

Memory Subsystem Performance Evaluation with FPGA based Emulators

Shih-Lien Lu*, Eriko Nurvitadhi[‡], Jumnit Hong[†], Steen Larsen*

*Intel Corp.
2111 N.E. 25th Avenue,
Hillsboro, OR 97124
sllu@jf.intel.com

[‡]ECE Department
Carnegie Mellon University
Pittsburgh, PA

[†]School of EECS
Oregon State University
Corvallis, OR

The performance of a computer system like any other systems is affected by its weakest component. As the latency difference between CPU and memory continues to grow, memory subsystems are becoming the main bottleneck that effectively dictates the performance of the entire system. The criticality of such a problem has been evident from the plethora of existing memory system studies. With this, various ways to conduct a better memory study have been proposed. One of the typical ways to perform such a study in the recent days is through the use of a simulator.

Trace-driven simulation could be considered as one of the most well-known simulation methods. Here, traces of a system are collected and stored in a particular form. Then, they are used as inputs to a software simulator, which models the memory system of interest. Typically the simulator would emulate in detail the system under study as if it was processing these traces. Thus, the behavior and performance of the experimental system can be observed and assessed without the need for actually building the system. Further, various parameters of the system can be altered and their impact on performance can be evaluated, allowing for design space exploration. One disadvantage of this method is that the simulator is limited by the size of the traces, which could make it difficult to conduct a trace-driven simulation for long-running complicated workloads.

Another well-known simulation method for memory study is full-system simulation. In this approach, the software simulator models the entire system under study. The main benefit of this approach is that the impact of the benchmark of the entire system can be studied. Furthermore, the design space explored is not limited on a particular subsystem. Instead, it is expanded to multiple subsystem models that constitute the overall simulated system. However, simulating a complete system is complicated and may require a long time. Additionally, it may be more difficult to provide a very detailed software model for each of the subsystems. Finally, setting up a benchmark on a simulated system is typically more difficult than on a real system.

Recently, hardware-based trace-driven simulators have been proposed for complementing the aforementioned software-based simulators. With hardware-based simulators, the modeling of the subsystem of interest is done by hardware that processes system traces in real-time and generates the desired simulation statistics. By doing so, simulation time and trace size requirements can be reduced significantly as hardware speed is achieved and non-volatile trace storage eliminated. Further, since an actual system is used, there is no need to develop multiple subsystem models to

simulate a full system. Only the memory subsystem under study needs to be modeled by the hardware, other aspects of the system will utilize real components. Lastly, setting up benchmarks on a real system equipped with hardware simulators would be more trivial than on a completely simulated system.

However, existing hardware simulators are passive. They only monitor and collect data from their host systems, and do not emulate the impacts of the simulated subsystem on the real system. Thus, emulation of the idea under study is conducted within the boundary of the hardware emulator only, and within its subsystem context. For example, the performance measurements of an experimental cache structure evaluated by hardware emulator can only be obtained from the simulation statistics generated by the hardware (e.g. miss ratio). The impact of this experimental cache design on the performance of the whole system (e.g. the change in the CPU's IPC) cannot be obtained because they are not emulated within the system context.

One concept to extend passive emulation is active emulation. The idea is to have the hardware emulate the impact of an experimental design under study on its host system, thereby allowing for the overall system to perform as if the experimental subsystem was integrated within it. We have implemented an Active Cache Emulator (ACE), a hardware simulator that actively models L3 cache through injection of delays to its host system's Front-Side Bus (FSB) using this concept. Latency scaling (time dilation) is applied to the ACE framework to reason with the simulation results and extract the overall system performance.

There are some limitations with time dilated emulation system. By implementing an existing chipset in FPGAs, the FPGAs can be modified to examine different possible performance enhancements. We propose to replace the chipset of an existing board with FPGAs where various memory subsystem architectural proposals could be tested out in a real OS environment running real applications with no time dilations. Memory compression, chipset prefetching and virtual memory channels are a few things we have in mind in evaluating. New DRAM technology such as fully buffered DRAM can also be exploited.

References

- [1] S. A. Herrod, "Using Complete Machine Simulation to Understand Computer System Behavior", *Ph.D. Thesis*, Stanford University, February 1998.
- [2] R. A. Uhlig and T. N. Mudge, "Trace-driven Memory Simulation: A Survey", *ACM Computing Surveys*, 29 (2), 1997, pp. 128-170.
- [3] S.-L. Lu and K. Lai, "Implementation of Hardware Cache Simulator (Hw\$im) - A Real Time Cache Simulator", *Proc. of FPL 2003*, Portugal, Sept. 2003