Poster in the International Conference on Measurement & Modeling of Computer Systems (SIGMETRICS 2005)

TurboSMARTS: Accurate Microarchitecture Simulation Sampling in Minutes

Thomas F. Wenisch Roland E. Wunderlich Babak Falsafi James C. Hoe
Computer Architecture Laboratory (CALCM)
Carnegie Mellon University, Pittsburgh, PA 15213-3890
{twenisch, rolandw, babak, jhoe}@ece.cmu.edu
http://www.ece.cmu.edu/~simflex

ABSTRACT

Recent research proposes accelerating processor microarchitecture simulation through statistical sampling. Prior simulation sampling approaches construct accurate model state for each measurement by continuously warming large microarchitectural structures (e.g., caches and the branch predictor) while emulating the billions of instructions between measurements. This approach, called functional warming, occupies hours of runtime while the detailed simulation that is measured requires mere minutes.

To eliminate the functional warming bottleneck, we propose TurboSMARTS, a simulation framework that stores functionally-warmed state in a library of small, reusable checkpoints. TurboSMARTS enables the creation of the thousands of checkpoints necessary for accurate sampling by storing only the subset of warmed state accessed during simulation of each brief execution window. TurboSMARTS matches the accuracy of prior simulation sampling techniques (i.e., ±3% error with 99.7% confidence), while estimating the performance of an 8-way out-of-order superscalar processor running SPEC CPU2000 in 91 seconds per benchmark, on average, using a 12 GB checkpoint library.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: *Measurement techniques, Modeling techniques*; C.1.1 [**Processor Architectures**]: Single Data Stream Architectures; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

General Terms

Measurement, Performance, Design

Keywords

Checkpointed microarchitecture simulation, simulation sampling

1. INTRODUCTION

Computer architecture research routinely uses detailed cycle-accurate simulation to estimate the performance of microarchitectural innovations. Unfortunately, benchmark applications that are tuned to assess real hardware in minutes can require over a month to execute on today's high performance microarchitecture simulators.

There has been much research that advocates statistical sampling—i.e., measuring only a subset of benchmark execution—as a technique to accelerate microarchitecture simulation. Our prior work, SMARTS[1], determined that a sampling simulator can minimize instructions simulated in detail by collecting a large number (e.g., 10,000) of brief (e.g., 1000-instruction) simulation windows. To produce unbiased estimates, the simulated microarchitecture's state must be accurately warmed prior to each of these sample measurements. We achieved low error by continuously warming large microarchitectural structures (the caches and branch predictor) while emulating the billions of instructions between measurements. This state updating process, referred to as functional warming, dominates simulation turnaround time because the entire benchmark's execution must be emulated, even though only a tiny fraction of the execution is simulated using cycle-accurate microarchitecture timing models (detailed simulation).

We present a simulation framework, TurboSMARTS, that stores a minimal subset of functionally-warmed state in checkpoints. These checkpoints provide the architectural, cache, and branch predictor state required for subsequent experiments, while allowing microarchitectural parameters for other structures (e.g., pipeline configuration) to be varied. This approach amortizes the cost of functional warming over many experiments. We present results demonstrating a TurboSMARTS-based simulator executing the SPEC CPU2000 (SPEC2K) benchmarks to show:

- **Benchmark simulation in minutes.** TurboSMARTS achieves a mean simulation time for an 8-way out-of-order superscalar of just 91 seconds per benchmark—over 250*x* faster than existing simulation sampling approaches—while maintaining the estimated CPI error to ±3% with 99.7% confidence.
- **Small, accurate, and reusable checkpoints.** Although functional warming produces an aggregate of 36 TB of state while sampling SPEC2K, only a tiny fraction of this state is accessed during detailed simulation. The precise state subset cannot be known *a priori* because of the speculative nature of execution in modern microprocessors. However, whereas wrong-path (speculative) instruction *latency* affects scheduling through pipeline resource contention, wrong-path operand *values* rarely affect instruction throughput. By checkpointing only the state required to simulate correct path execution and estimate wrong-path scheduling, we reduce our `gzip`-compressed SPEC2K checkpoint library to 12 GB.

2. SAMPLING WITH CHECKPOINTS

Simulation sampling approaches seek to estimate parameters of benchmark programs on a simulated microarchitecture while simultaneously minimizing the error in the estimated result and the

SIGMETRICS'05, June 6-10, 2005, Banff, Alberta, Canada.
ACM 1-59593-022-1/05/0006.

1. R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe. SMARTS: Accelerating microarchitecture simulation via rigorous statistical sampling. In *Proc. Int'l Symp. Computer Architecture*, June 2003.

