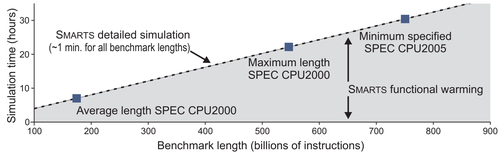
TurboSMARTS: Accurate Microarchitecture Simulation Sampling in Minutes

Thomas F. Wenisch, Roland E. Wunderlich, Babak Falsafi, and James C. Hoe – Computer Architecture Laboratory at Carnegie Mellon (CALCM), Pittsburgh, PA, USA

1. Introduction

TurboSMARTS samples microprocessor simulation using *live-points* that are small, fast loading, reusable, and accurate checkpoints.

Conventional simulation sampling reduces runtime, but does not scale to tomorrow's trillion-instruction benchmarks such as SPEC CPU2005.

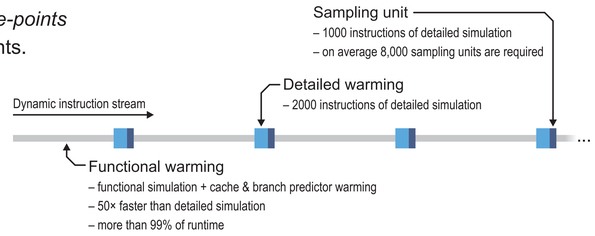


Breakdown of SMARTS execution time for SPEC CPU benchmarks

We replace functional warming with live-points to reduce simulation runtime drastically while maintaining high sampling accuracy. Although modern computer architecture simulators frequently provide checkpoint creation and loading capabilities, current checkpoint implementations:

(1) do not provide complete microarchitectural model state, and
(2) cannot scale to the required checkpoint library size because of multi-terabyte storage requirements.

Our live-point implementation demonstrates:
- *Accelerated simulation with practical storage*
- *Parallel simulation and online results*
- *Reusable live-point libraries*

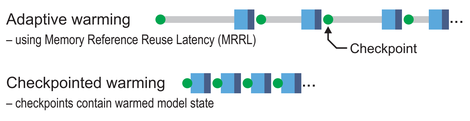


Conventional simulation sampling: The SMARTS framework can achieve tight confidence intervals e.g. 99.7% confidence ±3% for CPI estimates.

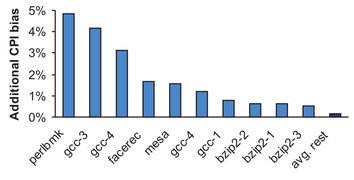
2. Checkpointed warming

Checkpointed warming stores warmed branch predictor and cache state in a reusable format, and eliminates the problem of determining how much warming is needed.

We choose checkpointed warming over an adaptive warming technique to minimize warming bias. Even the best adaptive warming approaches cannot precisely predict how much warming is required for each sampling unit.



Warming methods for simulation sampling: All methods use the same sample design and confidence intervals, only bias differs.

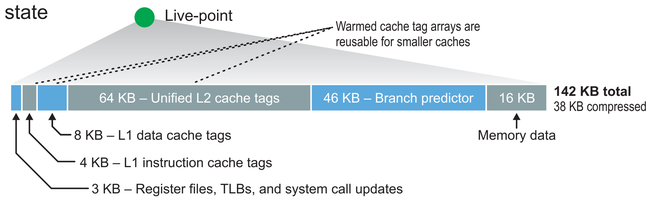


Adaptive warming bias: Additional error is introduced by adaptive warming using MRRL vs. full warming (SMARTS).

3. Live-state

We reduce the size of conventional checkpoints from hundreds of megabytes to hundreds of kilobytes. *Live-state* stores only the subset of state necessary for limited execution windows of thousands of instructions.

The minimal state subset can be known *a priori* only for the commit instruction stream, and is not known for wrong-path (speculative) instructions. However, whereas wrong-path instruction latency affects scheduling through pipeline resource contention, wrong-path operand values rarely affect instruction throughput. We exploit this observation by storing only the state required for correct path execution and approximate wrong-path scheduling.



