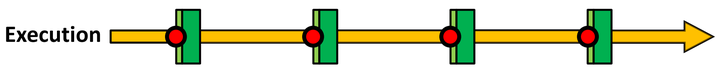
PROTOFLEX:
Complexity-Effective FPGA-Accelerated Instrumentation

Michael K. Papamichael, Eric S. Chung, James C. Hoe, Babak Falsafi, and Ken Mai
papamix@cs.cmu.edu, {echung, enurvita, jhoe, babak, kenmai}@ece.cmu.edu
http://www.ece.cmu.edu/~simflex/protoflex.html

Computer Architecture Lab at Carnegie Mellon

Problem: Functional warming is slow

- **Performance Simulation via Simulation Sampling**
  - perf. measurements by sampling small segments of execution

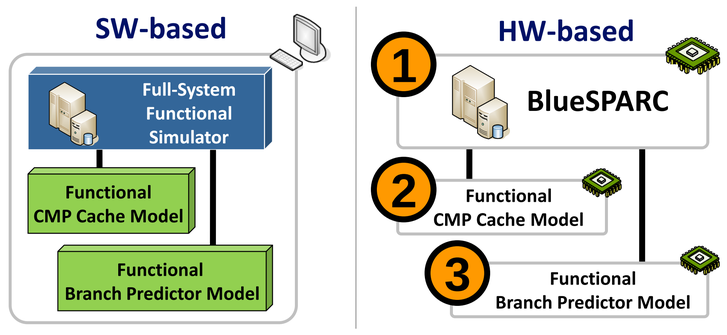


Functional Warming (functional simulation) **Checkpoints** (i.e. system state snapshots)	Long & NOT Parallelizable
Detailed Warm-up (cycle-accurate simulation) **Measurement** (cycle-accurate simulation)	Short & Parallelizable

- **Speed of cycle-accurate simulator inconsequential**
  - Sampling regions are small & can be simulated in parallel
- **Functional Warming is the Real Bottleneck**
  - Checkpointing is a sequential process that takes a long time

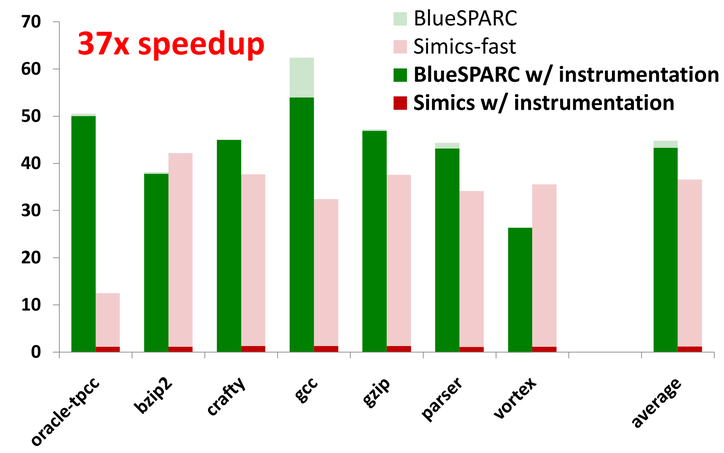
Solution: Make it fast using FPGAs

- **Functional Warming requires**
  - Full-system functional simulator (e.g. Simics)
  - Instrumentation (e.g. functional cache model)



Instrumented HW Simulator → Fast Functional Warming

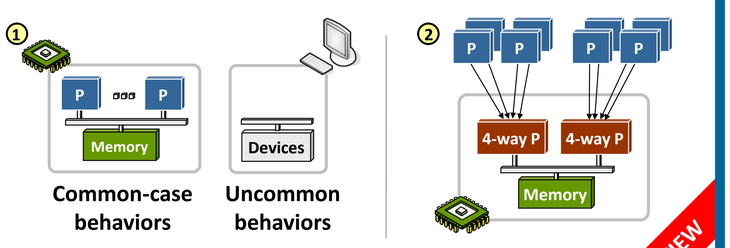
Performance Improvement



HW-based instrumentation has minimal overhead

1 BlueSPARC Simulator

- **Full-system HW-based Functional Simulator**
  - Models 16-cpu UltraSPARC III server
  - Can boot OS, run commercial apps
- **Virtualization Techniques**
  - ① Hybrid Full-System Simulation
  - ② Multiprocessor Host Interleaving

