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

Michael K. Papamichael, James C. Hoe, Onur Mutlu
papamix@cs.cmu.edu, jhoe@ece.cmu.edu, onur@cmu.edu

Computer Architecture Lab at Carnegie Mellon

Our work was supported by NSF. We thank Xilinx and Bluespec for their FPGA and tool donations.
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

![Diagram of network on chip models]

FPGA area requirements for state-of-the-art mesh NoC*

*NoC RTL from http://nocs.stanford.edu/router.html
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
FIST Approach

- Treat each hop as a set of load-delay curves
  - Trade-off between model complexity and fidelity

- Keep track of load at each node
  - To track router load monitor traffic over window of time

[Diagram showing network nodes and load-delay curves]
FIST in Action

- Route packet from source to destination
  - Determine routers that will be traversed

- Sum up the delays for each traversed router
  - Index load-delay curves using current load at each router
Outline

- Introduction to FIST
- FIST-based Network Models
- Evaluation
- Related Work & Conclusions
Outline

- Introduction to FIST
- FIST-based Network Models
- Evaluation
- Related Work & Conclusions
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*
Offline and Online FIST

**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

Provide feedback and receive updated curves
Online Training in Action

- Example with no initial training

![Graph showing latency and elapsed cycles before and after training.](image)

<table>
<thead>
<tr>
<th>Elapsed cycles (in 1000s)</th>
<th>Actual Latency</th>
<th>Estimated Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2082</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2214</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2346</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2412</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2478</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CALCM  Computer Architecture Lab at  Carnegie Mellon
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 $\rightarrow$ increased packet latency variance
  - Large buffers $\rightarrow$ increased range of observed packet latencies
- Cannot capture fine-grain packet interactions
- Cannot replace cycle-accurate detailed network models

FIST only as good as its training data
Applying FIST to NoCs

NoCs affected by on-chip limitations and scarce resources

- Employ simple routing algorithms
  - Usually simple deterministic routing

- Operate at low loads
  - NoCs usually over-provisioned to handle worst-case
  - Have been observed to operate at low injection rates

- Small buffers
  - On-chip abundance of wires reduces buffering requirements
  - Amount of buffering in NoCs is limited or even eliminated

NoCs are “FIST-Friendly”
Outline

- Introduction to FIST
- FIST-based Network Models
- Evaluation
- Related Work & Conclusions
FIST Implementations

- **Software Implementation of FIST** (written in C++)
  - Implements online and offline FIST models
- **Hardware Implementation** (written in Bluespec)
  - Precisely replicates software-based FIST
  - Block diagram of architecture
Peeking Under The Hood

- **Tracking Latency**

  - **Store-and-forward**

    ![Diagram of Store-and-forward]

    | Router | Packet Latency |
    |--------|---------------|
    | R0     | 9             |
    | R1     | 14            |
    | R2     | 7             |

    Packet Latency = 30
    - **R0** Latency = 9
    - **R1** Latency = 14
    - **R2** Latency = 7

  - **Wormhole-routed**

    ![Diagram of Wormhole-routed]

    | Router | Packet Latency |
    |--------|---------------|
    | R0     | 9             |
    | R1     | 14            |
    | R2     | 7             |

    Packet Latency = 30
    - **R0** Latency = ?
    - **R1** Latency = ?
    - **R2** Latency = ?

  Use separate injection and traversal latency curves per router

- **Similar issues arise for load tracking & dynamic training**
Methodology

- 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 hosts core+coherent 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
  - Initial training using uniform random synthetic traffic

- Please see paper for more details!
Accuracy Results (offline)

- **8x8 mesh using FIST offline model**
  - Average Latency and Aggregate IPC Error

<table>
<thead>
<tr>
<th>Latency/IPC Error (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Latency Error</td>
</tr>
<tr>
<td>+ IPC Error</td>
</tr>
</tbody>
</table>

**IPC Error < 4%**

**Latency Error < 8%**

MT (SPL/PAR)  
MP (High)  
MP (Med)  
MP (Low)
Accuracy Results (online)

- 8x8 mesh using FIST online model
  - Average Latency and Aggregate IPC Error

Both Latency and IPC Error below 3%
What about a very simple model?

- **8x8 mesh using hop-based model**
  - How does simple network model affect high-order results?

Latency/IPC Error (in %)

- Latency Error
- IPC Error

FIST models always within this range

Very high error for both latency and IPC!
Performance Results

- Qualitative comparison

![Diagram]

- SW-based speedup results for 16x16 mesh
  - Offline FIST: 43x
  - Online FIST: 18x

- HW-based speedup (offline): ~3-4 orders of magnitude
Hardware Implementation Results

- FPGA resource usage & clock frequency
  - Different mesh configurations
  - Xilinx Virtex-5 LX155T FPGA

<table>
<thead>
<tr>
<th>Size</th>
<th>FIST Model</th>
<th>Direct Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPGA Area</td>
<td>Freq.</td>
</tr>
<tr>
<td>4x4</td>
<td>4%</td>
<td>380 MHz</td>
</tr>
<tr>
<td>8x8</td>
<td>15%</td>
<td>263 MHz</td>
</tr>
<tr>
<td>12x12</td>
<td>34%</td>
<td>250 MHz</td>
</tr>
<tr>
<td>16x16</td>
<td>60%</td>
<td>214 MHz</td>
</tr>
<tr>
<td>20x20</td>
<td>94%</td>
<td>200 MHz</td>
</tr>
</tbody>
</table>

FIST can scale to large NoCs with many routers

**Will not fit**
Outline

- Introduction to FIST
- FIST-based Network Models
- Evaluation
- Related Work & Conclusions
Related Work

- Vast body of work on network modeling
  - Analytical models, hardware prototyping, etc.

- 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 & Future Directions

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 within a CMP 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
Thanks!

Questions?