HPC Node Performance and Power Simulation with the Sniper Multi-Core Simulator

Trevor E. Carlson,

Wim Heirman, Lieven Eeckhout

http://www.snipersim.org
Saturday, February 1st, 2014
FOSDEM 2014 – HPC Devroom – Brussels, Belgium

Major Goals of Sniper

- What will node performance look like for next-generation systems?
  - Intel Xeon, Xeon Phi, etc.
- What optimizations can we make for these systems?
  - Software Optimizations
  - Hardware / Software co-design
- How is my application performing?
  - Detailed insight into application performance on today's systems

Optimizing Tomorrow's Software

- ~~Design tomorrow's processor using today's hardware~~
- Optimize tomorrow's software for tomorrow's processors
- Simulation is one promising solution
  - Obtain performance characteristics for new architectures
  - Architectural exploration
  - Early software optimization

WHY CAN'T I JUST …

use performance counters?

- perf stat, perf record

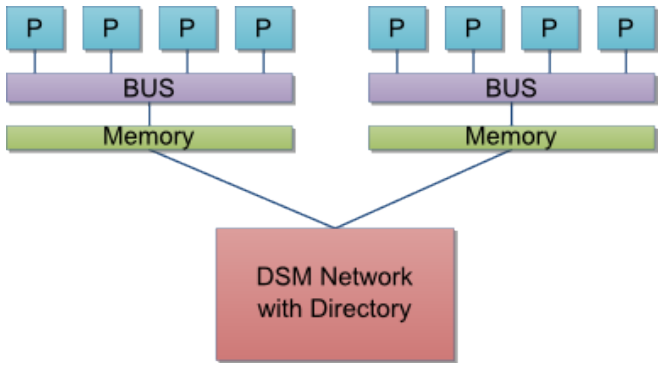
use Cachegrind?

It can be difficult to see exactly where the problems are

- Not all cache misses are alike – latency matters
- Modern out-of-order processors can overlap misses
- Both core and cache performance matters

Node-Complexity is Increasing

- Significant HPC node architecture changes
  - Increases in core counts
    - More, lower-power cores (for energy efficiency)
  - Increases in thread (SMT) counts
  - Cache-coherent NUMA



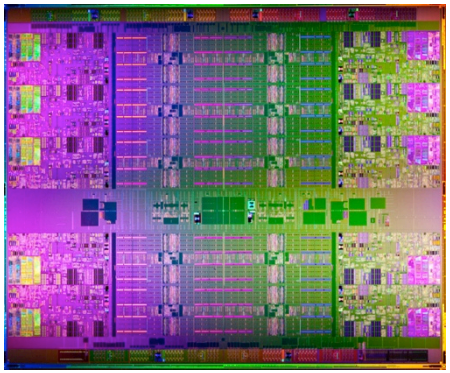
Source: Wikimedia Commons

- Optimizing for efficiency
  - How do we analyze our current software?
  - How do we design our next-generation software?

TRENDS IN PROCESSOR DESIGN: CORES

Number of cores per node is increasing

- 2001: Dual-core POWER4
- 2005: Dual-core AMD Opteron
- 2011: 10-core Intel Xeon Westmere-EX
- 2012: Intel MIC Knights Corner (60+ cores)
- 2013: Intel MIC Knights Landing announced[1]

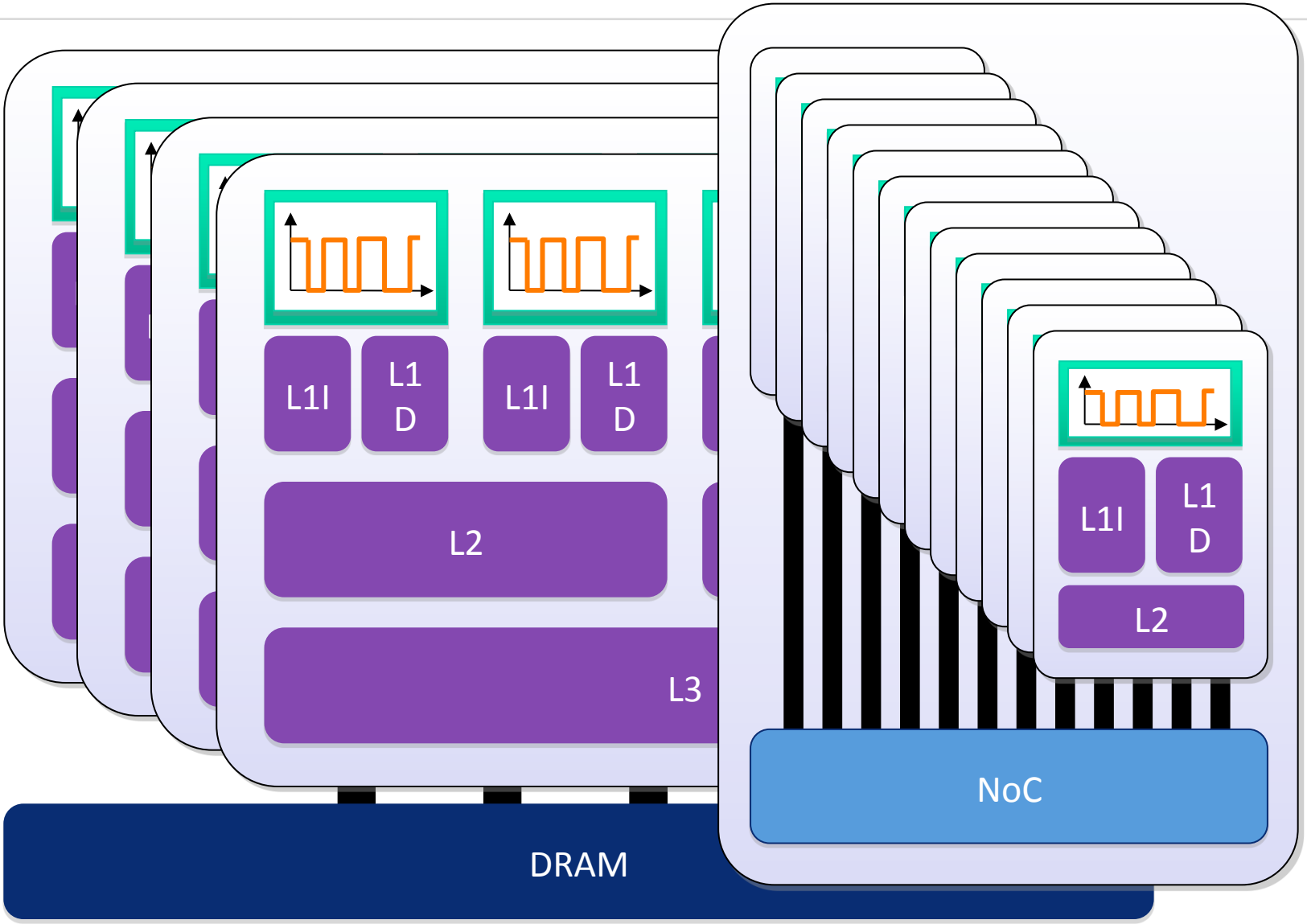


Westmere-EX, Source: Intel

Xeon Phi, Source: Intel

[1]http://newsroom.intel.com/community/intel_newsroom/blog/2013/06/17/intel-powers-the-worlds-fastest-supercomputer-reveals-new-and-future-high-performance-computing-technologies

Many Architecture Options



Upcoming Challenges

- Future systems will be diverse
  - Varying processor speeds
  - Varying failure rates for different components
  - Homogeneous applications show heterogeneous performance
- Software and hardware solutions are needed to solve these challenges
  - Handle heterogeneity (reactive load balancing)
  - Handle fault tolerance
  - Improve power efficiency at the algorithmic level (extreme data locality)
- Hard to model accurately with analytical models

Fast and Accurate Simulation is Needed

- Evaluating current software on current hardware is difficult
  - Performance counters do not provide enough insight
- Simulation use cases
  - Pre-silicon software optimization
  - Architecture exploration
- Cycle-accurate simulation is too slow for exploring multi/many-core design space and software
- Key questions
  - Can we raise the level of abstraction?
  - What is the right level of abstraction?
  - When to use these abstraction models?

