Energy Efficient Network-on-Chip with Runtime Optimization Selection for MPSoCs

Masaaki Kondo

Graduate School of Information Science and Technology,
The University of Tokyo

Background

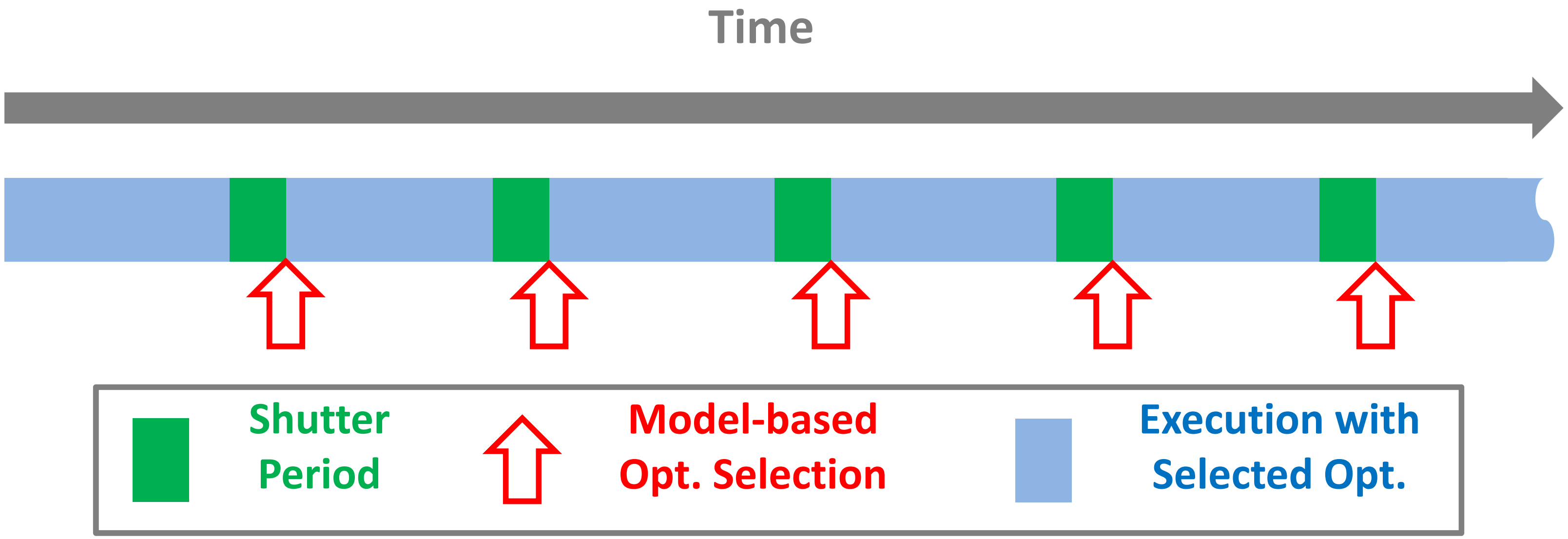
- NoCs are becoming the communication backbones for Manycore Processor SoCs
- Performance and power is affected by the efficiency of NoCs
  - Cache and memory access latencies are NoC-dependent
  - Up to 30% of processor power can be drawn by NoCs
- Optimization techniques for energy efficient NoCs
  - Many optimization techniques have been proposed so far
  - Performance optimization techniques comes with power overhead while power optimization techniques comes with performance penalties
  - How do these optimization techniques affect their energy efficiency?
  - How to utilize these optimization techniques to achieve the best energy efficiency?

Challenges and Strategies

- To find best combination of NoC optimization techniques for executing applications
  - Need to select suitable optimization combinations dynamically
  - Huge number of candidates of combinations
  - Time overhead for simulation or profiling is not acceptable
- Our strategies
  - Runtime framework to adaptively control NoC optimization
  - Implement possible optimizations and make their functionality controllable
  - Create performance and energy models to estimate the impact of optimization techniques on them
  - Based on the estimated performance and energy, apply the best mix of optimization techniques

Proposed Runtime Framework

- Epoch-based control
  - Runtime is divided into shutter periods and execution epochs
  - Switches all optimization techniques on in the shutter period to collect performance stats
  - Using these stats and the performance and energy models, makes the best throttling decisions of optimization techniques
  - Applies them for the succeeding execution epoch



Time

Shutter Period

Model-based Opt. Selection

Execution with Selected Opt.