Uncompressed live-point contents: Live-state stores only touched memory data and complete cache tag arrays in each live-point. For comparison, a conventional checkpoint is 105 MB.

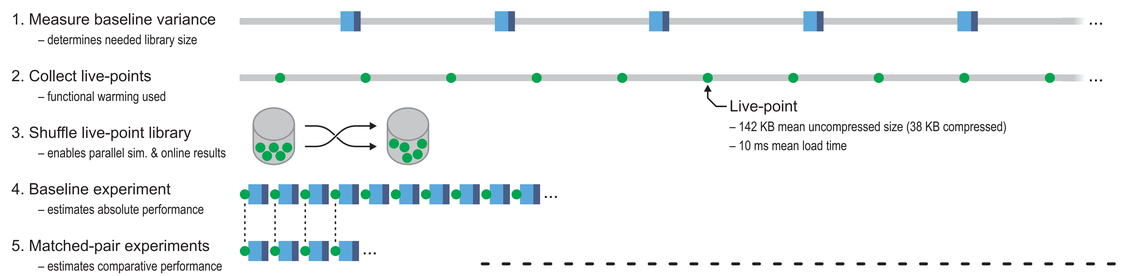
7. Results summary

	SimpleScalar Complete simulation	SMARTS Full warming	MRRL Adaptive warming	TurboSMARTS using Live-points
Average (worst) CPI bias	None	0.6% (1.6%)	1.1% (5.4%)	0.6% (1.6%)
Average benchmark runtime	5.5 days	7.0 hours	1.5 hours	**91 seconds**
Scaling behavior	$O(B \times DS)$	$O(B)$	$O(1)$	$O(C)$
Parallel sim. and online results	N/A	N/A	No	Yes
SPEC CPU2000 ckpt. library size	N/A	N/A	30 GB	12 GB (1 MB L2)
Scaling behavior	N/A	N/A	$O(1)$	$O(C)$
Fixed microarchitecture parameters	None	None	None	Max cache, TLB, branch predictors

B = benchmark length, *C* = maximum cache size, *DS* = detailed simulation speed

4. Live-point experiment procedure

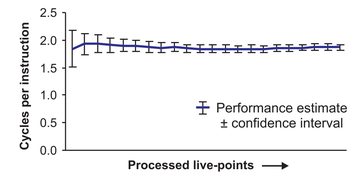
Each live-point uses checkpointed warming and live-state. Our experiment procedure enables parallel simulation, online results, and matched-pair comparisons.



5. Parallel simulation & online results

We construct independent live-points that can be processed in parallel and in an arbitrary order. By randomizing the processing order, we can report unbiased results and their statistical confidence intervals continuously during microarchitecture simulation.

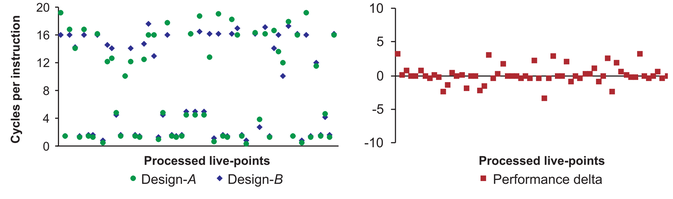
In contrast, simulators that use functional warming cannot report results until simulation is complete.



Online results: As live-points are processed, results converge toward their final values and confidence improves.

6. Matched-pair comparison

Researchers are often more interested in the relative performance of two designs than absolute performance. Matched pair comparison exploits the phenomenon that the change in performance from design-*A* to design-*B* tends to vary less than the absolute change in performance.



Matched-pair experiments: The lower variability of performance deltas reduces sample size by 3.5 to 150×.

www.ece.cmu.edu/~simflex

Visit our web site to download the simulator.

Carnegie Mellon