SNIPER: A FAST AND ACCURATE SIMULATOR

- Hybrid simulation approach
  - Analytical interval core model
  - Micro-architecture structure simulation
    - branch predictors, caches (incl. coherency), NoC, etc.
- Hardware-validated, Pin-based
- Models multi/many-cores running multi-threaded and multi-program workloads
- Parallel simulator scales with the number of simulated cores
- Available at `http://snipersim.org`

Top Sniper Features

- Interval Model
- Multi-threaded Application Sampling
- CPI Stacks and Interactive Visualization
- Parallel Multithreaded Simulator
- x86-64 and SSE2 support
- Validated against Core2, Nehalem
- Thread scheduling and migration
- Full DVFS support
- Shared and private caches
- Modern branch predictor
- Supports pthreads and OpenMP, TBB, OpenCL, MPI, …
- SimAPI and Python interfaces to the simulator
- Many flavors of Linux supported (Redhat, Ubuntu, etc.)

SNIPER LIMITATIONS

- User-level
  - Not the best match for workloads with significant OS involvement
- Functional-directed
  - No simulation / cache accesses along false paths
- High-abstraction core model
  - Not suited to model all effects of core-level changes
  - Perfect for memory subsystem or NoC work
- x86 only
- But … is a perfect match for HPC evaluation

SNIPER HISTORY

- November, 2011: SC'11 paper, first public release
- March 2012, version 2.0: Multi-program workloads
- May 2012, version 3.0: Heterogeneous architectures
- November 2012, version 4.0: Thread scheduling and migration
- April 2013, version 5.0: Multi-threaded application sampling
- June 2013, version 5.1: Suggestions for optimization visualization
- September 2013, version 5.2:
  MESI/F, 2-level TLBs,
  Python scheduling
- Today: 700+ downloads from 60 countries

The Sniper Multi-Core Simulator Visualization

UNIVERSITEIT GENT

intel

ExaScience Lab
Intel Labs Europe
EXASCALE COMPUTING

http://www.snipersim.org
Saturday, February 1st, 2013
FOSDEM 2014 – HPC Devroom – Brussels, Brlgium

VISUALIZATION

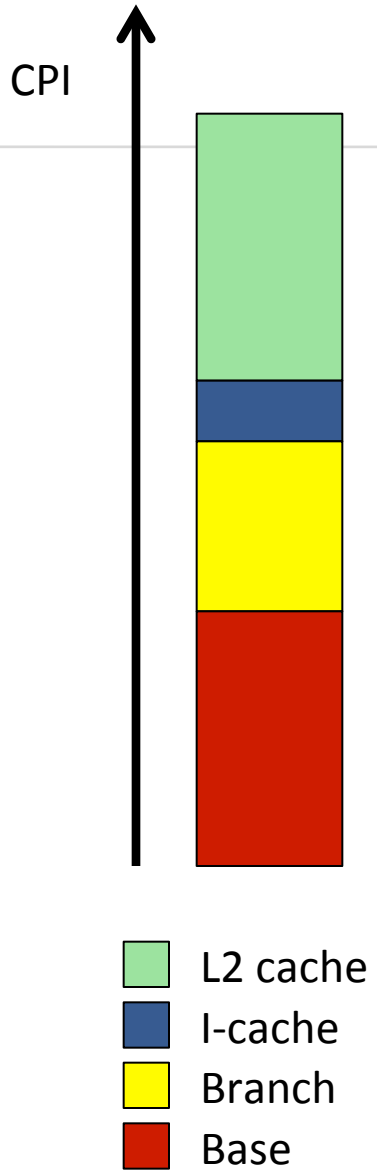
Sniper generates quite a few statistics,
but only with text is it difficult to understand
performance details

```
periodic-0.L2[1].hits-prefetch 54
periodic-0.L2[2].hits-prefetch 8
periodic-0.L2[0].evict-prefetch 56594
periodic-0.branch_predictor[0].num-correct 3373827
periodic-0.branch_predictor[1].num-correct 1363
periodic-0.branch_predictor[2].num-correct 294
periodic-0.branch_predictor[0].num-incorrect 161987
periodic-0.branch_predictor[1].num-incorrect 112
periodic-0.branch_predictor[2].num-incorrect 29
periodic-0.L1-D[0].loads-where-L1 8969301
periodic-0.L1-D[1].loads-where-L1 2063
periodic-0.L1-D[2].loads-where-L1 196
periodic-0.L1-D[0].loads-where-L2 54731
periodic-0.L1-D[1].loads-where-L2 62
periodic-0.L1-D[2].loads-where-L2 1
periodic-0.L1-D[0].stores-where-L3_S 1
periodic-0.L1-D[1].stores-where-L3_S 5
periodic-0.L1-D[2].stores-where-L3_S 5
periodic-0.L1-D[0].stores 9095
periodic-0.L1-D[1].stores 7
periodic-0.L1-D[2].stores 5
--More--(0%)
```

Text output from Sniper (sim.stats)