The NoC Optimization Techniques

- We focus on three techniques as examples
  - Applicable for wide variety of techniques if modeled
- *Power Gating (PG)*
  - Turns each router off if it is idle for a specific time duration
  - Saves static power with dynamic power and latency overhead
- *Prediction Router (PR)*
  - Bypasses the router’s datapath if output port of a packet is predicted
  - Reduces latency with power overhead of predictors
- *Traffic Compression (TR)*
  - Reduces packet size (or number of flits) if the compression succeeds
  - Has positive effect on performance and dynamic power
  - Compression circuitry consumes power and takes time

Performance Models at a Glance

- Performance model
  - Inputs: num. of hops per flit, the total number of packets, and the average number of flits per packet
  - The base network latency model:

$$\overline{L_{Net}} = 2L_{NI} + L_{Route} \times \overline{H} + L_{Link} \times (\overline{H} + 1) + \overline{L_{Queue}}$$

- Energy model
  - Inputs: num. of router and link accesses
  - Parameters: static power, clock power, energy per access for the links and routers
  - The base network energy model:

$$\overline{E_{Net}} = E_{D_{Router}} \times \overline{H} + E_{D_{Link}} \times (\overline{H} + 1) + \frac{P_{D_{Clk}} \times T(\overline{L_{Net}})}{N_{Flit}} + \frac{P_{S_{Net}} \times T(\overline{L_{Net}})}{N_{Flit}}$$

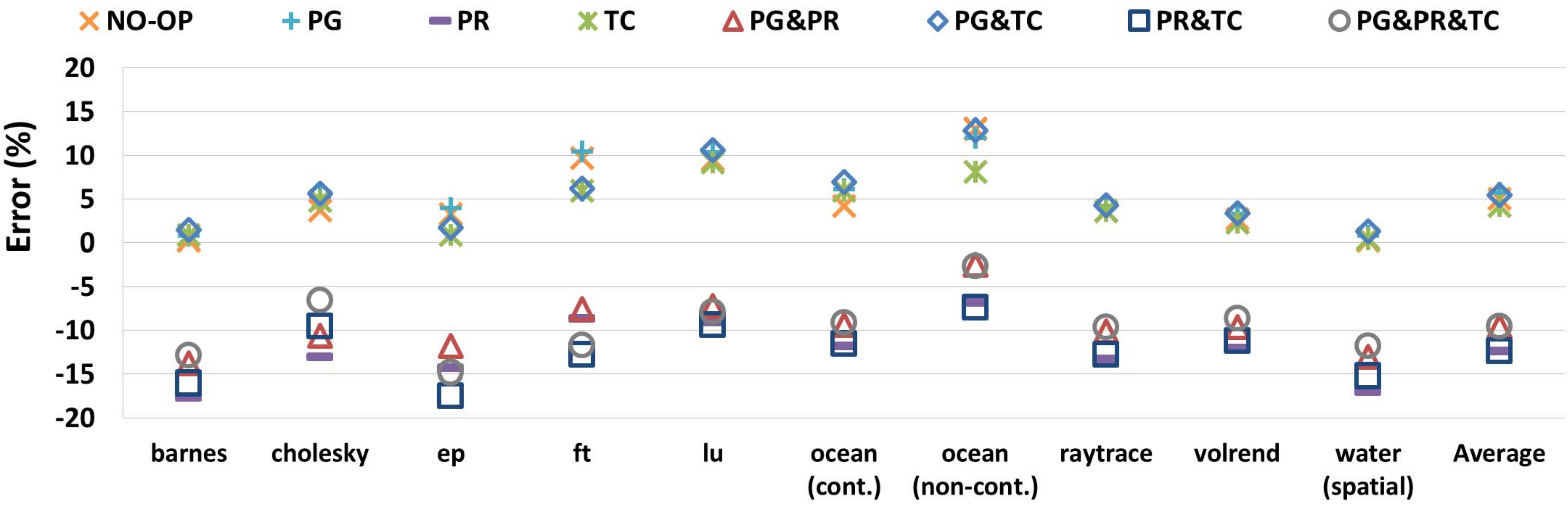
See the following paper for details:

Y. He, et. al., “Runtime Multi-Optimizations for Energy Efficient On-chip Interconnections”, ICCD2015.

Performance Model Validation

- Models are validated against simulations (the baseline)

NO-OP: w/o Opt., PG: Power Gating, PR: Prediction Router (PR), TC: Traffic Compression

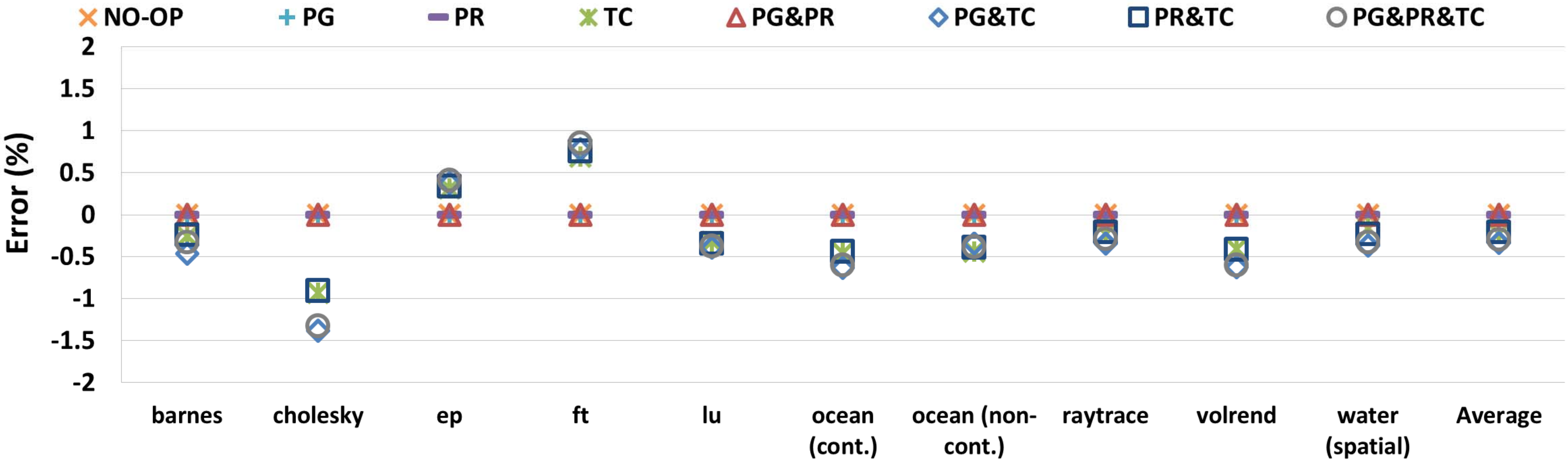


- Errors are between +15% and -20%
  - Mostly come from the queueing model
- Not perfectly accurate, but enough for optimization selection

Energy Model Validation

- Models are validated against simulations (the baseline)

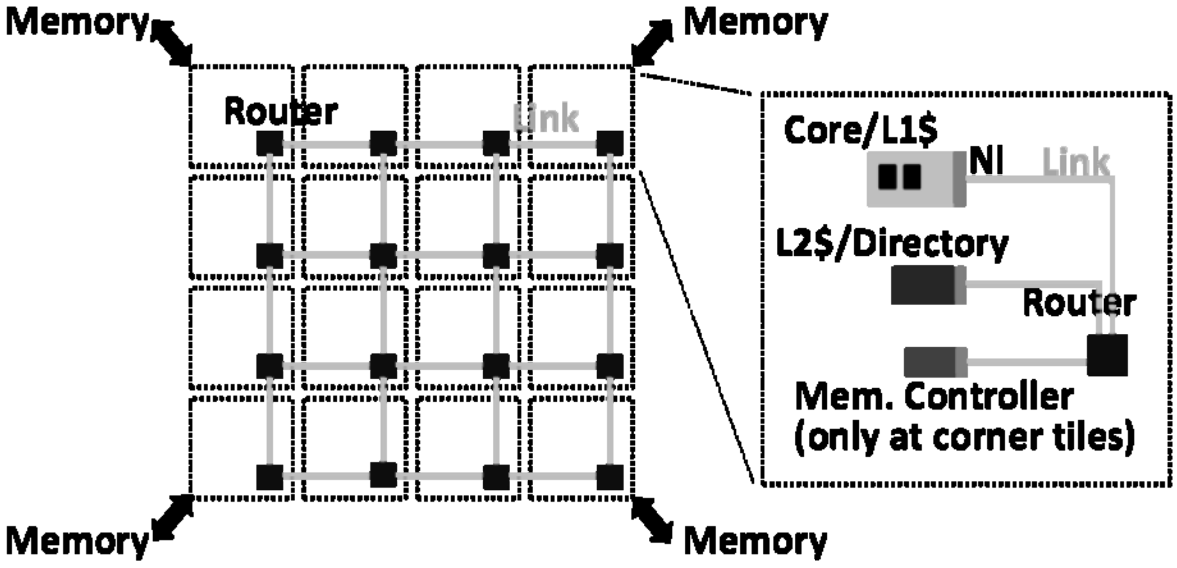
NO-OP: w/o Opt., PG: Power Gating, PR: Prediction Router (PR), TC: Traffic Compression

