



FACS

FPGA-Accelerated Multiprocessor Cache Simulator

Michael Papamichael [papamix@cs.cmu.edu], Wei Yu [wy@andrew.cmu.edu], Yongjun Jeon [yongjunj@andrew.cmu.edu]
<http://www.cs.cmu.edu/~mpapamic/projects/facs.html>

Summary

Current architectural-level full-system software-based simulators (e.g. Virtutech Simics) are limited in throughput, especially when simulating multiprocessor systems. The slowdown becomes even higher when attaching additional modules to the simulator, such as cache simulators.

FACS (FPGA-Accelerated Cache Simulator) is a fully parameterizable hardware functional **piranha-based** multiprocessor cache simulator that precisely replicates the behavior of the existing software-based **TraceCMPFlex** cache model. Our results show that FACS is over 200x faster than TraceCMPFlex.

FACS in a Nutshell

- **Hardware Multiprocessor Cache Model**
 - Piranha-based 2-level Coherent Cache Hierarchy
 - 6-stage Pipeline Implementation
 - Fully Parameterizable Design
 - number of cores, L1/L2 block size, L1/L2 cache size, L2 associativity
- **Runs on Real Hardware (FPGA)**
 - Tested on BEE2 and XUP boards @ 100 MHz
 - High Throughput: 100 million references/sec
 - PowerPC Interface
 - feed references, read out cache contents, read/write statistics memory
 - Traces reside on DRAM or Compact Flash Cards
- **Precise Replication of TraceCMPFlex SW model**
 - Over 200x Speedup!

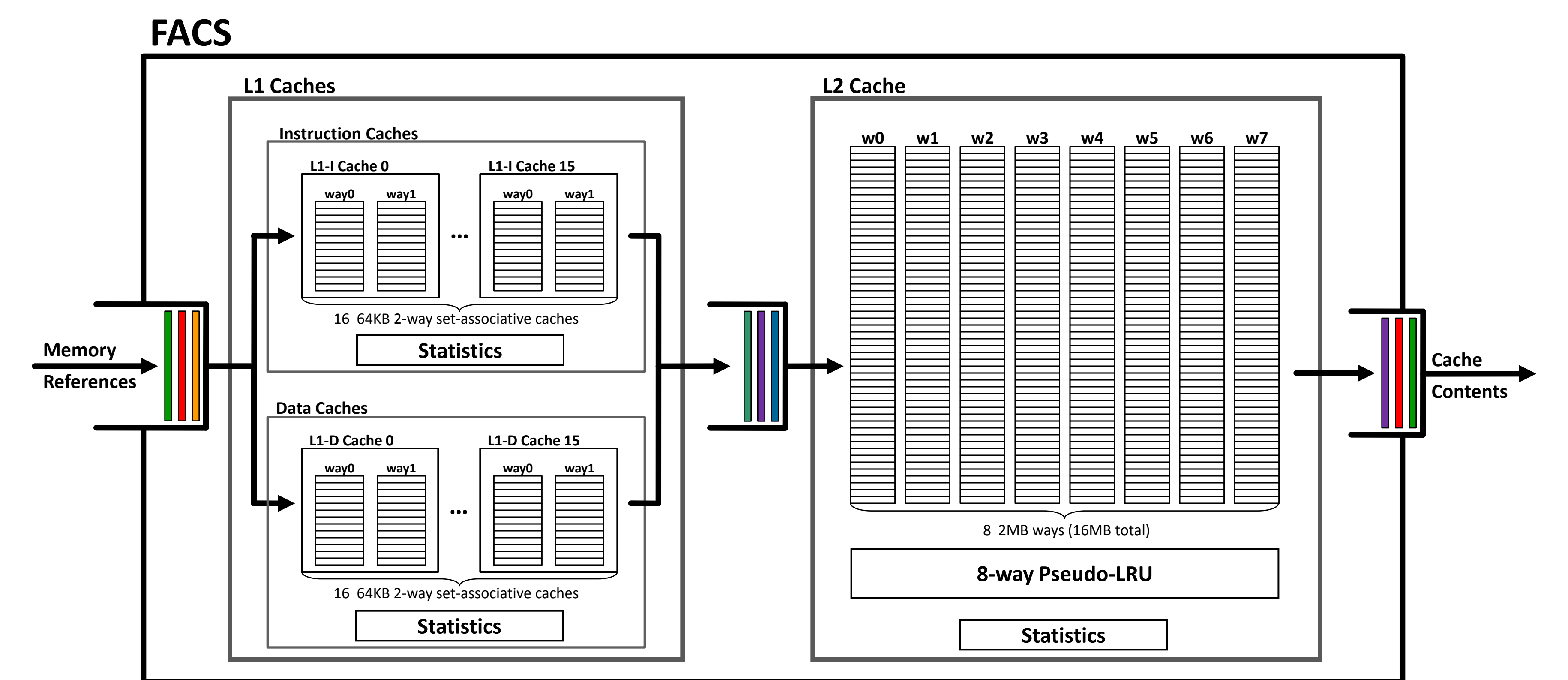
Hardware Implementation

- **2500 lines of Verilog code**
- **Functional Model**
 - Only tags and status bits stored and updated
- **L1 Caches**
 - Implemented as 2-stage pipeline
 - All 64 L1 caches simultaneously accessed
- **L2 Cache**
 - Victim cache (only inserts evicted blocks from L1)
 - Each reference is processed in 4 cycles

