A System-Level ISA and its Applications to Energy-Performance-Reliability Scheduling and Scratchpad Allocation

Yavuz Yetim, Wenhao Jia, Margaret Martonosi, Sharad Malik, Kelly Shaw*
Princeton University * University of Richmond

Platforms and AlternativeThemes,
Task #5.2.1 and 5.6.3

GSRC Annual
Symposium
September 28, 2010
through
October 1, 2010

Motivation

Problems:

- Rapidly increasing core counts
- Various heterogeneity (GPU, faster/slower cores)
- Decreasing reliability

Goal: Achieve performance, energy and reliability demands in future heterogeneous and dynamically changing multicore systems

SISA: Approach

Represent programs as graphs with app characteristics

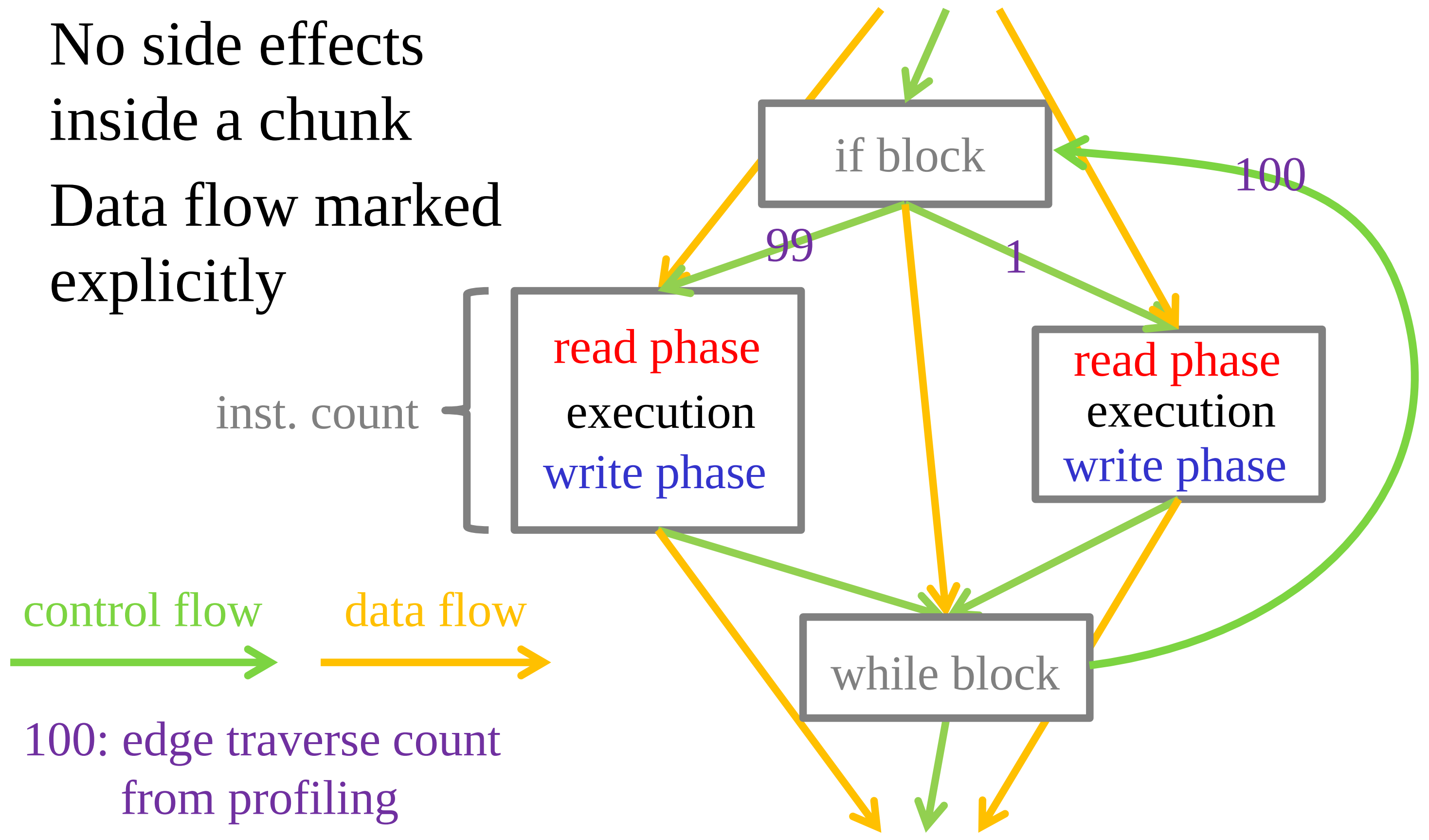
- Data communication
- Length of computational tasks
- Reliability requirements
- Task dependency

Use SISA representation to do

- Static scheduling with Integer Linear Programming
- Pre-run resource mapping (such as scratchpad)
- Dynamic task management

SISA Graph

- No side effects inside a chunk
- Data flow marked explicitly



Acknowledgements

The authors acknowledge the support of the Gigascale Systems Research Center, one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program. In addition, this work was supported in part by National Science Foundation grant CCF-0916971.