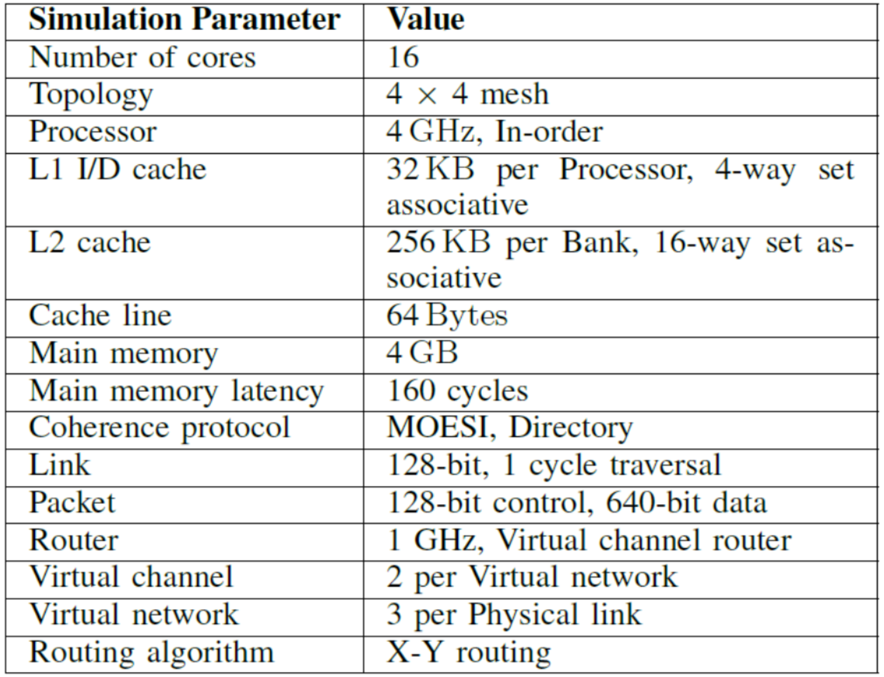


- Errors are between +1% and -1.5%
- Very accurate since the way to model power consumption is the same as simulation environment

