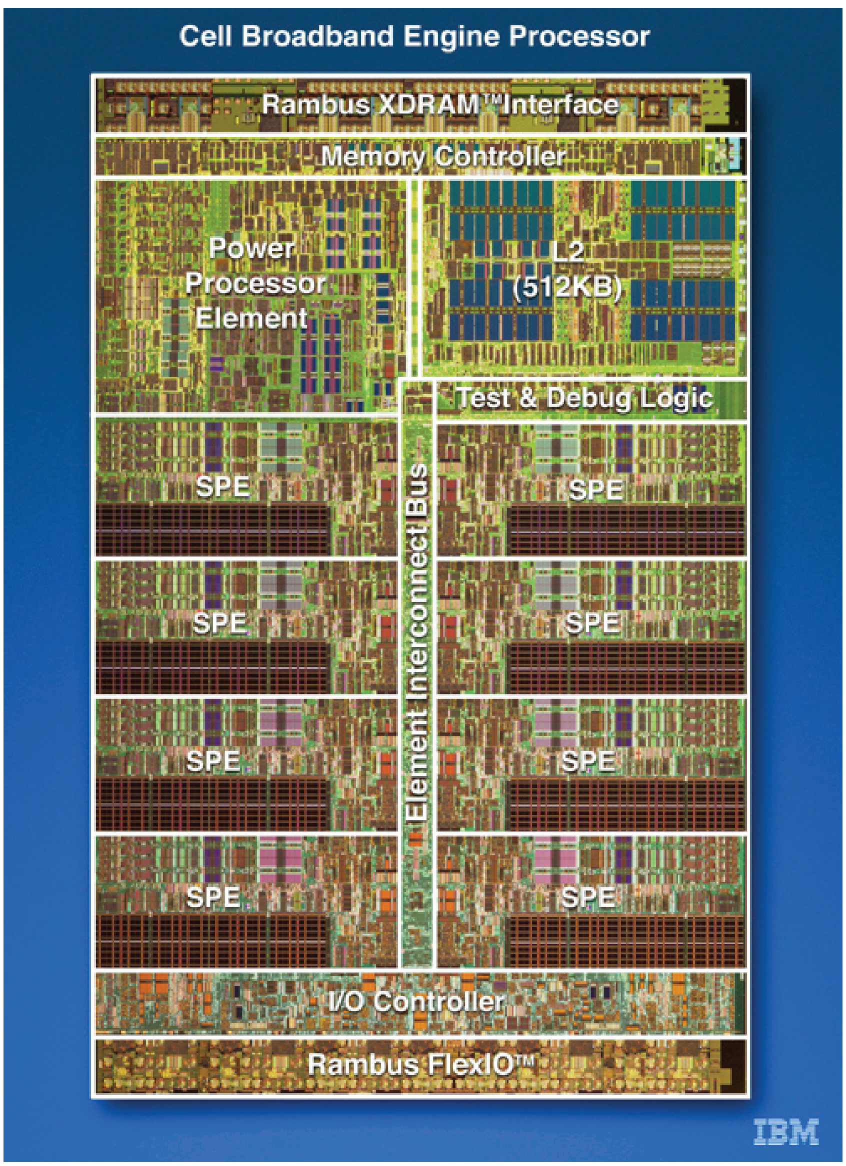

Fig. 1.
Developing dynamic probing technology for the complex Cell Broadband Engine™ processor is a goal of this research. Image courtesy of IBM.