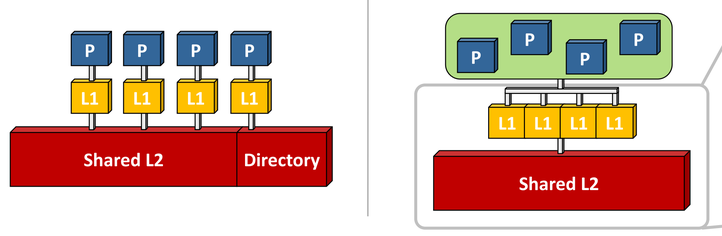


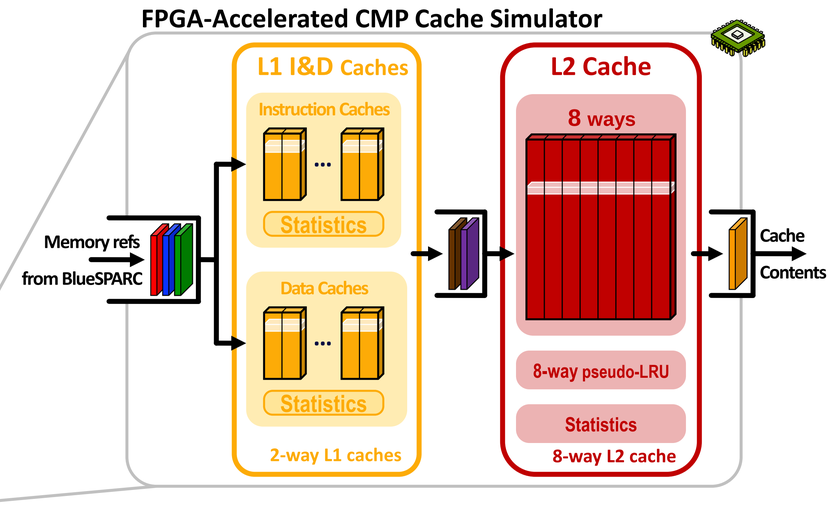
2 FPGA-Accelereated Functional Chip-Multiprocessor Cache Simulation

- **Piranha-like CMP Cache Hierarchy**
  - Private L1 I&D Caches
  - Single Shared L2 Cache (Victim Cache)
  - L1 coherence maintained through directory in L2

Target Cache Model
- Multiple concurrent memory refs
- Directory for coherence

Virtualized Cache Model
- Memory refs serialized
- Parallel L1 accesses for coherence

