Exploring Many-Core Architecture Design Space for Parallel Discrete Event Simulation

Yi Zhang, Jingjing Wang, Dmitry Ponomarev, and Nael Abu-Ghazaleh

Computer Science Department
Binghamton University, State University of New York
{yzhang25, jwang36, dima, nael}@cs.binghamton.edu

May 20, 2014

ACM SIGSIM Conference
on
Principles of Advanced Discrete Simulation (PADS)

BINGHAMTON
UNIVERSITY
STATE UNIVERSITY OF NEW YORK

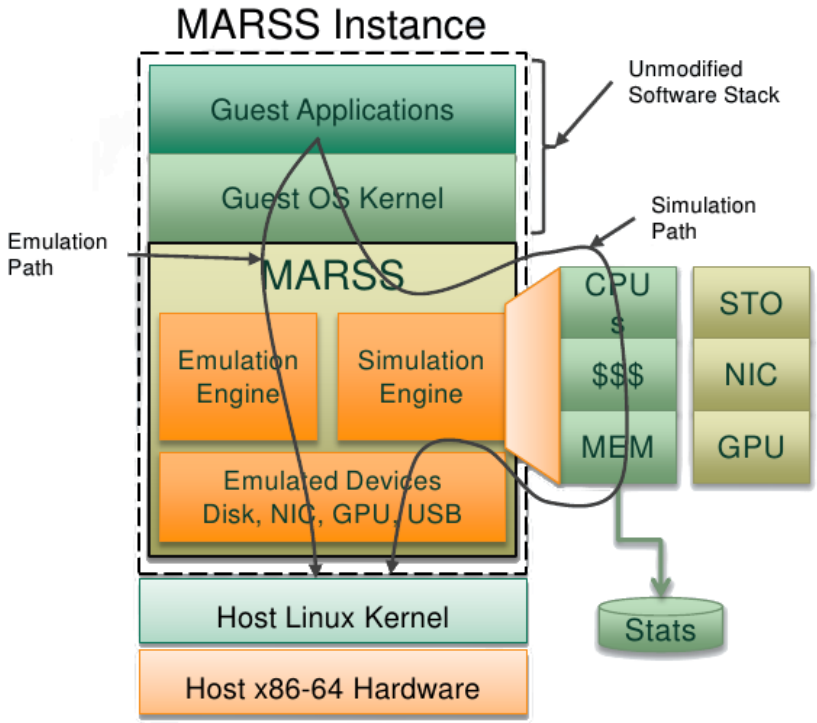
Motivation

- Emerging many-core architectures have a potential to significantly accelerate PDES
- Manycores differ substantially in their core architectures, inter-core connectivity and memory hierarchy design
- **Goal:** understand PDES performance on manycores systematically, without being limited by existing designs
- **Approach:** simulation-based study using cycle-accurate full-system performance simulator with PDES as benchmark

Simulating PDES: Questions to Answer

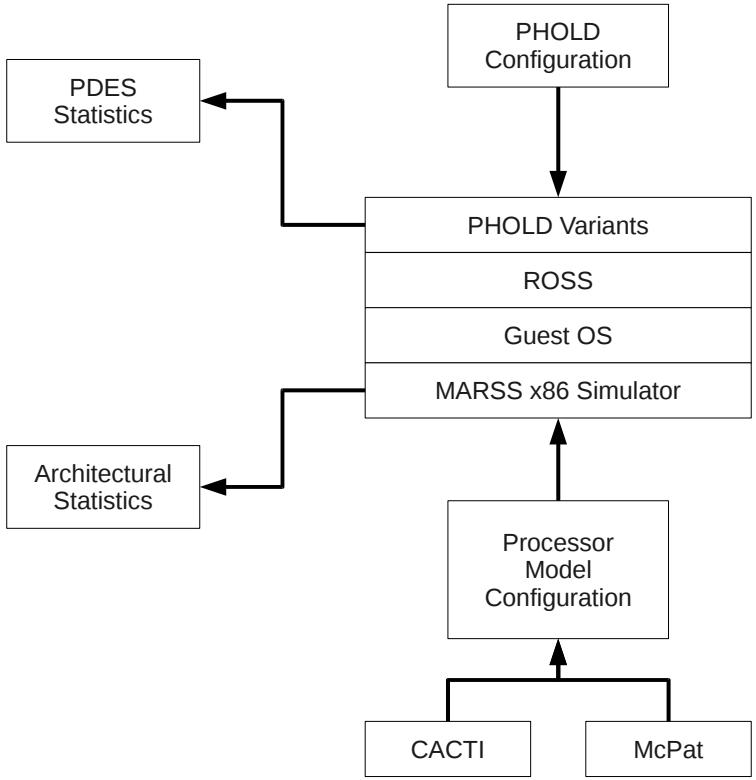
- What is the optimal trade-off between the core count and the cache size for PDES?
- What kind of cores provide the best performance/complexity point?
- What is the performance impact of heterogeneous architectures?
  - Can better PDES partitioning help?