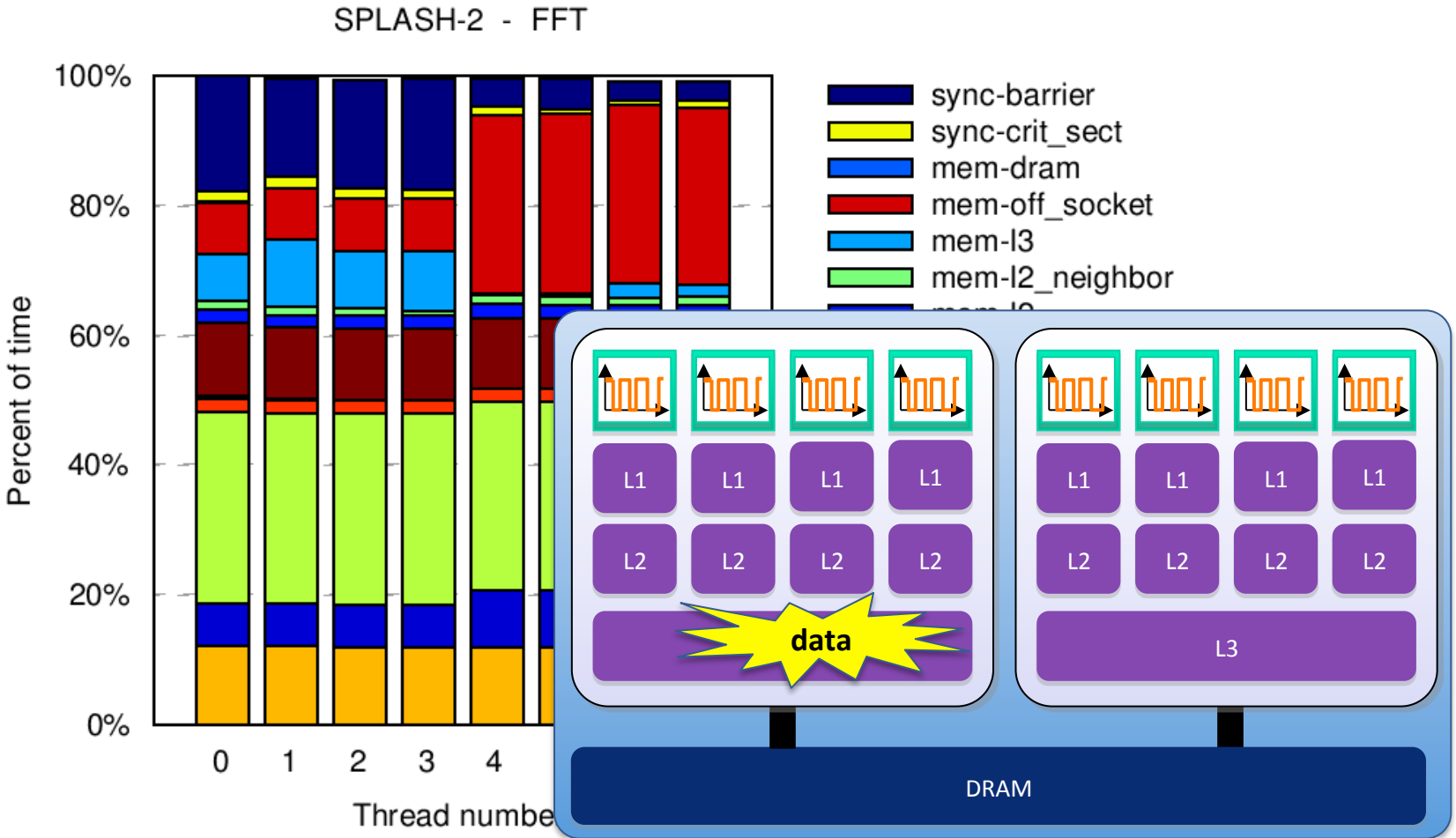
CYCLE STACKS

- Where did my cycles go?
- CPI stack
  - Cycles per instruction
  - Broken up in components
- Normalize by either
  - Number of instructions (CPI stack)
  - Execution time (time stack)
- Different from miss rates: cycle stacks directly quantify the effect on performance

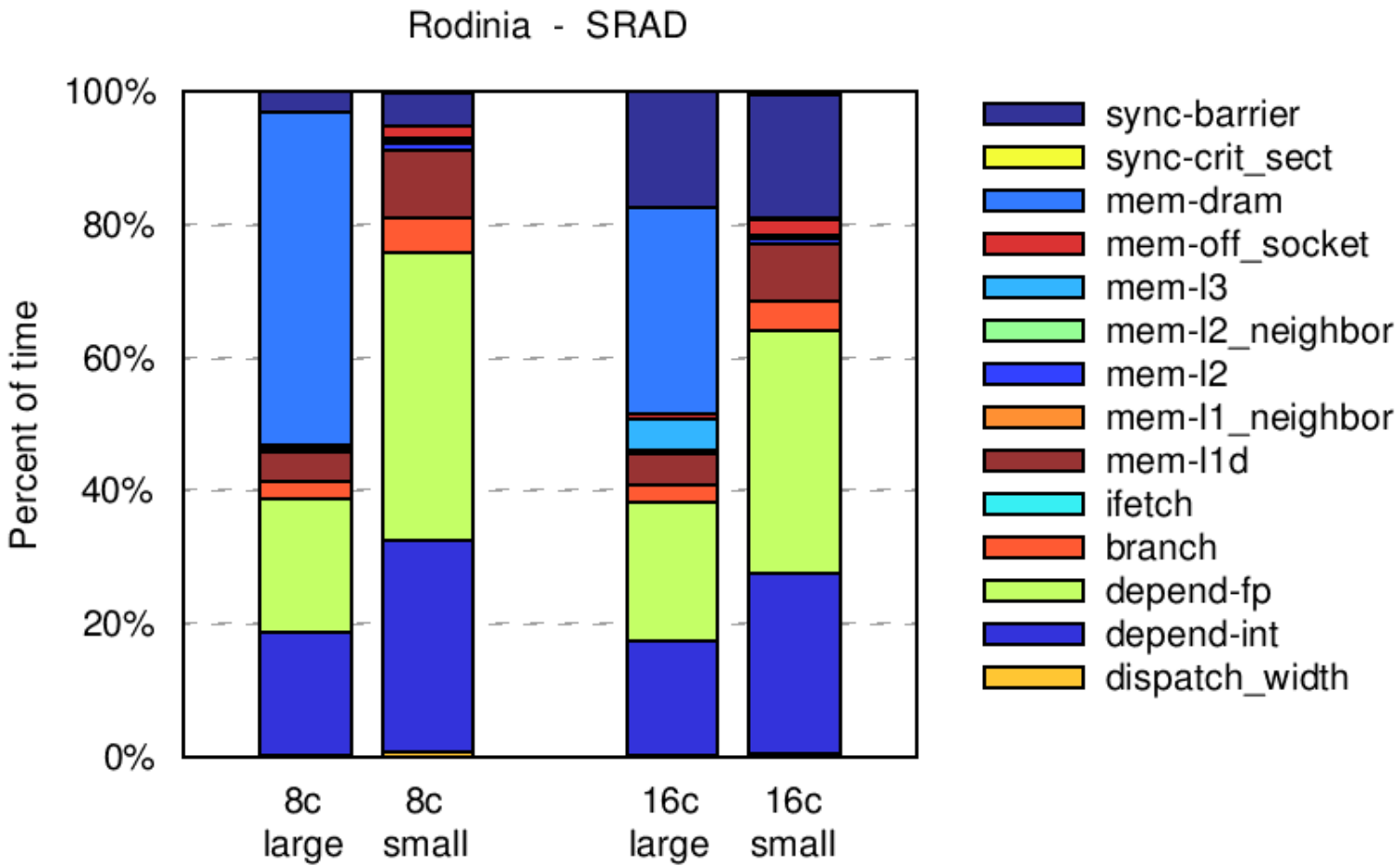


CYCLE STACKS FOR PARALLEL APPLICATIONS

By thread: heterogeneous behavior
in a homogeneous application?