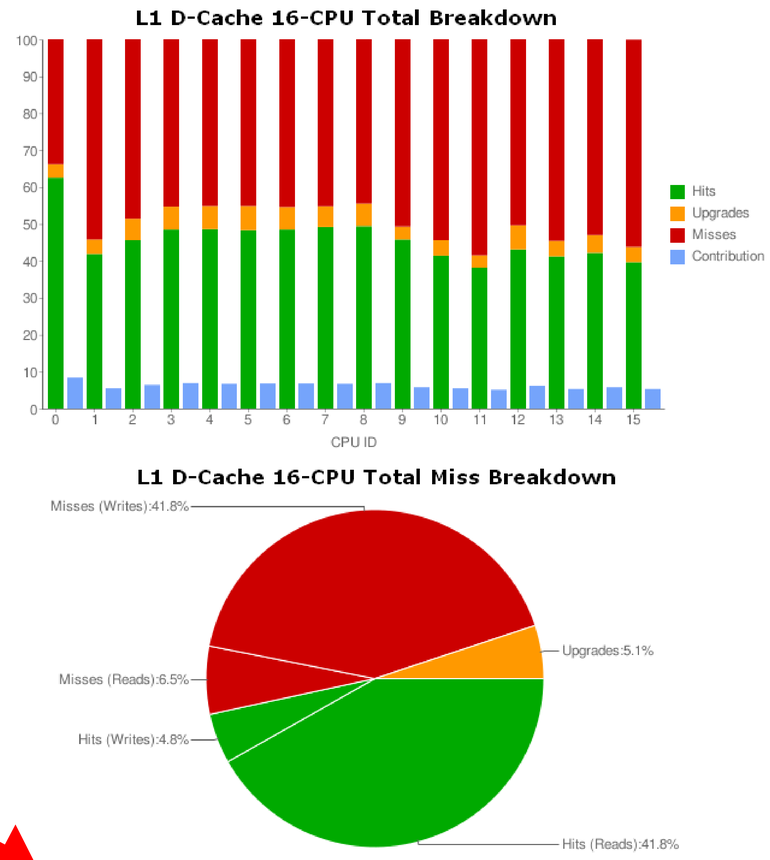




Prototype Configuration & Implementation Details

Number of CPUs	16
L1 Caches	16 private 2-way split I&D (64B blocks) (2 stage pipeline, 1 cycle/ref)
L2 Cache	8-way single shared victim cache (64B blocks) (4 stage pipeline, 2 cycles/ref)
Clock frequency	100MHz
Resources	7483 LUTs (11%), 134 BRAMs (40%) (64KB L1s, 4MB L2) 7277 LUTs (11%), 292 BRAMS (89%) (128KB L1s, 16 MB L2)
EDA tools	Xilinx EDK 9.2i, XST 9.2i
Statistics	2500 lines of fully-parametereized Verilog, 8 modules

Web-based Real-time Viewing of Statistics



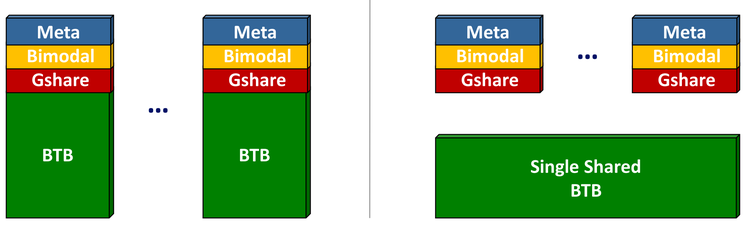
Check out our DEMO!

3 FPGA-Accelerated Branch Predictor Simulation

- **Typical 2-level Branch Predictor**
  - Meta predictor selects Bimodal or Gshare predictor
  - 8-way Branch Target Buffer
- **16 BTBs (one per cpu) too large for BEE2 FPGA**

Target BP Model
- One BTB per CPU

Virtualized BP Model
- Single Shared BTB for all CPUs

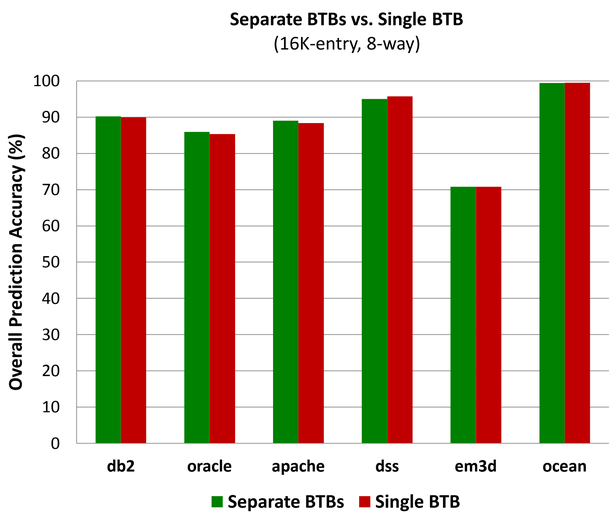


Prototype Configuration & Implementation Details

Number of CPUs	16
Predictors	Bimodal: 32K-entry Gshare: 8K-entry Meta: 8K-entry
BTB	Single shared 16K-entry 8-way BTB
Clock Frequency	100MHz
Resources (Virtex-II Pro 70)	3938 LUTs (5%), 193 BRAMs (40%)
EDA tools	Xilinx EDK 10.1i, XST 10.1i
Statistics	700 lines of Bluespec

Single vs. Multiple BTBs

- **OK to use single BTB? Generally no, but OK for**
  - functional warming of homogeneous workloads



Single BTB achieves same accuracy (within 1%)

Future Work

Other Instrumentation Applications

- **Software Monitoring/Analysis**
  - e.g. debugging, performance tuning, instruction set profiling
- **Rapid Exploration of new Architectures**
  - Simple functional models for first-order perf. results
  - Detailed cycle-accurate models for high-fidelity simulation
- **Functional warming of µArch structures**
  - Acceleration of sampling-based timing simulation
- **SW Developer/Educational Tool**
  - E.g. Real-time viewing of system state and statistics

Future Directions

- **Scale to Larger Number of CPUs**
- **Augment Simulation Models with Timing Info**

Acknowledgements

We would like to thank our colleagues in the RAMP and TRUSS projects. Our work in this area has been supported in part by NSF, IBM, Intel and Xilinx.