Simulation framework: MARSS



- Models both inorder and out-of-order cores
- Provides flexible configurations for the on-chip cache hierarchy
- Supports full-system simulation
- Can switch between detailed simulation and emulation mode

Modeling Power, Area and Delays



- McPAT tool (Jouppi et al., MICRO 2009) estimates the area requirements of the individual cores
- CACTI tool provides latencies for differently sized last-level caches
- Multithreaded ROSS (ROSS-MT) with Phold model serves as the benchmark

Models for Evaluating the Impact of Cache Size and Core Type/Count

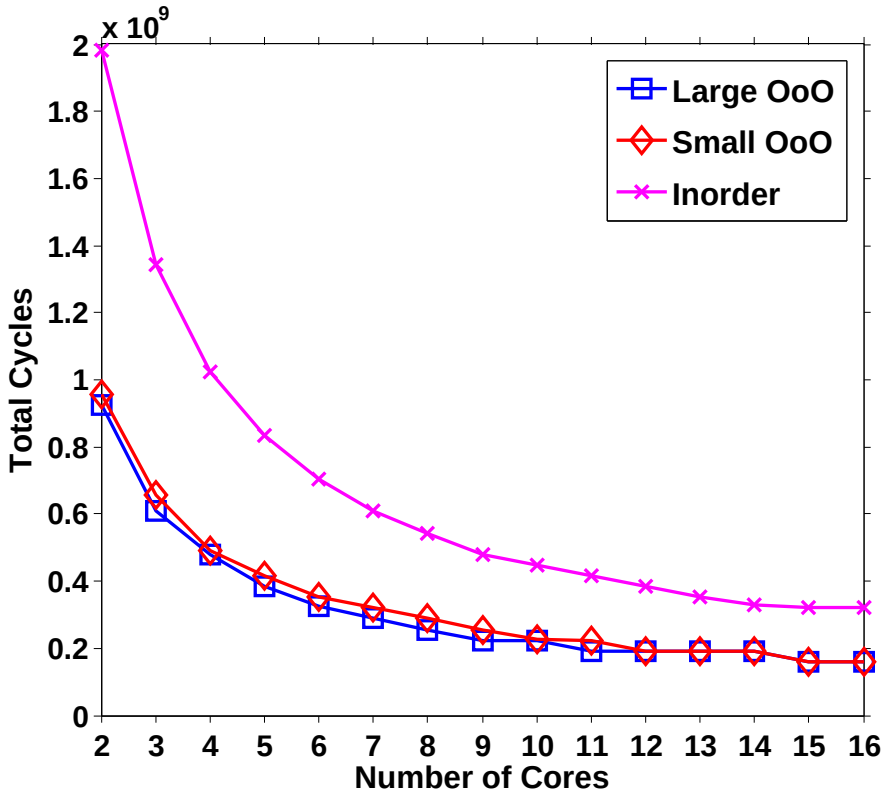
- Model 1: Area-unconstrained homogeneous systems with fixed-size shared L3 cache and variable number of cores
  - Three types of cores considered:
    - Large out-of-order core (Large OoO)
    - Small out-of-order core (Small OoO)
    - Small in-order core
- Model 2: Area-constrained homogeneous systems, with a trade-off between the size of the on-chip cache and the number of cores