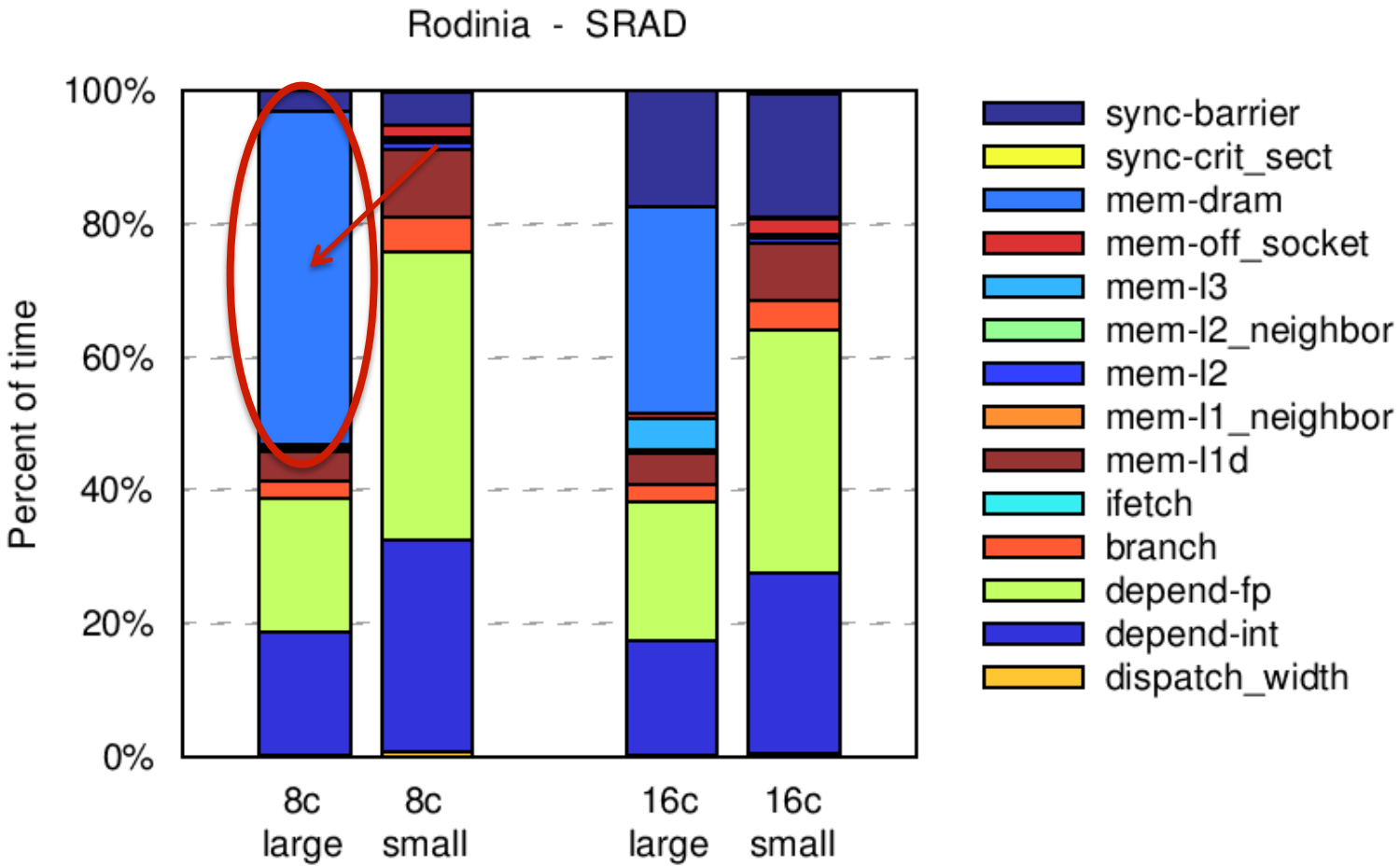


USING CYCLE STACKS TO EXPLAIN SCALING BEHAVIOR



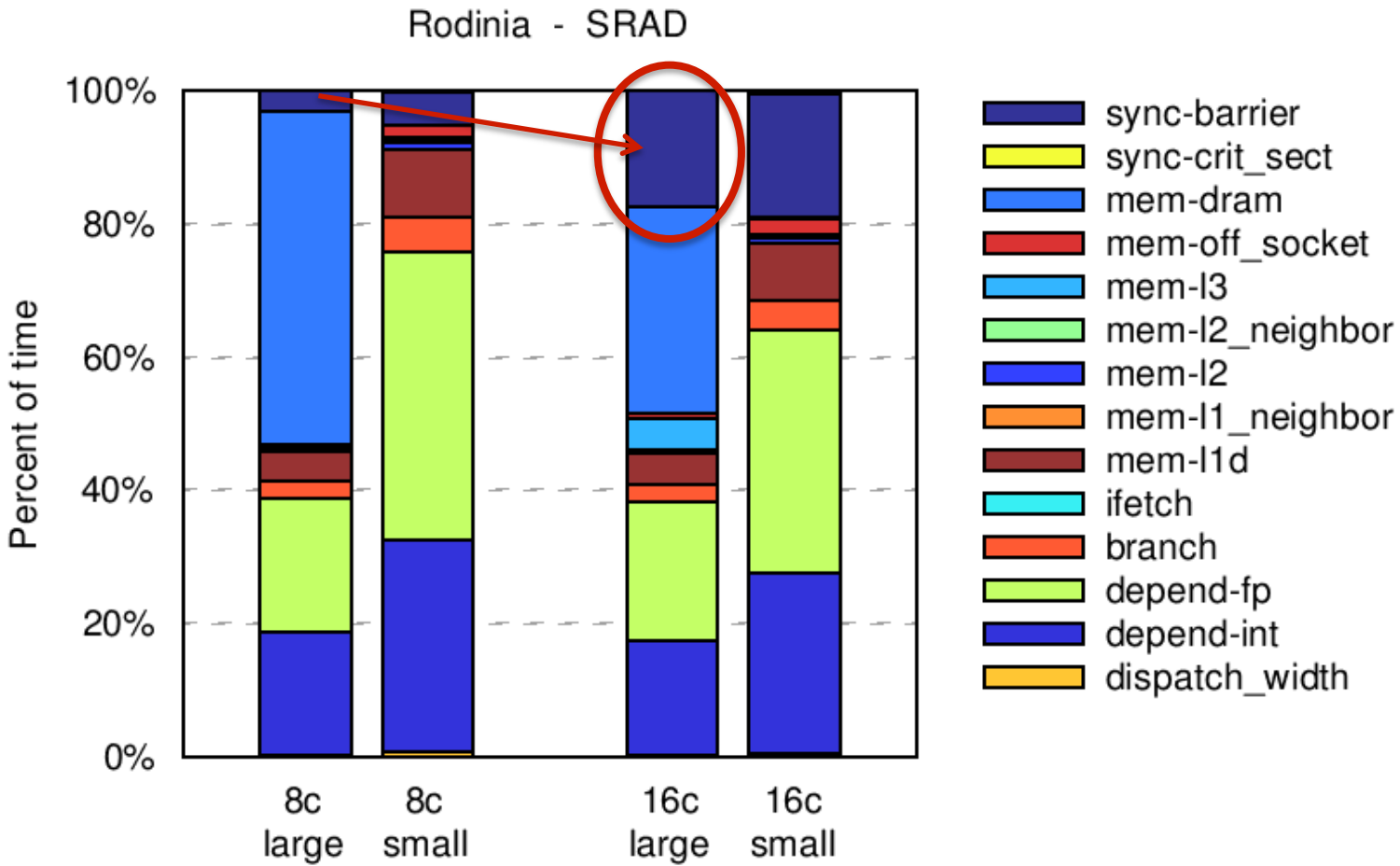
USING CYCLE STACKS TO EXPLAIN SCALING BEHAVIOR

- Scale input: application becomes DRAM bound