Evaluation Methodology

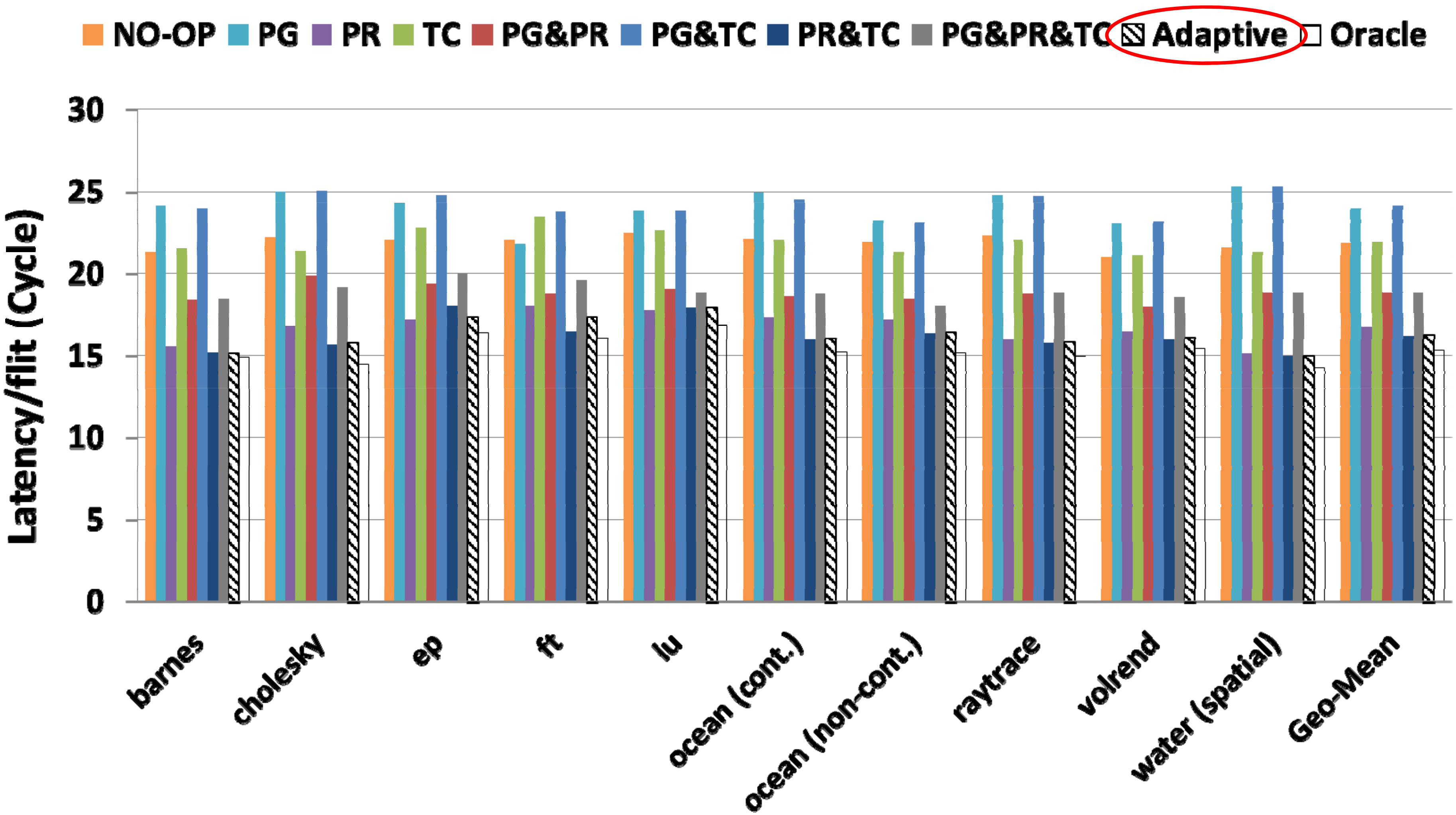
- GEMS/Simics with Garnet and Orion 2 for simulating the target manycore SoC with NoC
  - During simulation, we collected periodic traces of related performance stats and then obtain the results using offline analyses
- Shutter period: 10K instructions
- Epoch size: 100K instructions





