Tejas Simulator : Validation against Hardware

Smruti R. Sarangi, Rajshekar Kalayappan, Prathmesh Kallurkar and Seep Goel
Computer Science and Engineering
Indian Institute of Technology, New Delhi, India

Abstract—In this report we show results that validate the Tejas architectural simulator against native hardware. We report mean error rates of 11.45% and 18.77% for the SPEC2006 and Splash2 benchmark suites respectively. These error rates are competitive and in most cases better than the numbers reported by other contemporary simulators.

I. INTRODUCTION

This article serves to establish Tejas as a validated micro-architectural simulator. To do so, a range of serial and parallel benchmarks were run on a PowerEdge R620 server (for details see Table I). The linux “perf” command was used to measure the number of cycles taken to execute the benchmarks. The same set of benchmarks were then run on the Tejas simulator, configured to mimic the DELL server as closely as possible (see Table II for details). The comparison of the simulated cycle counts with their hardware counterparts is given in the next section.

II. RESULTS

Figure 1 shows the results for a set of 17 benchmarks from the SPEC2006 suite. We compute the time it takes for a benchmark to complete on native hardware (averaged across 10 runs). Then, we simulate the benchmark on Tejas, and compute the absolute error. The average absolute error is 11.45%. 10 out of 17 benchmarks have an error less than 10%. Only 4 benchmarks have errors in the 20-30% range (sjeng, astar, mcf, and gcc).

Figure 2 shows the results for a set of 11 benchmarks from the SPLASH-2 suite. The average absolute error was observed to be 18.77%. It has been computed the same way as was done for the case of sequential benchmarks. In this case, the average error is more primarily because the jitter introduced by the operating system is not predictable, there are hardware events that induce jitter, and lastly we are not privy to all details of the operation of the cache coherence protocols in Intel systems. Only 3 benchmarks had errors more than 25% namely radiosity, radix and water-spatial. For most of the benchmarks, the error ranges from 10 to 17%.

III. COMPARISON WITH OTHER SIMULATORS

Let us now put our numbers in the right perspective by comparing similar numbers obtained on other simulators. We shall observe that Tejas is more accurate on both serial and parallel benchmarks as compared to most of the other widely used architecture simulators (for which published results are available).

MARSS [5] is a cycle-accurate simulator based on PTLSim. It is a tool built on QEMU and provides fast and full system simulation. MARSS has been validated against a x86 target machine with the Intel Xeon E5620 processor. For the SPEC CPU 2006 benchmark suite, it has errors ranging from -59.2% to 50%, with an average absolute error of 23.46%.

Sniper [1] is an approximate simulator. It tries to find a middle ground between the simulators that are fast but inaccurate and the simulators that are accurate but slow. It has been validated against a 4-socket Intel Xeon X7460 Dunnington shared-memory machine. An average absolute error of 25%...
IV. CONCLUSION

Tejas has been demonstrated to be a reliable simulator that provides a highly accurate reflection of the simulated hardware. It must be noted that many details of the underlying hardware are not known – the branch predictor, the coherence protocol, the NOC parameters, the select/data forwarding logic, to name a few. Better knowledge of these will allow further reduction of the error in the accuracy. The incomplete knowledge aside, Tejas shows an average absolute error of 11.45% in serial benchmarks, and 18.77% in parallel benchmarks, which is 5-10 percentage points better than some of the most popular architecture simulators currently in use (as of 2014).

REFERENCES

[1] Wim Heirman, Trevor Carlson, and Lieven Eeckhout. Sniper: scalable and accurate parallel multi-core simulation. In 8th International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems (ACACES-2012), pages 91–94. High-Performance and Embedded Architecture and Compilation Network of Excellence (HiPEAC), 2012.

[2] Shobhit Kanaujia Irma Esmer Papazian Jeff and Chamberlain Jeff Baxter. Fastmp: A multi-core simulation methodology.

[3] H Kim, J Lee, NB Lakshminarayana, J Lim, and T Pho. Macsim: Simulator for heterogeneous architecture, 2012.

[4] Geetika Malhotra, Pooja Aggarwal, Abhishek Sagar, and Smruti R Sarangi. Partejas: A parallel simulator for multicores processors.
[5] Avadh Patel, Furat Afram, Shunfei Chen, and Kanad Ghose. Mars: a full system simulator for multicore x86 cpus. In Proceedings of the 48th Design Automation Conference, pages 1050–1055. ACM, 2011.

[6] Jose Renau, Basilio Fraguela, James Tuck, Wei Liu, Milos Prvulovic, Luis Ceze, Smruti Sarangi, Paul Sack, Karin Strauss, and Pablo Montesinos. Sesc simulator, 2005.

[7] André Seznec and Pierre Michaud. A case for (partially) tagged geometric history length branch prediction. Journal of Instruction Level Parallelism, 8:1–23, 2006.