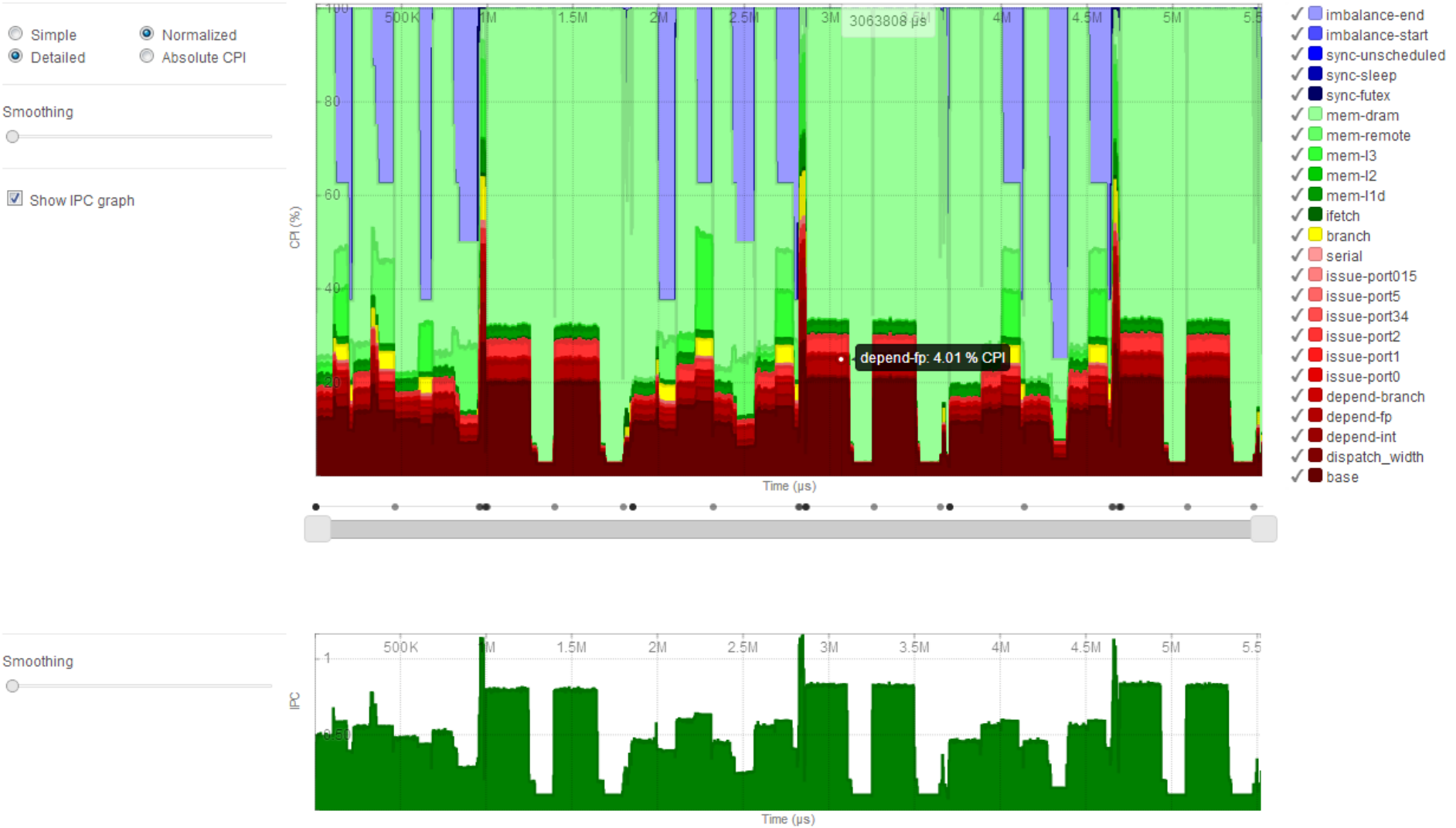


USING CYCLE STACKS TO EXPLAIN SCALING BEHAVIOR

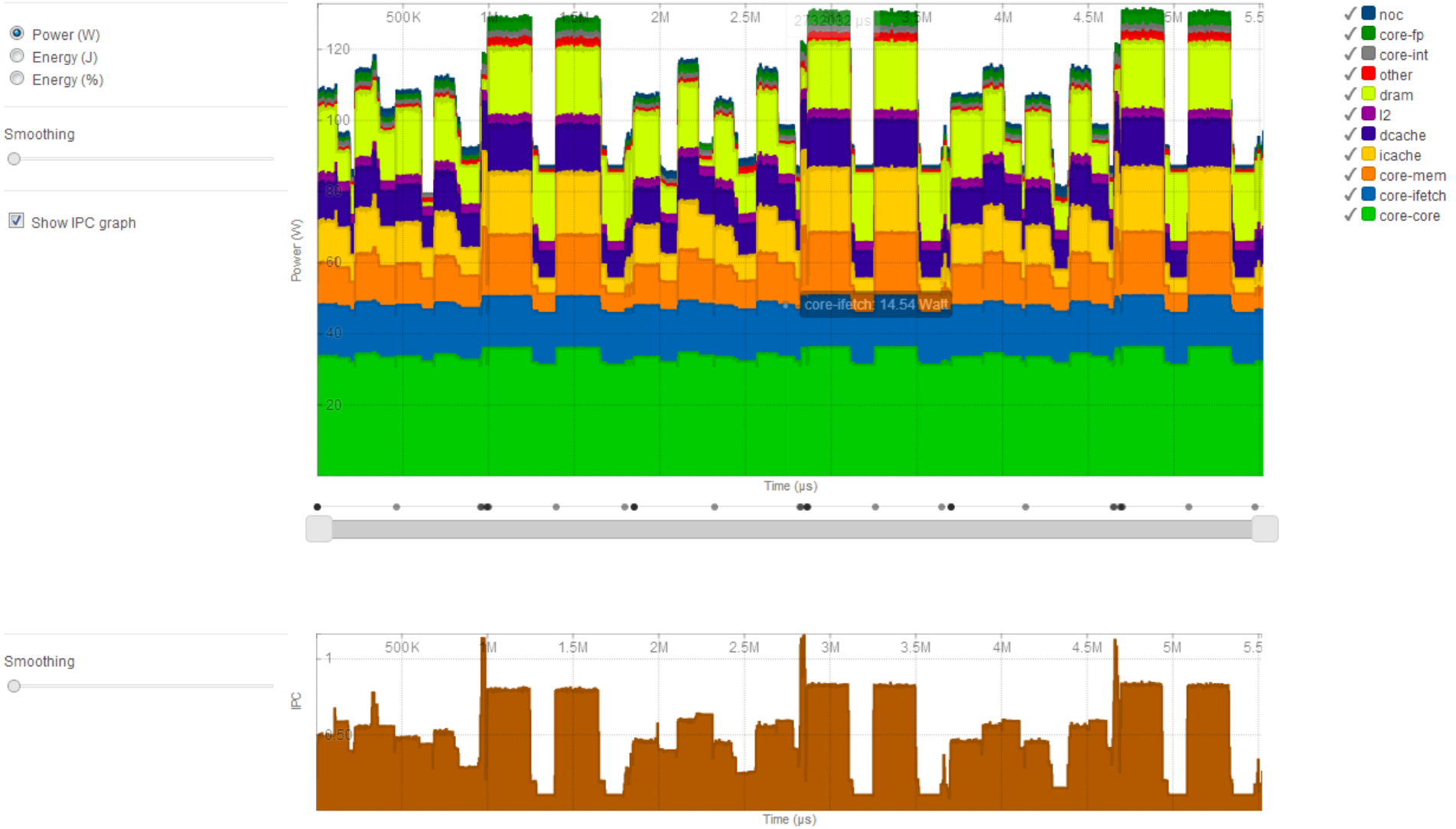
- Scale input: application becomes DRAM bound
- Scale core count: sync losses increase to 20%