Core Types and their Parameters

	Large OoO	Small OoO	Small In-order
Issue Width	6	3	2
Commit Width	4	3	N/A
ROB Size	168	64	N/A
Instruction Queue Size	32	32	16
ALU	6	3	1
FPU	6	3	1
Load Queue Size	48	24	N/A
Store Queue Size	96	24	N/A
Private L1-I Cache Size	32KB	32KB	32KB
Private L1-D Cache Size	32KB	32KB	32KB
Private L2 Cache Size	256KB	256KB	256KB
Core Size (sqmm)	19.6154	13.0023	5.2573

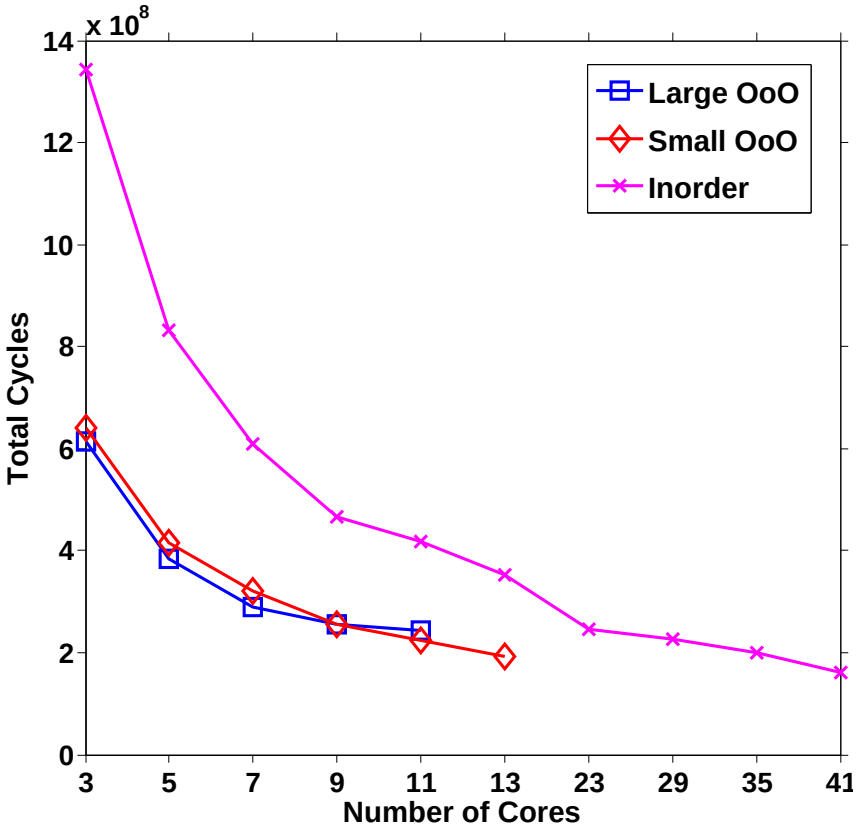
- Core size: Large OoO $>$ Small OoO $>$ Inorder core
- Processing speed: Large OoO $>$ Small OoO $>$ Inorder core