Application 1: Static Scheduling for Energy/Performance/Reliability

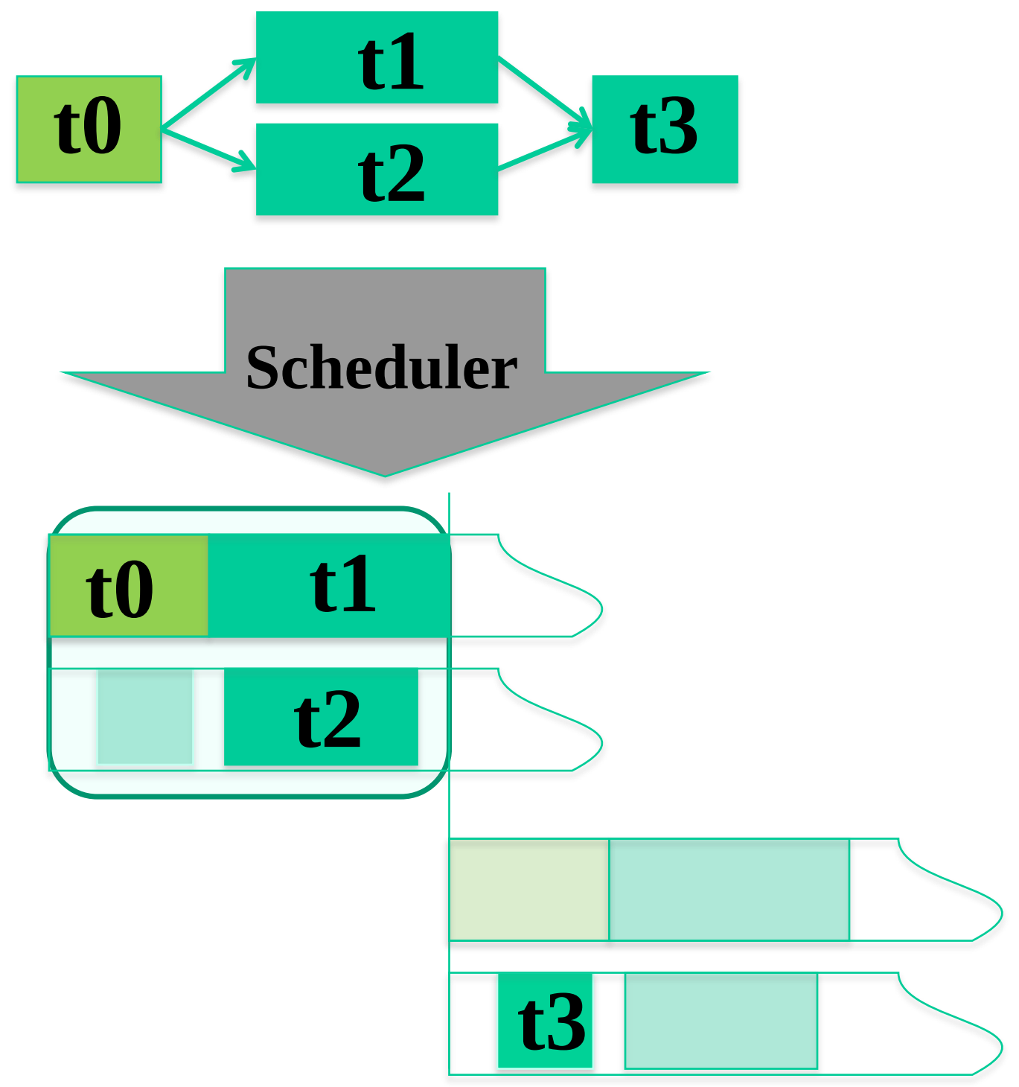
Novel Integer Linear Programming Formulation that

✓ optimizes periodic applications using pipelining
✓ includes communication overheads
✓ handles “lock” variables through mutual exclusion

Objective: Busy Energy + Idle Energy + Data Migration Overhead

Constraints:

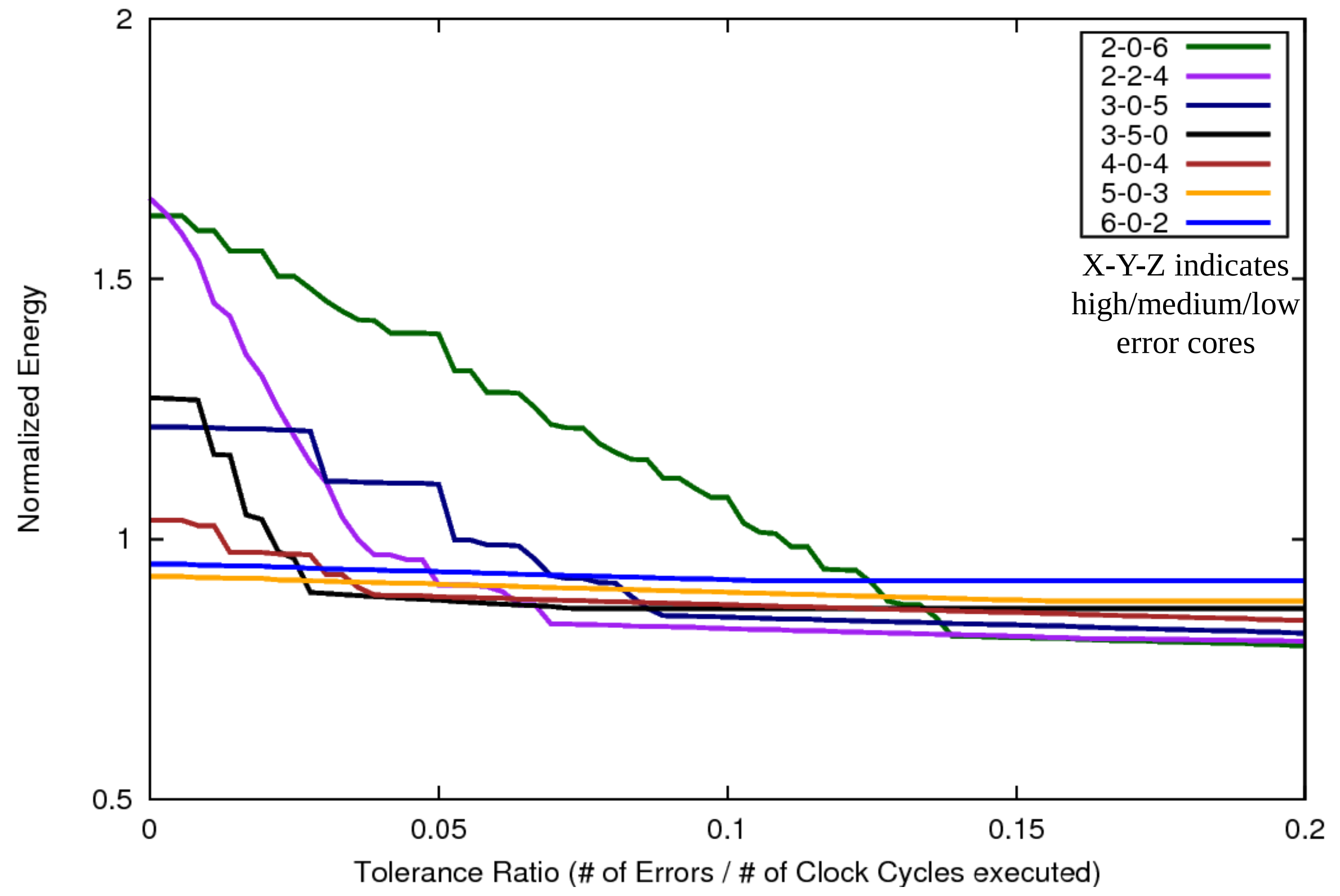
- Overlap and Sequencing
- Error Tolerance
- Mutual Exclusion
- Pipelining
- Period and Deadline



Results

Scheduler is run for StreamIt benchmarks on different configurations, which are assumed to be DVFS overscaled. Energy gains depend on:

- Hardware configuration
  - High idle power decreases gains
  - High error rate decreases gains for low error tolerance but increases gains for high error tolerance
- Application characteristics
  - Applications with workload-balanced tasks have better gains
  - Error tolerance of an application decreases energy significantly up to a point and flattens after all cores can be utilized