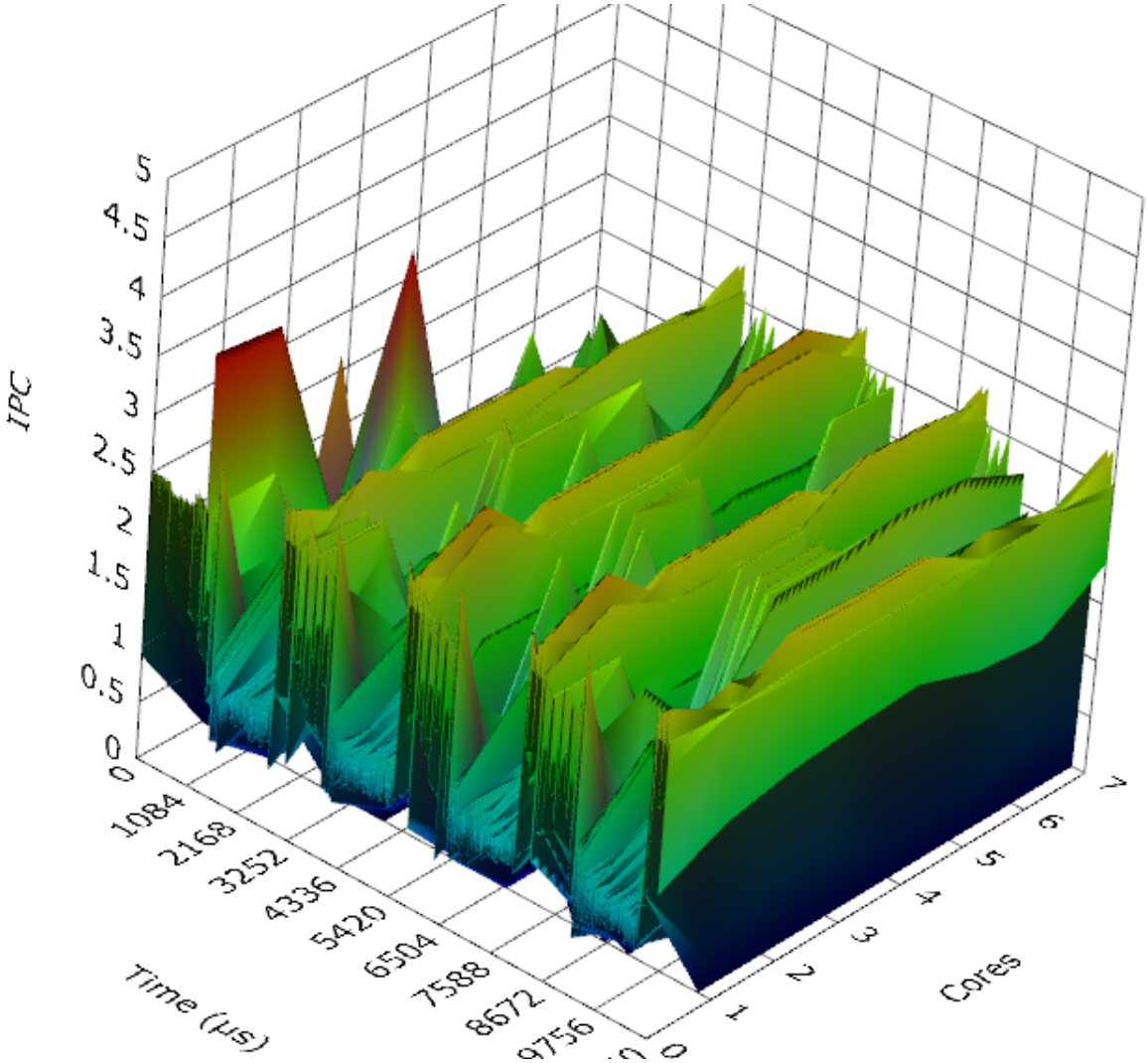
VIZ: CYCLES STACKS IN TIME



VIZ: ENERGY OUTPUT OVER TIME

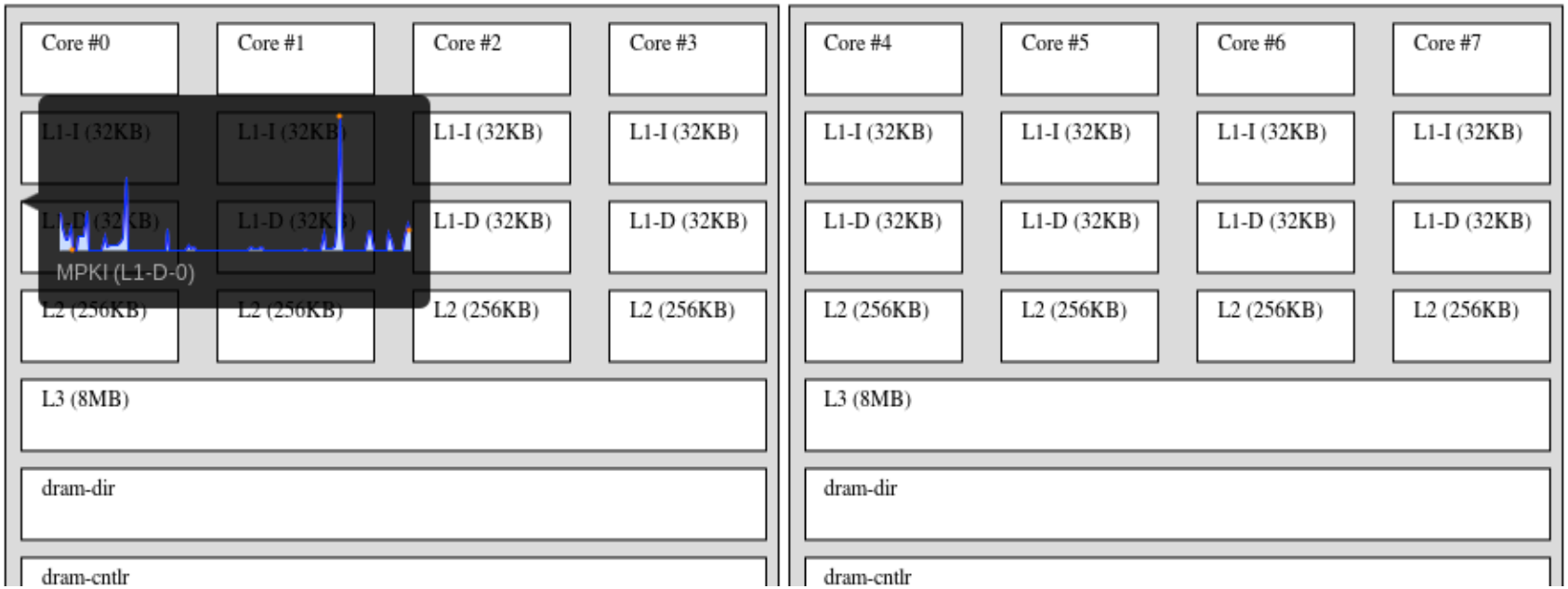


3D Visualization: IPC vs. Time vs. Core

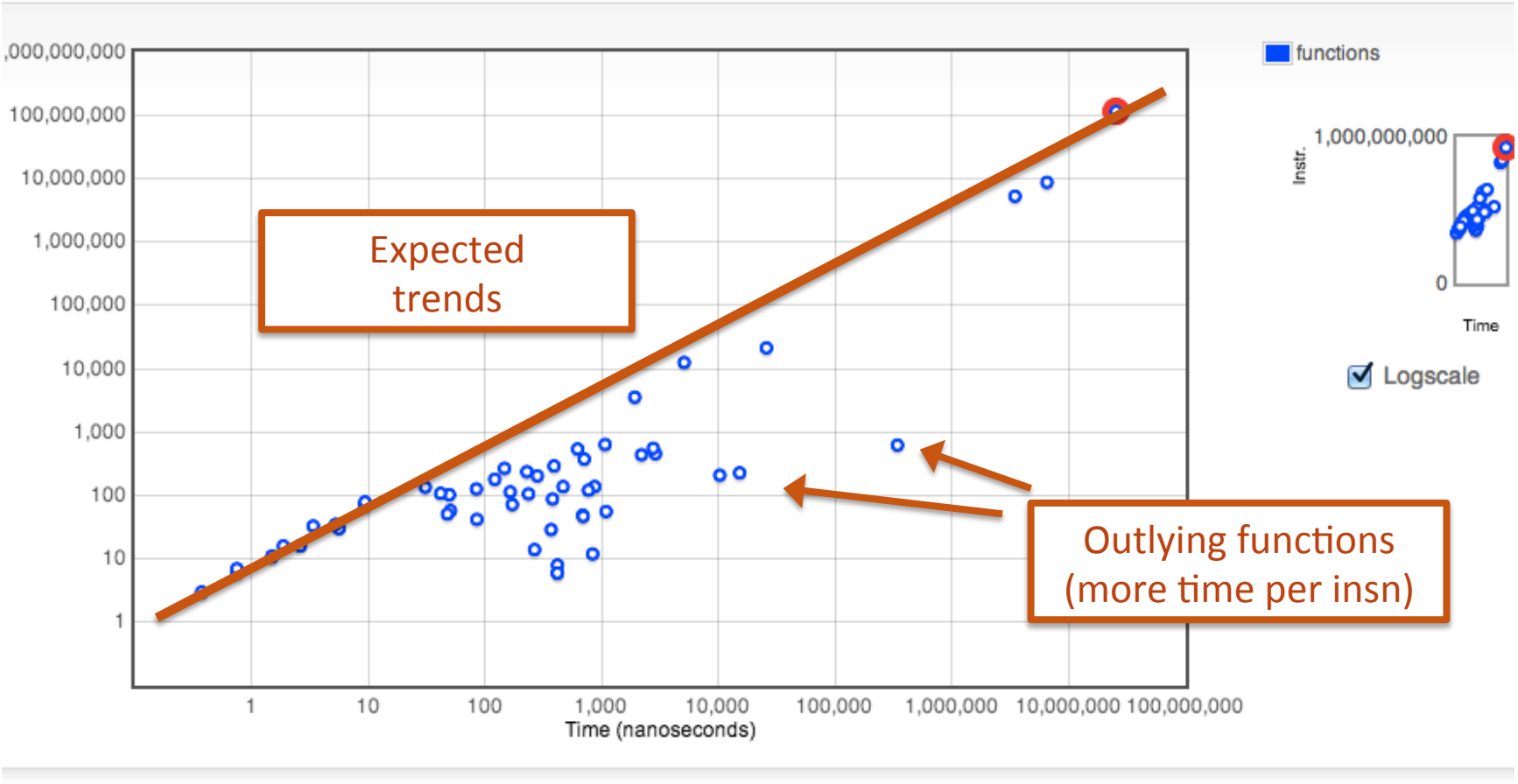


ARCHITECTURE TOPOLOGY VISUALIZATION

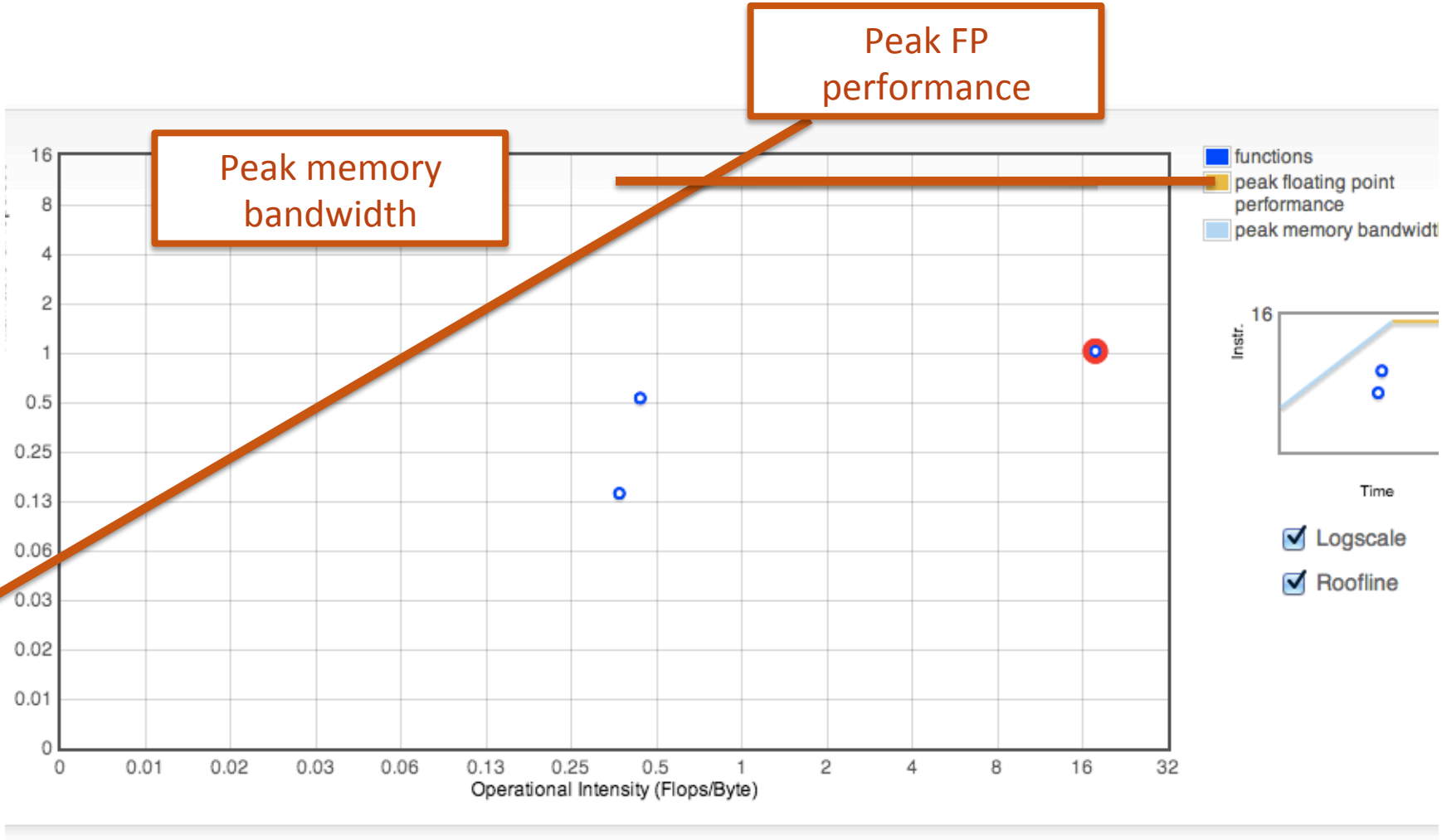
- System topology information
  - With IPC/MPKI/APKI stats for each component



Suggestions for Optimization: Instructions vs. Time Plot



Suggestions for Optimization: Roofline Model



S. Williams, A. Waterman, and D. A. Patterson, "Roofline: An insightful visual performance model for multicore architectures," Communications of the ACM, vol. 52, no. 4, pp. 65–76, Apr. 2009.

The Sniper Multi-Core Simulator
Power-Aware HW/SW Optimization