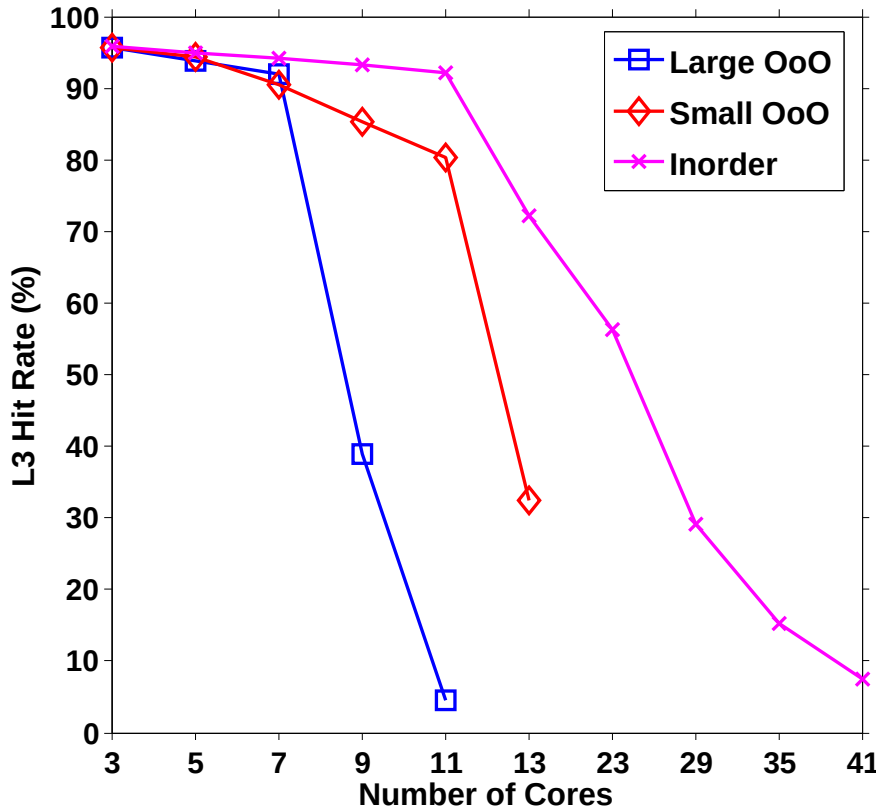
Area-unconstrained Systems with Fixed L3 Cache Size



- Large OoO cores provide similar performance with small OoO cores, but significantly outperform in-order core

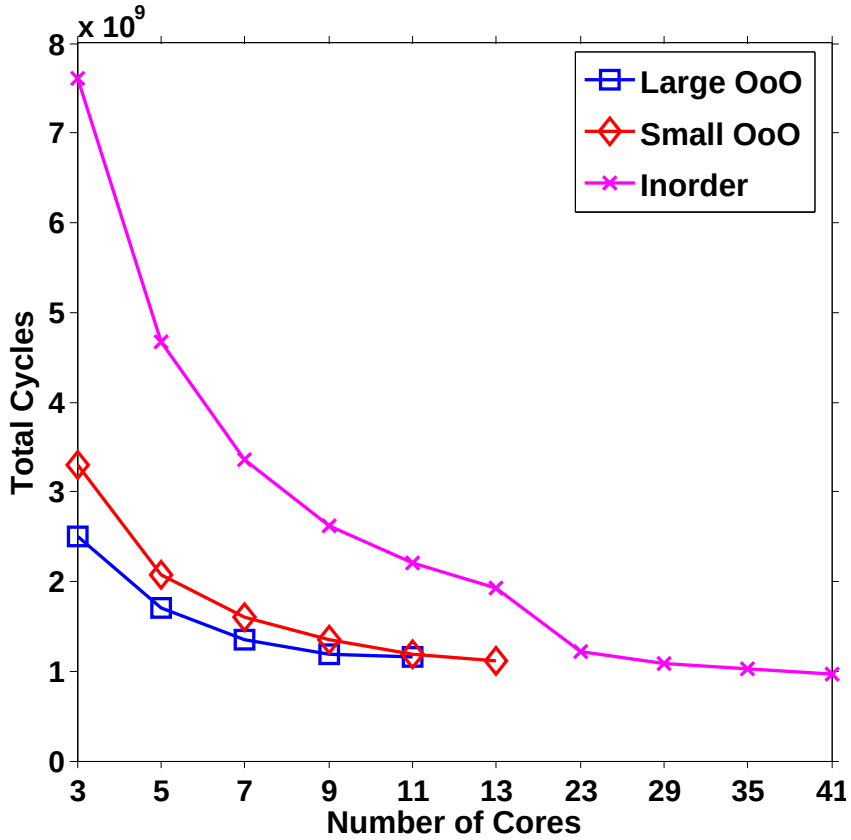
Area-constrained Systems: Tradeoff between L3 Cache Size and Core Counts





- A larger number of in-order cores results in higher performance
- With more cores, L3 cache becomes smaller and the cache hit rate decreases. However, impact on PDES is small.