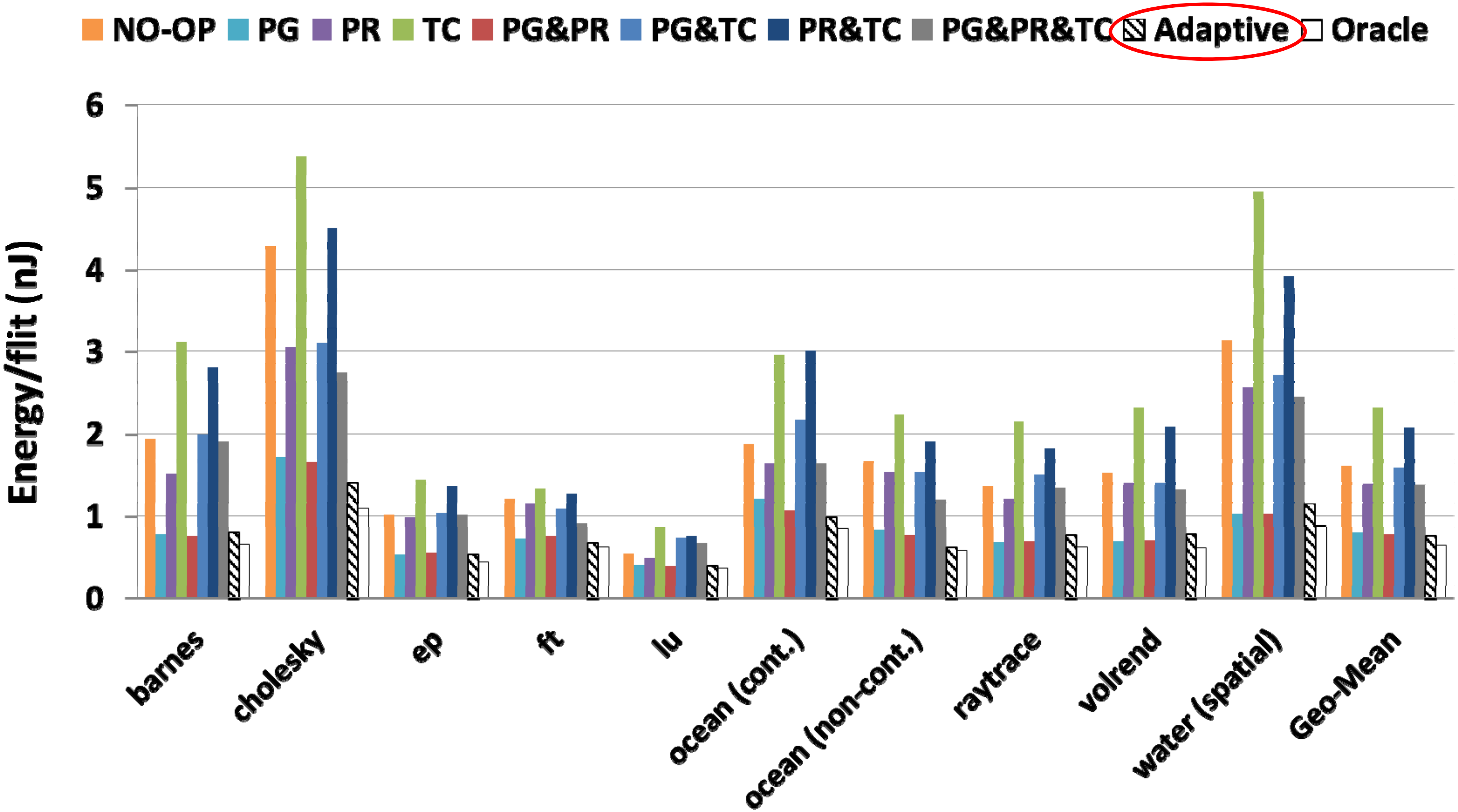
Simulation Parameter	Value
Number of cores	16
Topology	4 × 4 mesh
Processor	4 GHz, In-order
L1 I/D cache	32 KB per Processor, 4-way set associative
L2 cache	256 KB per Bank, 16-way set associative
Cache line	64 Bytes
Main memory	4 GB
Main memory latency	160 cycles
Coherence protocol	MOESI, Directory
Link	128-bit, 1 cycle traversal
Packet	128-bit control, 640-bit data
Router	1 GHz, Virtual channel router
Virtual channel	2 per Virtual network
Virtual network	3 per Physical link
Routing algorithm	X-Y routing

Performance Result



- Adaptive runtime framework has very good outcome
  - Second to oracle, which means almost the best groups of optimizations are chosen for each epoch

Energy Efficiency Result



- Again, adaptive runtime framework achieves good result
  - Second to oracle, which means almost the best groups of optimizations are chosen for each epoch in terms of energy

Summary

- We proposed and evaluated a model-based runtime adaptive framework for determining the best group of NoC optimizations
- The framework works well as its resulting network performance and energy are only second to oracle
- We can achieve 26% performance improvement and 57% energy saving, respectively over “no optimization” case
- Acknowledgement
  - Yuan He (The University of Tokyo)
  - Takashi Nakada (The University of Tokyo)
  - Hiroshi Sasaki (Columbia University)
  - Shinobu Miwa (The University of Electro-Communications)
  - Hiroshi Nakamura (The University of Tokyo)