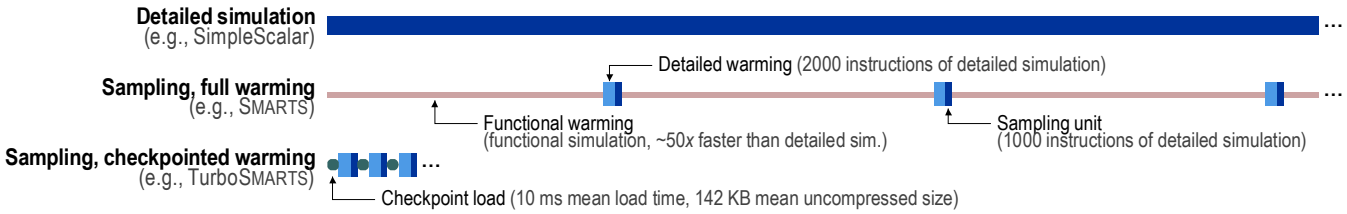

Figure 1. CPU microarchitecture simulation sampling techniques.
Sampling of SPEC2K requires a mean sample size of 8000 sampling units to estimate CPI ±3% with 99.7% confidence. TurboSMARTS mean execution time is 91 seconds per benchmark, ~5,000*x* faster than SimpleScalar, and ~250*x* faster than SMARTS.

runtime of the sampled simulation. There are two components of any error present in sampling estimates: sampling error and non-sampling error. Sampling error results when, by chance, poor measurement locations are chosen, while non-sampling error is caused by bias in measurement, for example a cold-start bias from assuming empty caches at the start of each measurement.

Our recent study of optimal sample design for microarchitecture simulation, SMARTS, demonstrated that sampling error can be controlled with minimal simulation cost by measuring a large number (e.g., 10,000) of brief (e.g., 1000-instruction) execution windows (sampling units). SMARTS addresses non-sampling error through a two-tier warming strategy, depicted in Figure 1. Large microarchitectural structures are warmed continuously between each window (functional warming), while the remainder of the microarchitecture is warmed in a bounded period of cycle-accurate simulation without metric collection (detailed warming).

Sampled simulations that apply a SMARTS-like warming strategy spend more than 99% of execution time in functional warming. Over a series of simulations, functional warming repetitively generates the same state. In TurboSMARTS, we amortize the cost of functional warming over many simulations by storing this state in reusable checkpoints.

Functional warming generates an aggregate of 36 TB of state for the hundreds of thousands of sampling units required for our complete benchmark suite. The key innovation of TurboSMARTS lies in identifying the minimal subset of this state required for accurate simulation of each sampling unit. We can precisely identify the sequence of instructions which commit during detailed warmup and measurement—and therefore the state these instructions access—when creating checkpoints. However, the performance of modern microarchitectures is also affected by interactions between the committed instruction stream and instructions which are speculatively executed and then discarded. Unlike the commit instruction sequence, the wrong-path instruction sequence depends on microarchitecture configuration.

Wrong-path instructions interact with the commit stream through resource contention and in the cache tag arrays. In the vast majority of cases, we can use branch predictor outcomes to identify the wrong-path instruction sequence, and cache tag arrays to identify wrong-path load latency. This information is sufficient to identify contention and effects arising from speculative execution, without the need for the values accessed by wrong-path loads. By restricting our checkpoints to store only the values accessed by correct-path instructions in the sampling unit, plus the cache tags and branch predictor, we reduce total checkpoint size to 12 GB (after `gzip` compression of between 2:1 and 5:1).

3. RESULTS

We evaluate TurboSMARTS with a sampling simulator based on the widely-adopted SimpleScalar 3.0 `sim-outorder` microarchitecture simulator. This summary presents results of estimating the cycles-per-instruction (CPI) of SPEC2K benchmarks executing on an 8-way out-of-order superscalar processor with a 1 MB L2 cache, and other parameters chosen to represent near-future microarchitectures. Our samples are designed to achieve precisely 99.7% confidence of ±3% error in results. We compare TurboSMARTS to non-sampled runs of the complete benchmark with SimpleScalar, and sampling with full warming using SMARTS. We report runtimes for systems with 2.8 GHz Intel Xeon processors. Further details of our experimental setup and additional results are presented at `http://www.ece.cmu.edu/~simflex`.

Table 1 presents a summary of the characteristics of each of the warming approaches we evaluated. TurboSMARTS eliminates the functional warming bottleneck in previously proposed simulation sampling approaches, reducing average simulation time for SPEC2K benchmarks from 7 hours to just 1.5 minutes, with no increase in no-sampling error. TurboSMARTS simulations often complete faster than native execution of benchmarks on our host platform, which typically requires several minutes per benchmark. TurboSMARTS' checkpoints require a fixed maximum cache and branch predictor size and associativity for subsequent simulations.

TurboSMARTS reduces microarchitecture simulation time to the limit imposed by detailed simulation—mere minutes—by applying checkpointing to simulation sampling[1].

1. This work was funded in part by grants and equipment from IBM and Intel corporations, the DARPA PAC/C contract F336150214004-AF, and an NSF CAREER award.

Table 1. TurboSMARTS result summary.

	Detailed simulation (SimpleScalar)	Full warming (SMARTS)	Checkpointed warming (TurboSMARTS)
Mean (maximum) CPI non-sampling error	None	0.6% (1.6%)	0.6% (1.6%)
Mean benchmark runtime	5.5 days	7.0 hours	91 seconds
SPEC2K checkpoint library size	N/A	N/A	12 GB (for 1 MB max L2)
Fixed microarchitecture parameters	None	None	Max cache, branch predictor size