Performance Impact of Higher Memory Pressure



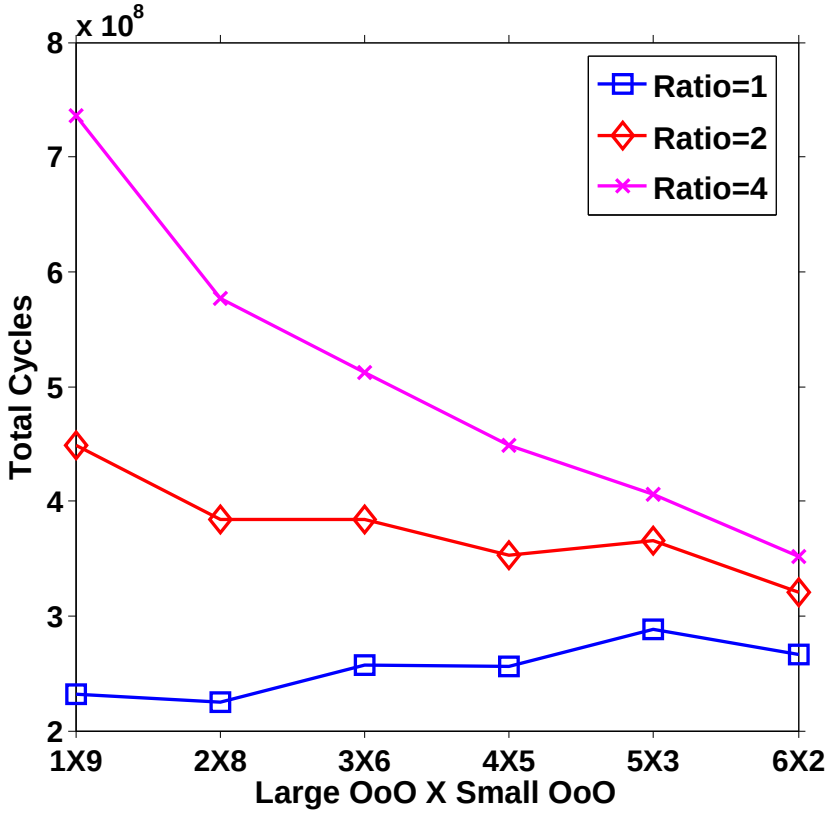
- We added memory operations during each event processing in the Phold model
- L3 cache has a limited impact on simulation performance for these models
  - High locality in the private L1 and L2 caches
  - The number of accesses to L3 is small

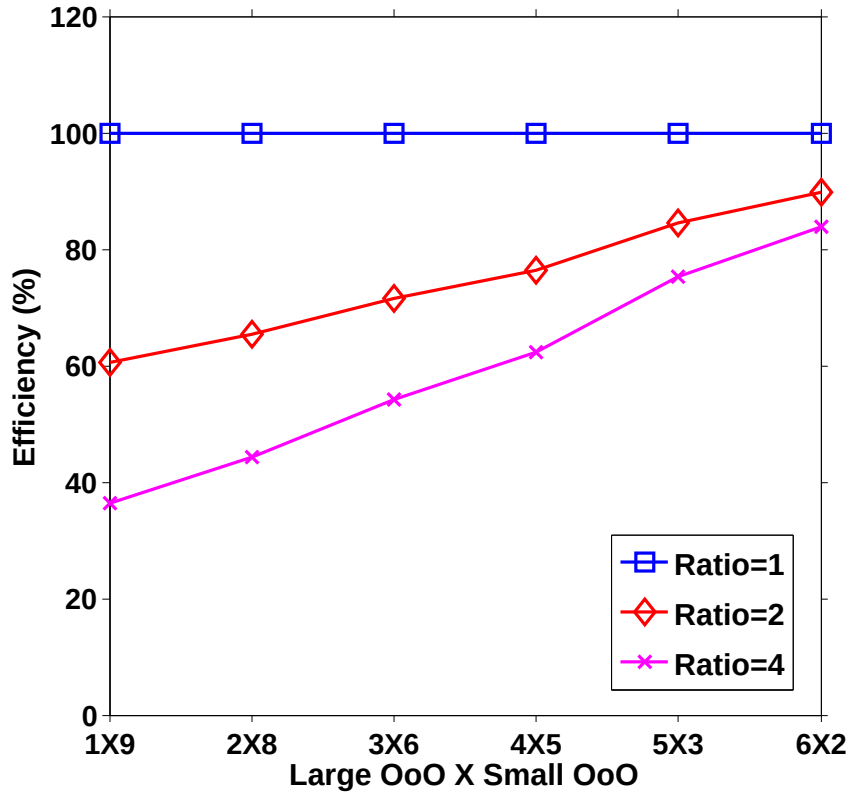
Performance Impact of Core Heterogeneity