UNIVERSITEIT GENT

intel

ExaScience Lab
Intel Labs Europe
EXASCALE COMPUTING

http://www.snipersim.org
Saturday, February 1st, 2013
FOSDEM 2014 – HPC Devroom – Brussels, Brlgium

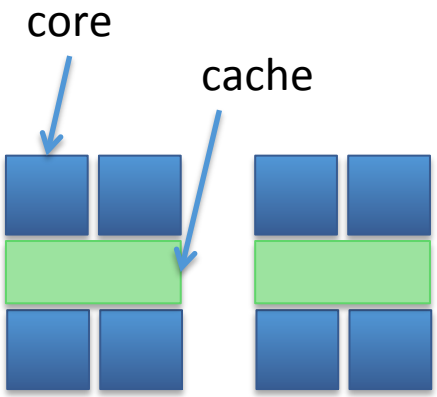
H/W UNDER/OVERSUBSCRIPTION

- Main idea:
  - For Xeon-Phi-style cores, cache performance is the biggest indicator of performance
- Each core has a private cache hierarchy
  - Private L1 + Private L2
    - Can access other L2s via coherency
- Each application has its own cache scaling characteristics
  - We see cache requirements both increasing, and decreasing per core
    - Increasing: globally shared working set
    - Decreasing: data is partitioned per core
- By controlling the core/thread count we can optimize placement