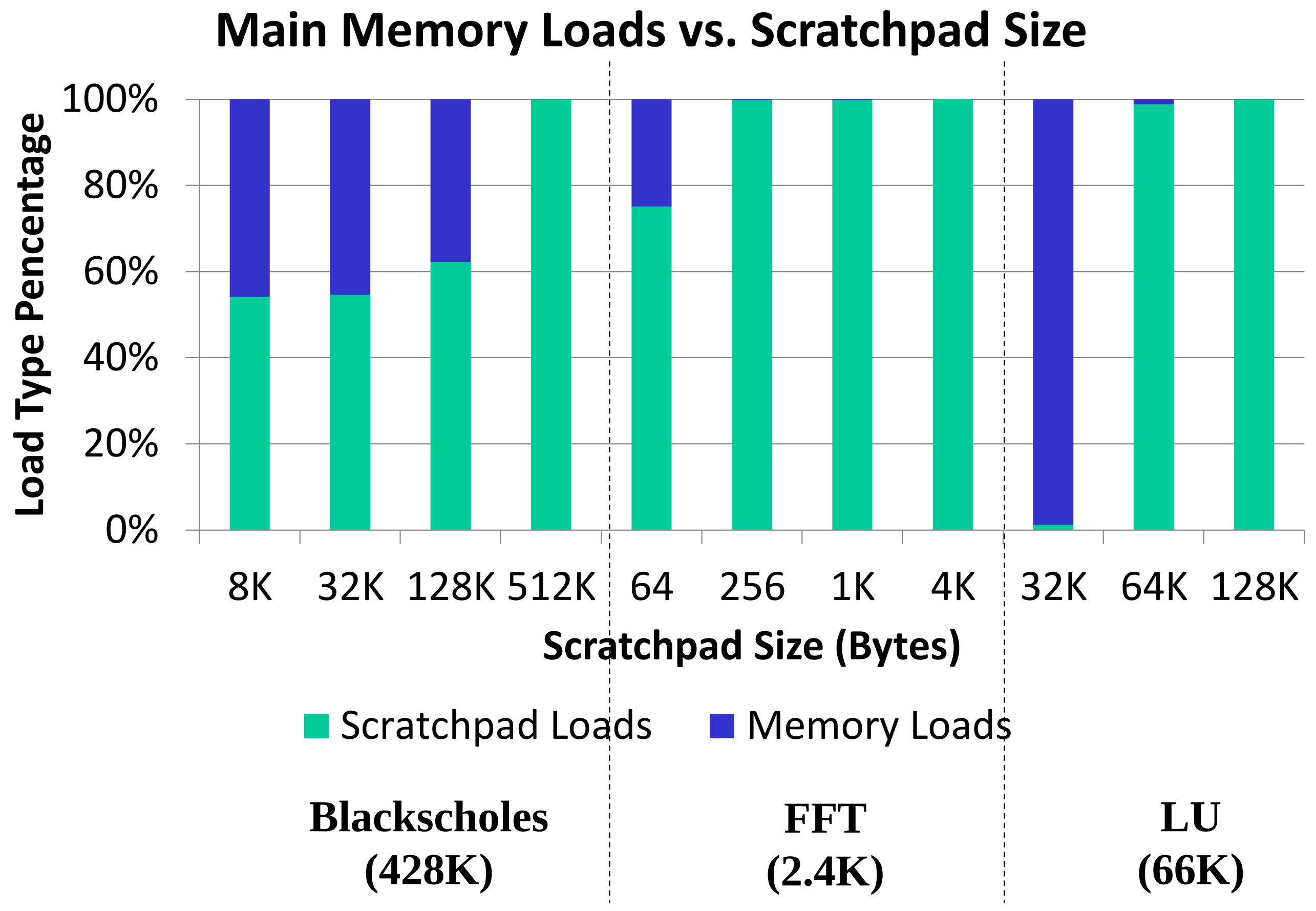


Application 2: Scratchpad Memory Allocation

Objective: Minimize program execution time on machines with local software-controlled memory

Method:

- Use the SISA graph representation of an application that has memory accesses and data flow marked
- Allocate the scratchpad for variables on the critical path through the Control Flow Graphs
- Allocate the scratchpad for the rest variables



Results:

- On PARSEC benchmark suite
- Global memory loads reduce as available scratchpad space increases

Conclusions

- SISA graph-based program representation effectively exploits Performance/Energy/Reliability space
- Up to 34% energy savings can be achieved given application reliability requirements
- Based on data flow exposed by SISA graph, up to 99% memory loads can be eliminated with reasonable scratchpad size