- Area-constrained heterogeneous system composed of multiple types of cores
- Workload partitioning: map different number of simulation objects to different cores using a specific ratio

$$Ratio = \frac{Objects\ on\ Each\ Larger\ Core}{Objects\ on\ Each\ Smaller\ Core}$$

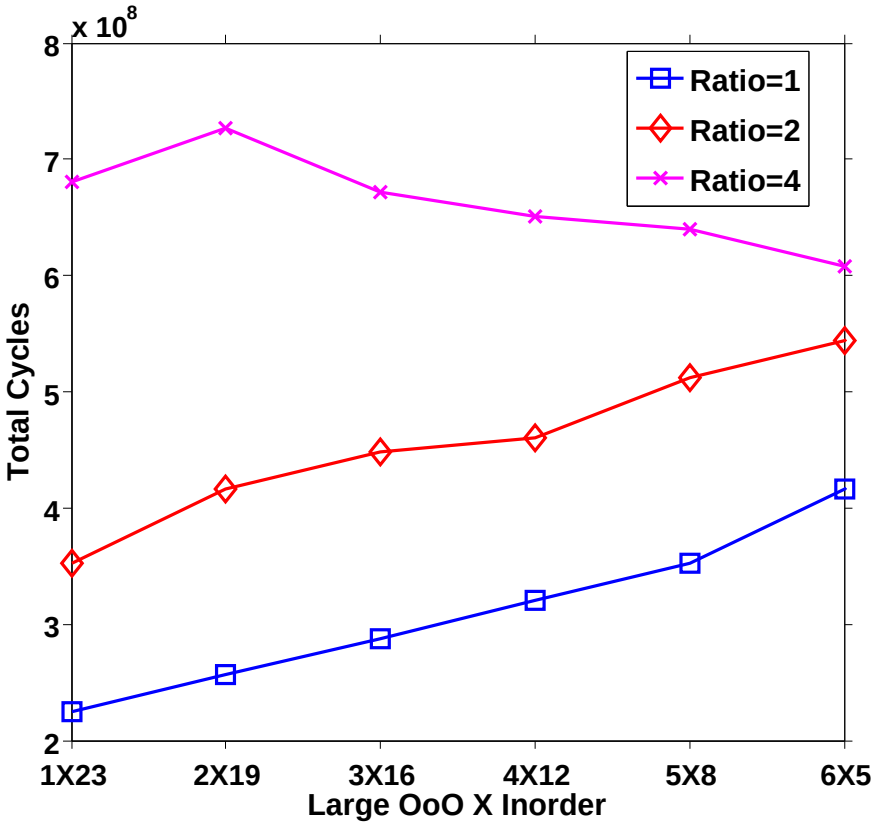
Heterogeneous System with Large and Small OoO Cores

