Department of Energy DE-FG02-05ER25686
Early Career Principal Investigator Award Final Report

Prof. Derek Chiou, The University of Texas at Austin
May 11, 2013

Introduction

This project is researching FAST, a methodology to build very fast, cycle-accurate full system computer simulators and building the first set of such simulators and the first set of tools to help construct those simulators. The methodology relies on a functional model that is a fast, full-system but not cycle-accurate simulator coupled with a timing model that is models the micro-architectural structure and arbitration of a computer system, but not its functionality. The way FAST simulators differ from other simulators partitioned in the same way is that the interface between the functional and timing model is optimized to minimize the need for round-trip communication. The optimized communication enables FAST timing models to be implemented in an FPGA and the functional model in software while still achieving extremely high performance.

As specified in the original proposal, the overall project goals are as follows:

1. Development of an initial prototype FAST system that models the IA-32 instruction set running on a 80486-like timing model. This system will run the entire SPEC benchmark suite on top of unmodified Linux and will run on an FPGA.
2. Development of an aggressive FAST system that models the IA-32 instruction set running on an out-of-order, superscalar timing model. This system will run arbitrary applications on top of both Linux and Windows and will run on an FPGA.
3. Development of a CMP FAST system that models the IA-32 instruction set running on a CMP timing model with simple cores. This system will run arbitrary applications on top of Linux and will run on an FPGA.
4. Development of a timing model language to define timing models using timing model modules and an associated compiler that will assemble a working timing model from a specification. Timing models will be developed and modified using the language and compiler.
5. Development of a functional model language to define functional models and an associated compiler that will assemble a working functional model from a specification. Functional models will be developed and modified using the language and compiler.
6. Development of statistics/debugging/performance monitoring additions to the timing model language and functional model language to allow users to specify statistics counters and breakpoints on a variety of conditions. Real DOE applications will be used to test these additions.

Our general strategy is to design a methodology and then prototype an example using that methodology to ensure the methodology is sound before committing to it. This strategy ensures that the selected methodology has been tested and provides an early example of the output of that methodology.

For clarity, we use the term “host” to mean the machine that the simulator runs on and “target” to mean the machine being simulated.

The next section of the progress report addresses our current status on each of the proposed objectives. The following section describes the planned activities for the next year.

Final Status: April 15, 2007 to Project End

During the past 11 months, we have made significant progress in a variety of areas listed below.

1. New host platform. We received a new host platform (ACP) as a donation from Xilinx and Intel, that consisted of a stacked FPGA board inserted into an Intel server processor socket. The FPGA communicated over the Intel front side bus (FSB), enabling it to issue commands to memory that would be kept coherent with transactions on the bus. We ran into serious problems with this platform that will be discussed in the next section.
2. Functional model. We completed an initial port to QEMU, another open source full-system simulator that supports multiple ISAs and runs much faster than Bochs.
3. Timing model. We built an out-of-order timing model that runs on the DRC platform along with our QEMU-based functional model.
4. CMP target/host. We finished work on the CMP host platform based on Sparc V8 cores, but switched to the ACP platform which meant using the standard Intel Xeon host processors to run a parallelized version of the functional model.
5. Continued Intel collaboration. I continued to work with Joel Emer, an Intel Fellow who leads much of Intel’s internal simulation work.

Host Platforms

The DRC platform that we had obtained was supposed to be improved to reduce latency to 200ns. We spent quite a bit of time getting upgraded boards with larger Xilinx FPGAs that would also reduce latencies. The last generation dropped read latencies down to 400ns, double the promised improvements to 200ns, but much less than the original 3000ns. Our experience with the DRC platform was mirrored by Los Alamos, who preferred a product from a competitor, XtremeData. Unfortunately, we did not have access to that system.

Around the same time, Xilinx and Intel were co-developing a system, called the "ACP", that consisted of a stack of two boards plugged into an Intel server processor socket. One of the FPGA boards ("Compute") had two of the largest Xilinx FPGAs at the time (5V330), whose logic was almost entirely available to the user, along with 8MB of SRAM attached to each FPGA. The second FPGA board ("Interface") had an FPGA dedicated to communicate with the Intel front side bus (FSB). The Intel server has four processor sockets, each supporting either an FPGA board stack or high-end Intel Xeon processors that contained up to six cores, and up to 256GB of RAM. Thus, we could have an 18 Intel core system with 2 large FPGAs, a very impressive system. Xilinx donated us a set of the boards, and Intel donated the server as well as the processors for the system. Xilinx put us in touch with Arches, a company started by Paul Chow of the University of Toronto that was building some of the IP infrastructure necessary to communicate between the FSB interface board and the compute board.

However, the ACP system was fundamentally a disaster that wasted two years of time to the end of this project. It was fundamentally unreliable, flipping a significant number of bits at a rate far too high to correct with any reasonable error correction scheme. We spent a significant amount of time trying to make the system reliable enough to be useful but, in the end, we failed, even with the help of both Xilinx and Intel.

Functional Model

Since the last report, we were able to get our QEMU functional model to the point of being able to boot Windows XP and run standard applications such as Explorer and Microsoft Word on top of it. In fact, we made a video of us typing into Microsoft Word running on top of Windows XP that can be viewed at the following link (http://users.ece.utexas.edu/~derek/win_word_MICRO2.avi). Thus, we had a complete uniprocessor functional model.

We started work on a multicore target functional model, which added significant complexity. At the completion of this project, we had not entirely completed the multicore functional model.

