Computer Architecture Reading Group Notes

Date: 1/29/04
Discussion Leader: Kelly
Notes: Francois
Topic: Simulation
Papers:

1. Timothy Sherwood, Erez Perelman, Greg Hamerly and Brad Calder. Automatically Characterizing Large Scale Program Behavior, *In the proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2002)*, October 2002. San Jose, California
2. Roland Wunderlich, Thomas Wenisch, Babak Falsafi, and James Hoe. SMARTS: Accelerating Microarchitecture Simulation through Rigorous Statistical Sampling. *International Symposium on Computer Architecture (ISCA)*, June 2003

Administrative

1. 2 Weeks from now, Mattan on Reliability
2. Set up email list.

Automatically Characterizing Large Scale Program Behavior

Summary:
The problem is simulation time, we don't want to simulate for ever, cut the simulation time and still get relevant data. Look at application and find blocks of application that have similar characteristics. Use clustering to find similar blocks and use simulation points to characterize all similar blocks.

Discussion:

1. Novel approach to speed up simulation
2. Christos: basic assumption is that characteristic of a block will be representative of other data for same instructions.
    a. IPC tracks pretty well with the method, data cache hit rate, not as well.
3. Mattan: one assumption that is not captured is that state between basic blocks doesn't matter
4. Ernesto: whatever the order of the basic blocks, they are in the same phase.
5. Vicky: choice of basic block of 100M I, to negate cold start cache effects? Just the common.
6. Christos, For the L2, there could be dependence between phases.
7. Mattan, Ernesto: Branch predictor could be one that takes a long time to warm up. Switching phases might be worse than a cold start for branch prediction, which could affect the error of this method.
8. Ernesto: Lag on data dependency, but novel matrix approach. What is the relevance of dataset
9. Brian: Paper would be better without section 5 finding simulation points. Doubtful they only find 4 phases in gcc, IPC can obscure a lot.

10. Mattan: made assumption that basic block vectors characterize the program, when you run the whole application, they looked at results which made sense.
11. Christos use of clustering result, 15 is a magic number.
12. Strong point of the paper, they found a metric that without measuring anything, it characterizes a part of the code by similarity with other parts of the program.

SMARTS: Accelerating Microarchitecture Simulation through Rigorous Statistical Sampling.

Summary:

The problem again is slow simulation and we tradeoff simulator complexity and accuracy. This paper tries to show that you can simulate less without losing too much accuracy. Simulate little bits of code and use fast-forwarding, preserving states such as cache and others. How many detailed instructions run, a warm-up, functional warming sets up state of caches branch predictors.

Discussion:

1. Kelly: They warn about sampling longer interval gets the average which in the end is what they want
2. Mattan, Brian, Ernesto: This is classical sampling tradeoff between long samples and more short samples.
3. Kelly: For both papers it would be interesting to see how these results would fare on different architectures.
4. Mattan: Average CPI is a good measure because it relates to run time.
5. Ayo: Doesn't provide any insight into memory bottlenecks.
6. Christos: How do you pick simulation parameters, sorta Ad Hoc, pick some for general case, run, pick W.
    a. Brian: same issue in Markov chains, try till it levels off
7. Mattan: Drawbacks of functional warmup for long simulations, on other architectures.
8. Speed of simulation: 2 times slower than functional simulator, 50x accurate simulator.
9. Christos: Useful for space exploration, still need hierarchy of tools to look into kernels.
10. Kelly: Both papers compare themselves to simulators not actual hardware.
11. Christos: Paper in HPCA 2003 Alaa *Alameldeen* Variability in Architectural Simulations of Multi-Threaded Workloads, used Simmics Ultrasparc varying some parameters like mem latency by small amounts and got very different results, their claim was that we should have error bars based on varying parameters slightly.
12. Mattan: This paper doesn't capture rare events like a synchronization which takes a long time but is rare.
    a. Find rare event in functional simulator, rollback and simulate
13. Kelly: Good for a single threaded application, how do you adapt this to workload:
    a. Mattan: apply same thing to SIMOS instead of SimpleScalar
    b. As long as you don't switch on L2 miss

14. Kelly: Liked the framework and try to validate them, find the error. Though the work would have been better had they tried in different architecture.
15. Beth: Not comparing themselves, just to their own method. Smarts attacks Sherwood paper
16. Nuwan: Smarts paper doesn’t have error rates on L2 cache miss