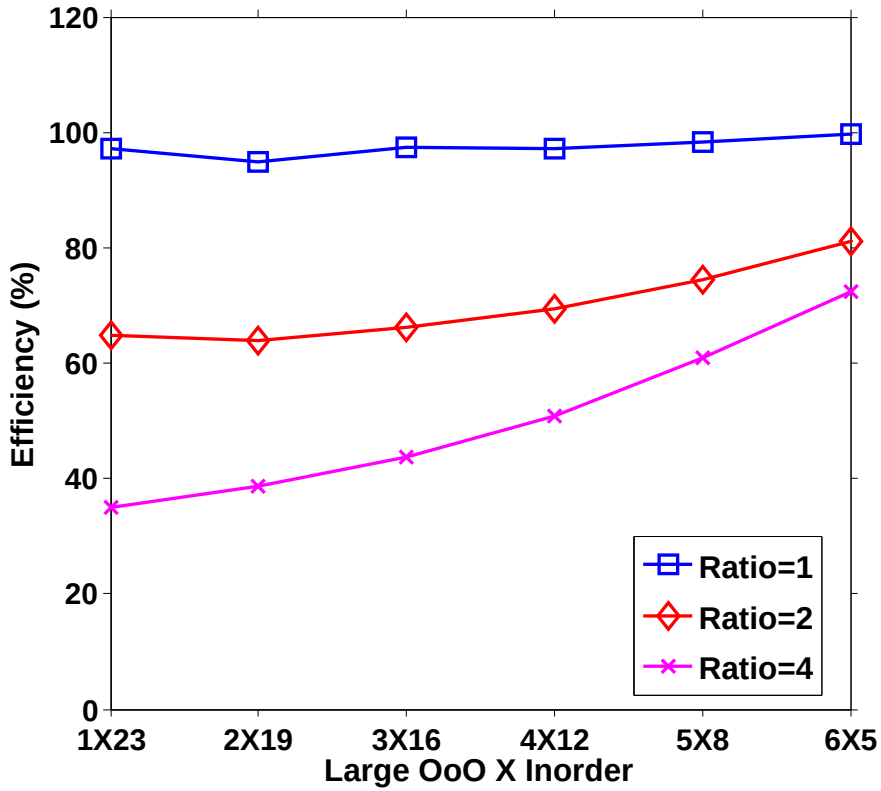




- Ratio=1: best performance is achieved at this ratio
- Ratio=2: performance improves when more larger cores are used
- Ratio=4: execution time is dominated by the rollbacks

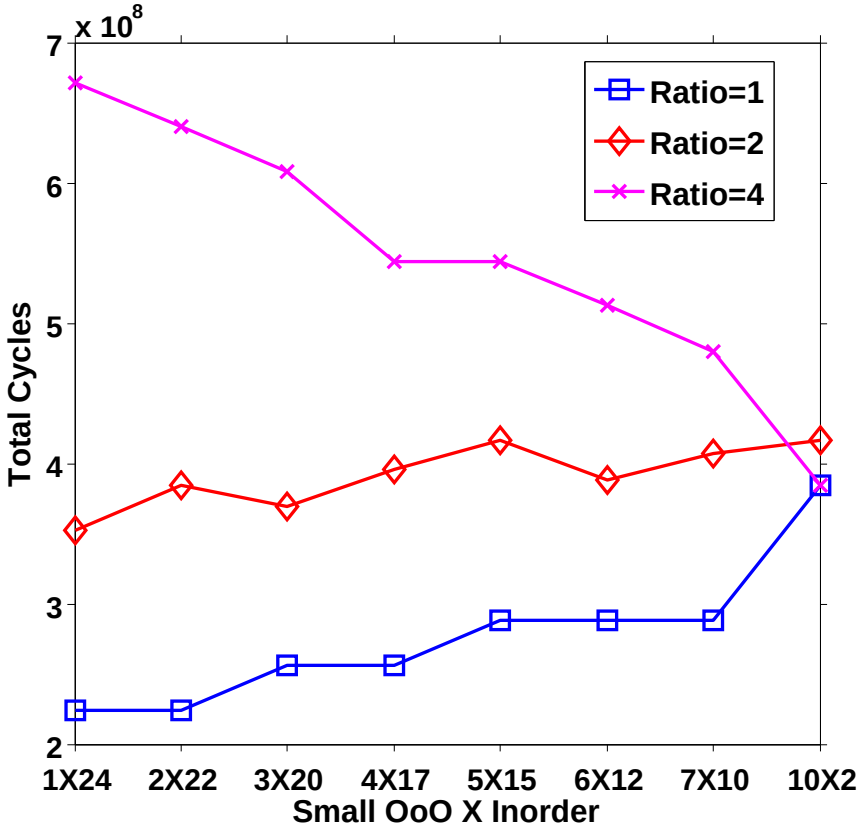
Heterogeneous System with Large OoO and In-order Cores