CMP FAST

We built RAMP-White, an FPGA-based multicore processor [CAS 2007]. However, we used the Leon 3 processor, which was a Sparc V8 processor, that had limited performance. With the adoption of the ACP platform, we decided instead to use the multicore Intel host platform and started parallelizing QEMU towards the end of this project. We did finish a parallelized version of QEMU that ran at 90%+ efficiency on multicore x86 hosts after this project ended.

We did not complete a CMP FAST timing model within this project, but did develop an elegant strategy to enable very aggressive parallelization of the FAST functional model, while still maintaining perfect cycle accuracy. To achieve that capability, we invented a "target memory oracle" that has the functional model log load and store values, pass them to the timing model that models stores at the correct target time to the target memory oracle and compares functionally loaded values with timing model loaded values from the target memory oracle[CAL 2009]. In the case that the functional load value and the target load value are identical, the functional model execution is target-correct, even if the load is reordered by the target. Only in the case where the functional load value and the target load value are different does functional rollback occur and is corrected with the target correct load value.

The new scheme required significant overhaul of the functional model, to support correction of load values, the interface between the functional model and the timing model, and the timing model itself. Those changes were made after this project completed.

FAST Simulator Performance

Our final FAST simulator performance for our uniprocessor system running on a DRC platform was roughly 5MIPS, twice the performance that was obtained at the end of the second year. That simulator modeled a two-issue single core with eight-way 32KB L1 instruction and data caches, an eight-way 256KB shared L2 cache, 64 ROB entries, 16 shared reservation stations entries, 16 load/store queue entries, a 4-way and 8K BTB gshare branch predictor, multiple branch units, one load/store unit, eight general-purpose ALUs and up to four nested branches. The pipeline is between eight and ten stages deep, not including accesses beyond the L1 caches. That target is comparable in complexity to some of the best academic simulators of the time.

Interactions with DOE Researchers

In the last two years of the project, I interacted with several DOE researchers, including Dr. Jeff Vetter of Oak Ridge and Dr. Arun Rodrigues of Sandia in simulation workshops (IAA Interconnect Workshop in July 2008) and on white papers for DOE.

Summary of Progress Relative to Stated Goals

- Goal 1 (80486-like model) completed in Year 1.
- Goal 2 (complex out-of-order processor) completed in Year 3 [MICRO 2007].
- Goal 3 (CMP model) Due to issues with the ACP system, we never completed this goal within the timeframe of the project.
- Goal 4 (timing model language and compiler) completed in Year 2 by a new factorization of the timing model into Modules and Connectors.
- Goal 5 (functional model language and compiler) completed in Year 3 through a micro-op compiler taking a functional model and generating micro-instructions for approximately 99% of dynamic instructions, with a fallback to a general super instruction.
- Goal 6 (statistics gathering/monitoring) completed in Year 3 and used to generate results for the MICRO 2007 paper. We gathered statistics in the context of the Connectors. We also integrated with the Intel Asim infrastructure that can provide visualization capabilities.

Students Supported

This grant provided the tuition for approximately three graduate students for the first two years of this project, while their salaries were provided by startup money from the University of Texas. This grant provided full funding for third and fourth years

Publications

Many publications were written as a result, at least in part, of the funding from this project. Those publications are listed below.

- John Wawrzynek, David Patterson, Mark Oskin, Shih-Lien Lu, Christoforos Kozyrakis, James C. Hoe, Derek Chiou and Krste Asanovic. RAMP: Research Accelerator for Multiple Processors. IEEE Micro, 27(2):46-57, March/April 2007.
- Hari Angepat and Dam Sunwoo and Derek Chiou. RAMP-White: An FPGA-Based Coherent Shared Memory Parallel Computer Emulator. 8th Annual Austin CAS Conference, March 2007.
- Derek Chiou, Hari Angepat, Nikhil A. Patil, and Dam Sunwoo. "Accurate Functional-First Multicore Simulators." Computer Architecture Letters, July 2009.
- Dam Sunwoo and Joonsoo Kim and Derek Chiou. "QUICK: A Flexible Full-System Functional Model." Proceedings of ISPASS, April 2009.
- Derek Chiou, Dam Sunwoo, Joonsoo Kim, Nikhil Patil, William Reinhart, Eric Johnson, Jebediah Keefe and Hari Angepat. "FPGA-Accelerated

Simulation Technologies (FAST): Fast, Full-System, Cycle-Accurate Simulators." Proceedings of MICRO, December 2007.
- Derek Chiou, Dam Sunwoo, Joonsoo Kim, Nikhil A. Patil, William H. Reinhart, D. Eric Johnson and Zheng Xu. "The FAST Methodology for High-Speed SoC/Computer Simulation." Proceedings of International Conference on Computer-Aided Design (ICCAD), November 2007.
- Derek Chiou, Huzefa Sanjeliwala, Dan Sunwoo, Zheng Xu and Nikhil Patil. "FPGA-based Fast, Cycle-Accurate, Full-System Simulators." Proceedings of the second Workshop on Architectural Research using FPGA Platforms (WARFP) held in conjunction with HPCA 2006, Austin, TX, February 2006.

Summary

I believe this project was quite successful overall, through the research and development of an aggressively accelerated simulation methodology. The work was extensively published and has spurred collaboration with Intel and many academics. Though we were never able to run real workloads, due to a defective host platform, I believe we were able to significantly advance the state-of-the-art in cycle-accurate capable simulators.

I thank DOE for providing the funds to do this work.