FIST: A Fast, Lightweight, FPGA-Friendly Packet Latency Estimator for NoC Modeling in Full-System Simulations

Michael K. Papamichael, James C. Hoe, Onur Mutlu
Carnegie Mellon University, Pittsburgh, PA USA
papamix@cs.cmu.edu, jhoe@ece.cmu.edu, onur@cmu.edu

Computer Architecture Lab at Carnegie Mellon

Simulation in Computer Architecture
- Slow for large-scale multiprocessor studies
- Full-system fidelity + long benchmarks

How can we make it faster?
- Speed, accuracy, flexibility trade-off
  - Full-system simulators sacrifice accuracy for speed and flexibility
- Accelerate simulation with FPGAs
  - Can simulate up to millions of gates
  - Orders of magnitude simulation speedup

The FIST Project
- Explores fast NoC models for full-system simulations
- FPGA-friendly, but avoid direct implementation
- Low error, many topologies, >10M packets/sec
- Simpler requirements of full-system simulation
- Estimate packet latencies, capture high-order effects

FIST Approach
- View NoC as set of routers/links
- Abstract router into black-box
- Represent by load-delay curves
- Specific to each router configuration and traffic pattern

Putting FIST Into Context
- Detailed network models
  - Cycle-accurate network simulators (e.g. BookSim)
  - Analytical network models
  - Typically study networks under synthetic traffic patterns

- Network models within full-system simulators
  - Model network within a broader simulated system
  - Assign delay to each packet traversing the network
  - Traffic generated by real workloads

FIST-based Network Models
- Offline FIST
  - Detailed network simulator generates curves offline
  - Can use synthetic or actual workload traffic
  - Load curves into FIST and run experiment

- Online FIST (tolerates dynamic changes in network behavior)
  - Initialization of curves same as offline
  - Periodically run detailed network simulator on the side
  - Compare accuracy and, if necessary, update curves

FIST Applicability
- "FIST-Friendly" Networks
  - Exhibit stable, predictable behavior as load fluctuates
  - Actual traffic similar to training traffic

- FIST Limitations
  - Depends on fidelity, representativeness of training models
  - Higher loads and large buffers can limit FIST’s accuracy
  - High network load → increased packet latency variance
  - Large buffers → increased range of observed packet latencies
  - Cannot capture fine-grain packet interactions
  - Cannot replace cycle-accurate detailed network models

- NoCs are “FIST-Friendly”
  - Employ simple routing algorithms
  - Operate at low loads
  - Small buffers

Evaluation
- Methodology
  - Software implementation of FIST (written in C++)
  - Examined online and offline FIST models
    - Replaced cycle-accurate NoC model in tiled CMP simulator
  - Network and system configuration
    - 4x4, 8x8, 16x16 wormhole routed mesh
    - Each network node host cores: 1 L1 and a slice of L2
    - Multiprogrammed and multithreaded workloads
    - 26 SPEC CPU2006 benchmarks of varying network intensity
    - 8 SPLASH-2 and 2 PARSEC workloads
  - Traffic generated by cache misses
    - Consists of control, data and coherence packets
  - Offline and Online FIST models with two curves per router
  - Curves represent injection and traversal latency at each router
  - Initialization training using uniform random synthetic traffic
  - Please see paper for more details!

- Latency and IPC accuracy for FIST-based models
  - Latency Error ≤ 8%
  - 8x8 mesh using FIST offline model
  - Both Latency and IPC Error below 3%
  - 8x8 mesh using FIST online model

Comparison against simple hop-based model
- Speedup for 16x16 mesh using offline FIST: 43x
- Speedup for 16x16 mesh using online FIST: 18x

FPGA Implementation of FIST
- Hardware Implementation (written in Bluespec)
  - Precisely replicates software-based FIST
  - 3-4 orders of magnitude speedup (offline FIST)

Related Work and Conclusions
- Related Work
  - Abstract network modeling
    - Performance vs. accuracy trade-off studies (Burger 95)
    - Load-delay curve representation of network (Lugones 09)
  - FPGAs for network modeling
    - Cycle-accurate fidelity at the cost of limited scalability
    - Time-multiplexing can help with scalability (Wang 10)
    - But still suffer from high implementation complexity

Conclusions
- Full-system simulators can tolerate small inaccuracies
- FIST can provide fast SW- or HW-based NoC models
  - SW model provides 18x-43x average speedup w/ <2% error
  - HW model can scale to 100s routers with >1000x speedup
- NoCs are “FIST-friendly”
  - But not all networks good candidates for FIST modeling

Future Directions
- FPGA-friendly NoC models at multiple levels of fidelity
- Configurable generation of hardware NoC models

Acknowledgements
Funding for this work has been provided by NSF CCF-0811702. We thank Xilinx for their FPGA and tool donations. We thank Bluespec for their tool donations and support.