Software

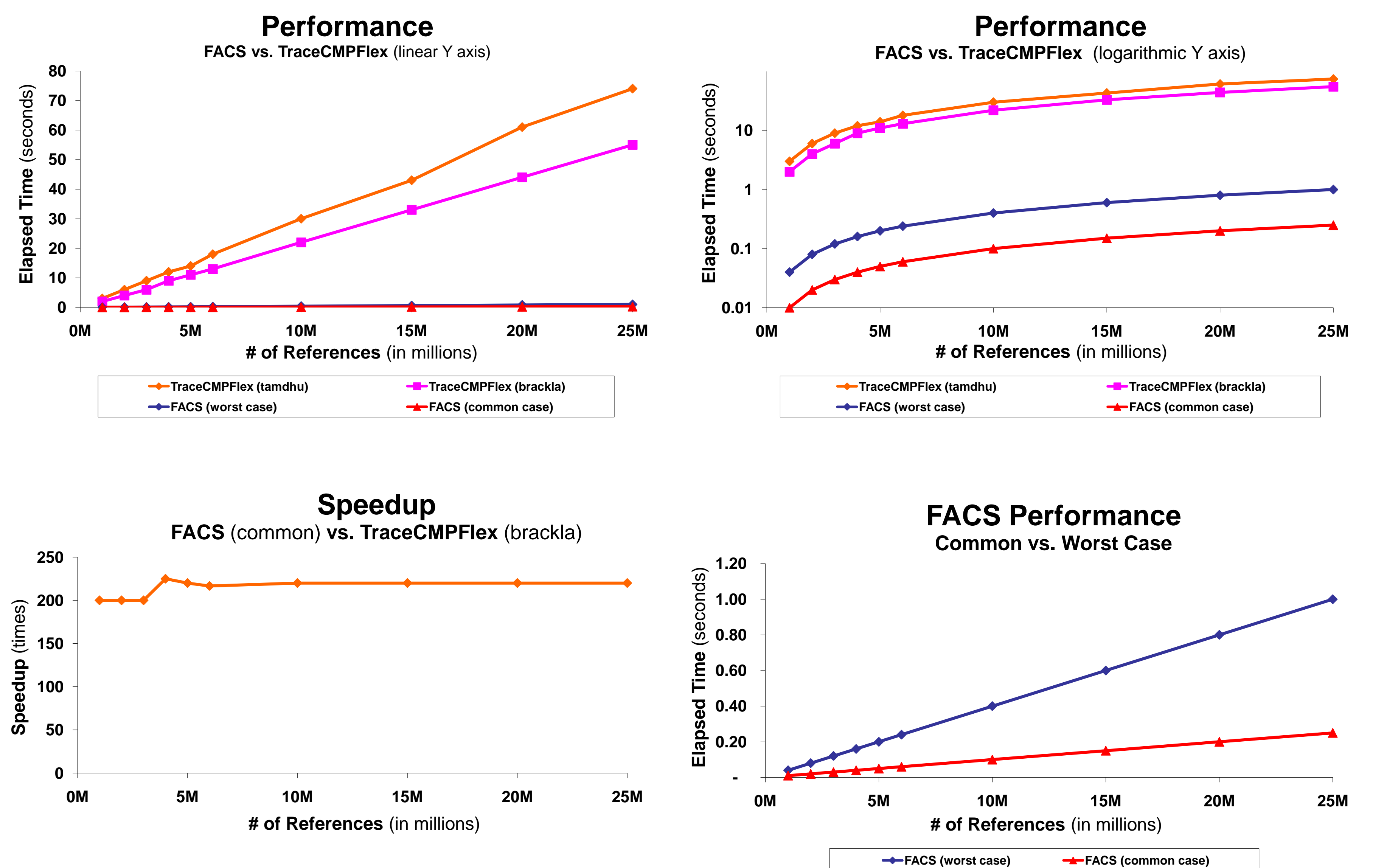
- **Modified Flexus Components**
 - **FastCache**: functional L1 cache model
 - **FastCMPCache**: functional L2 cache model
 - **DecoupledFeeder**: collects/feeds traces
 - **Stat-Manager**: tool for viewing statistics
- **Additional Tools Developed to**
 - Convert traces to between various formats
 - Compare FACS and TraceCMPFlex statistics

Architecture



Performance Results

- **Collected large memory reference traces from Apache workloads and fed to**
 - **brackla**: Intel Xeon 5130 @ 2GHz (4MB L2) with 8GB RAM (4 cores in total)
 - **tamdhu**: Intel Xeon MP @ 2.8 GHz (512KB L2) with 3GB RAM (2 cores in total)



Check out our Demo!

- **FACS Demo Configuration**
 - Runs @ 100 MHz
 - Supports 16 Cores
 - 64 Byte Block Size (for both L1 and L2)
 - 128 KB 2-way set-associative split I&D L1 cache per core
 - 4 MB 8-way set-associative shared L2 cache
- **Menu-driven terminal-based interface**
- **Real-time viewing of statistics**

Acknowledgments

We would like to thank Eric Chung for his support on the BEE2 development board and Nikos Hardavellas for helping us out with the TraceCMPFlex software cache model. We would also like to thank James Hoe for providing us with previously developed pieces of Verilog code that greatly reduced the required implementation time.