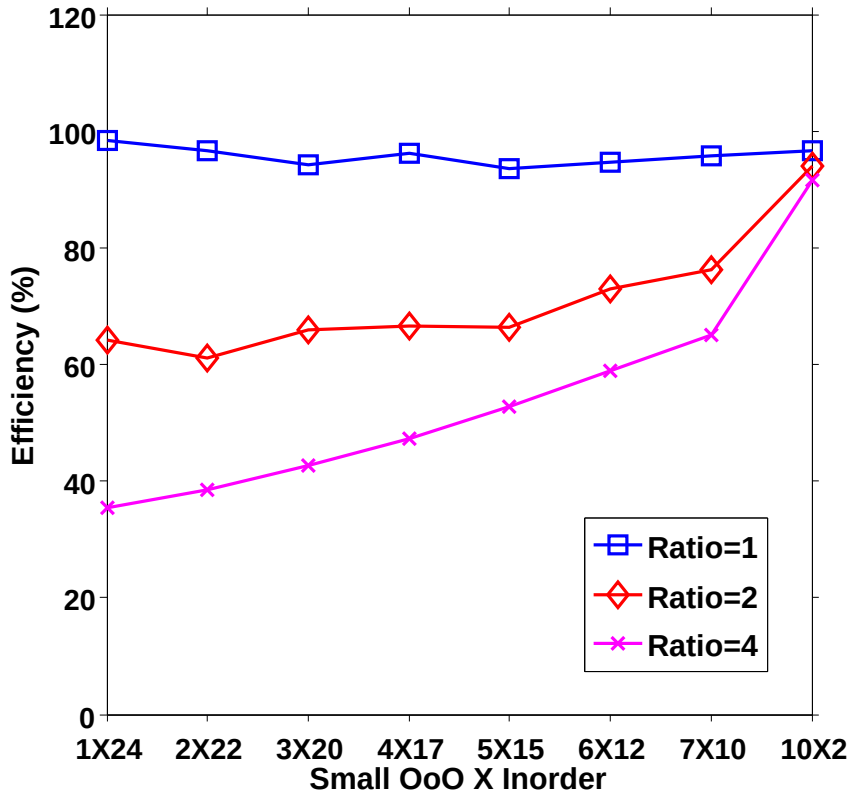




- Ratio=1 or 2: Performance degrades when the number of in-order cores gets smaller

Heterogeneous System with Small OoO and In-order Cores





- Similar trend to the system of large OoO and in-order cores
  - Large OoO and small OoO have similar performance
- Need to determine the best-performing partitioning ratio

Conclusions

- For area-unconstrained design, small out-of-order cores provide the best performance/complexity trade-off for the same size of the last-level cache
- For area-constrained design, the best performance is achieved with the largest possible number of simple in-order cores
- The shared L3 cache has a minimal performance impact for Phold-style PDES applications
- Core heterogeneity negatively impacts PDES performance because it distorts the simulation synchrony. Even if efficiency is high, performance is constrained by the smaller cores. Not clear if heterogeneity-aware partitioning can help — further studies are needed.

Future Work

- Determine the optimal partitioning strategy for heterogeneous systems
- Use reproducibility of simulation results to study causes of rollbacks in optimistic simulation and other subsystems of the PDES engine
- Evaluation of power/performance trade-offs in the architecture design space
- Evaluation of dedicated hardware support for accelerating critical PDES subsystems

Thanks!!

Q & A