Evaluating System-wide Monitoring Capsule Design using Xilinx Virtex-II Pro FPGA

Taeweon Suh
Georgia Institute of Technology
School of Electrical and Computer Engineering
suhtw@ece.gatech.edu

Hsien-Hsin S. Lee
Georgia Institute of Technology
School of Electrical and Computer Engineering
leehs@ece.gatech.edu

Sally A. McKee
Computer Systems Lab
Cornell University
School of Electrical and Computer Engineering
sam@csl.cornell.edu

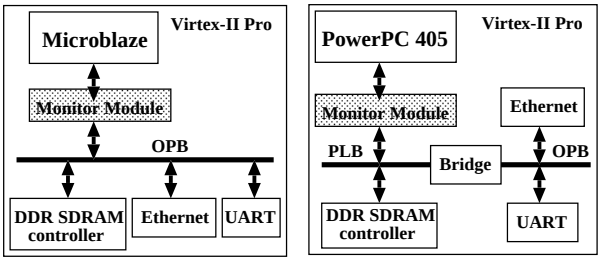
Martin Schulz
Livermore National Lab
Livermore, CA
schulzm@llnl.gov

The growth of system complexity increasingly hinders the understanding of system behavior. Meanwhile, more autonomous and robust systems that provide self-healing capability, fault-tolerance, and immunity to malicious attacks are in greater demand. Owl [1] is a proposed framework that overcomes the limitations faced by traditional performance counters and monitoring facilities. Owl addresses such complexity by pervasively deploying programmable monitoring elements called *monitor capsule* throughout a computing system at the microarchitecture and system architecture levels. These monitoring capsules can be programmed to collect and analyze system activities, and further optimize the performance and guarantee the robustness of the system in an active manner. In the following, we examples four usage models of Owl.

- Memory access logging compression & filtering: monitoring modules can perform compression such as run-length encoding, semantic trace compression and filter out unnecessary information, for instance, logging only accesses corresponding to address regions.
- Cache access histograms: cache-line eviction frequency, average total memory access times, and reuse distance can be measured using an associative counter array in monitoring modules.
- Memory access optimization: monitoring modules identify common access patterns and communicate back to compiler for optimizations such as prefetching or cache-conscious data placement.
- Network attack prevention: malicious network attacks such as buffer overflow can be identified and prevented by observing memory references in monitoring modules.

Our microarchitectural-level simulation results [5] using SPEC2000 benchmarks show that such a monitoring system has a relatively small impact on system performance and that with relatively low monitoring traffic rate or infrequent monitored events such as those between L2 and main memory, the overhead becomes negligible.

This abstract presents a preliminary framework for evaluating the design of a monitoring capsule in Owl using a Xilinx ML310 [2] board based on the Virtex-II Pro FPGA. ML310 provides a PC-like environment for system evaluation and development, featuring XC2VP30 (Virtex-II Pro), 256MB DDR SDRAM DIMM, Ethernet, etc. Two PowerPC 405 processors are fused into XC2VP30, and a softcore called Microblaze [3] can be integrated into XC2VP30 using Xilinx Platform Studio (XPS). PowerPC 405 and Microblaze platforms adopt the CoreConnect [4] bus, consisting of the Processor Local Bus (PLB), On-Chip Peripheral Bus (OPB), and Device Control Register (DCR) buses. In our initial study, we deploy a monitoring capsule in Dcaches to monitor the memory behavior between a CPU and L1 Dcache.



(a) Microblaze-based (b) PowerPC-based

Figure 1: Configurable Platforms using Xilinx Virtex-II Pro

Whenever the processor accesses the L1 Dcache, the monitoring capsule compresses address, data, and control information such as read/write and access length, and sends it to the reserved address space for monitoring in the external memory (DDR SDRAM DIMM). To avoid voluminous monitoring traffic, rules such as sending only write transactions can be applied in the monitoring capsule. Consumers of the monitored data can process data online and analyze the system behavior dynamically.

Figure 1 demonstrates two example platforms configured using XPS. Figure 1(a) shows a Microblaze platform. A Dcache with a monitoring capsule is placed between Microblaze and OPB to intercept data traffic and send appropriate packets to memory. Figure 1(b) illustrates a PowerPC 405 platform. We deploy a Dcache with a monitoring capsule between the PowerPC 405 and PLB as shown. Peripherals (UART and Ethernet interfaces) attached to the OPB bus are used in both platforms for debugging. The SPEC2000 benchmarks run on top of linux on both platforms. System perturbation of adopting a monitoring capsule and relevant system optimization techniques will be investigated using each platform.

1. REFERENCES

[1] http://owl.csl.cornell.edu.
[2] http://www.xilinx.com/products/boards/ml310/current/index.html.
[3] http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf.
[4] http://www-03.ibm.com/chips/products/coreconnect.
[5] Martin Schulz et al. Owl: Next Generation System Monitoring. submitted for publication.