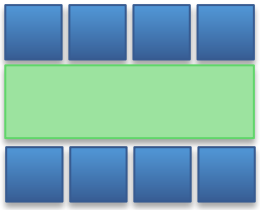
POWER-AWARE HW/SW CO-OPTIMIZATION

- Hooked up McPAT (Multi-Core Power, Area, Timing framework) to Sniper's output statistics
- Evaluate different architecture directions (45nm to 22nm) with near-constant area
- Compare performance, energy efficiency

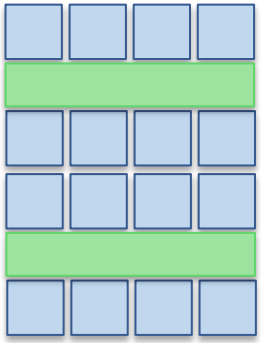
[Heirman et al., PACT 2012]



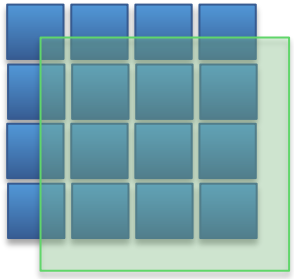
baseline: 2x quad-core



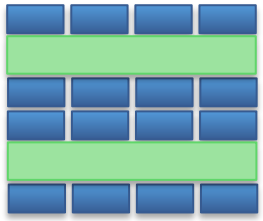
8 cores



16 slow cores



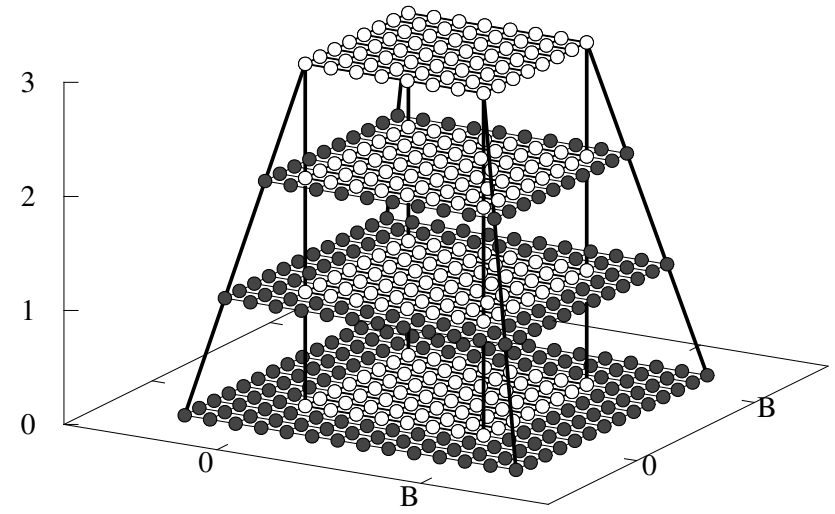
16 cores, no L3, stacked DRAM

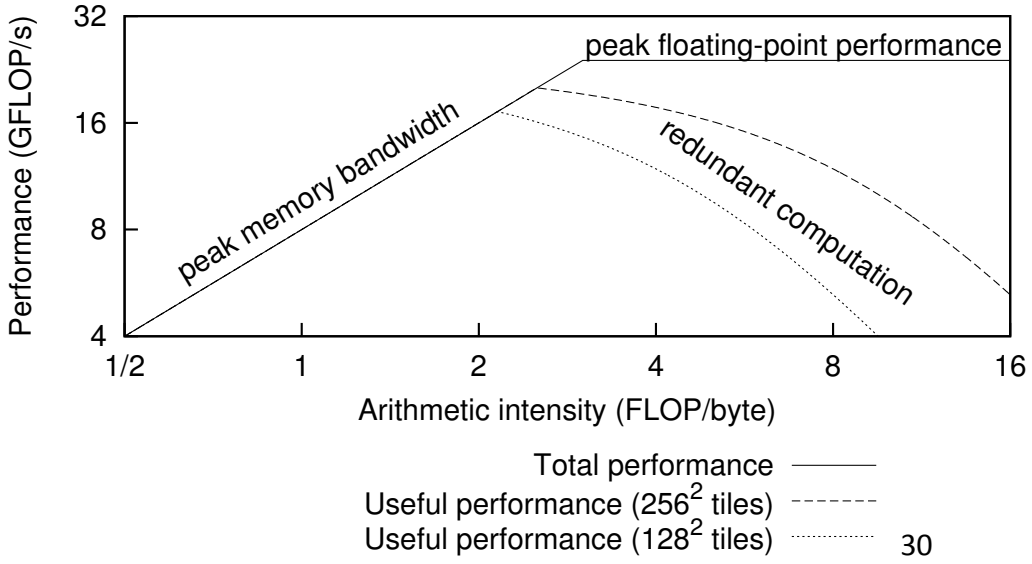


16 thin cores

POWER-AWARE HW/SW CO-OPTIMIZATION

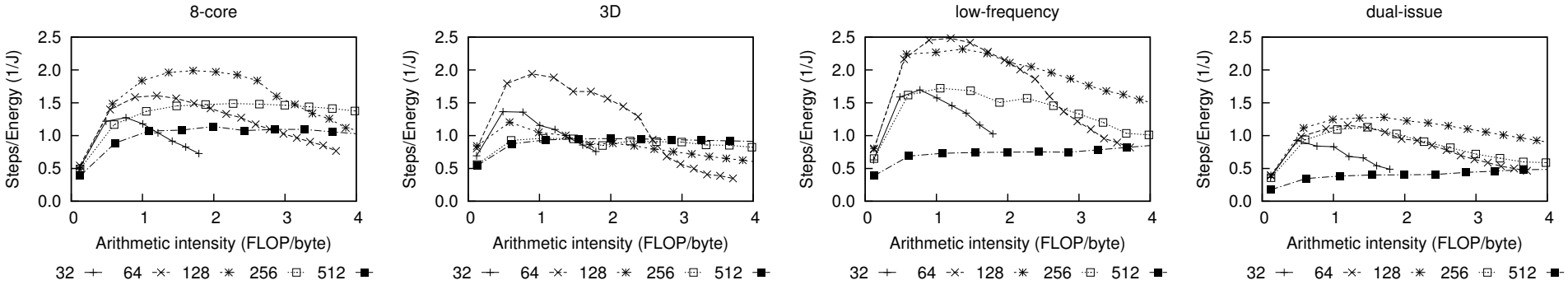
- Heat transfer: stencil on regular grid
  - Used in the ExaScience Lab as component of Space Weather modeling
  - Important kernel, part of Berkeley Dwarfs (structured grid)
- Improve memory locality: tiling over multiple time steps
  - Trade off locality with redundant computation
  - Optimum depends on relative cost (performance & energy) of computation, data transfer → *requires integrated simulator*





POWER-AWARE HW/SW CO-OPTIMIZATION

- Match tile size to L2 size, find optimum between locality and redundant work – depending on their (performance/ energy) cost
- Isolated optimization:
  - Fix HW architecture, explore SW parameters
  - Fix SW parameters, explore HW architecture
- Co-optimization yields 1.66x more performance, or 1.25x more energy efficiency, than isolated optimization



REFERENCES

- Sniper website
  - http://snipersim.org/
- Download
  - http://snipersim.org/w/Download
  - http://snipersim.org/w/Download_Benchmarks
- Getting started
  - http://snipersim.org/w/Getting_Started
- Questions?
  - http://groups.google.com/group/snipersim
  - http://snipersim.org/w/Frequently_Asked_Questions

HPC Node Performance and Power Simulation with the Sniper Multi-Core Simulator

Trevor E. Carlson,

Wim Heirman, Lieven Eeckhout

http://www.snipersim.org
Saturday, February 1st, 2014
FOSDEM 2014 – HPC Devroom – Brussels, Belgium