ModSim as Part of the ECP Hardware Evaluation Project

Approved for public release

Project PI: Si Hammond

David Donofrio. …many others

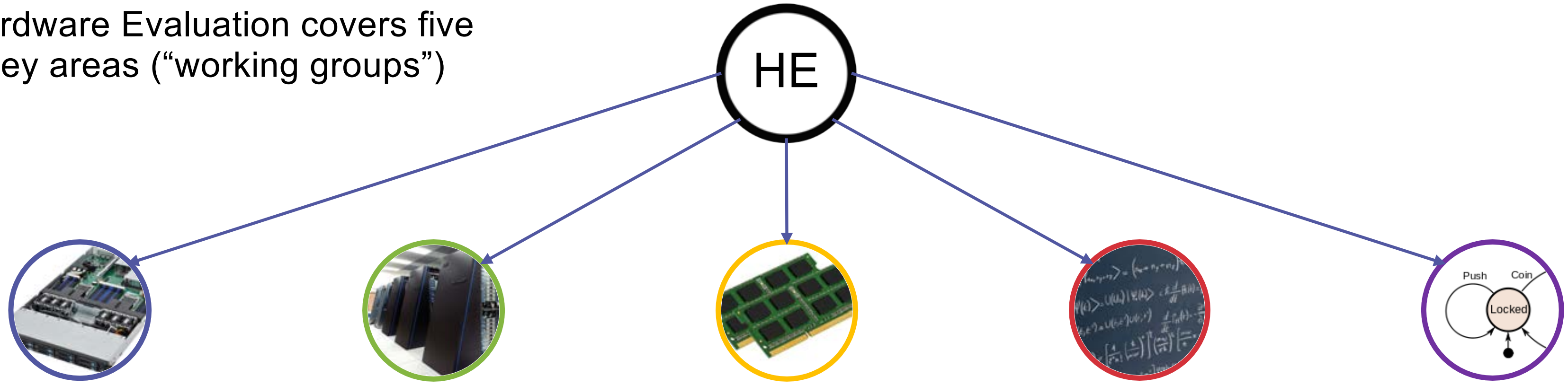
Hardware Evaluation – Bringing ModSim to ECP

- Hardware Evaluation brings together DOE's hardware research teams and low-level expertise
  - Covers seven DOE laboratories with strong investments in supercomputing

- Provide expertise to evaluate complex hardware analysis across vendor offerings
  - Each vendor provides detailed analysis of *their* system, but DOE needs analysis across architectures
  - Validation of vendor analysis
  - Often brings deep DOE application knowledge to studies which vendors do not have access to

Overview of Hardware Evaluation Project Capabilities

Hardware Evaluation covers five key areas ("working groups")



Node-Level Simulation

Processor, Pipelines, Threading, Caches, Coherency, Network-on-Chip, Network Interfaces

Interconnect Modeling

Network Topologies, Congestion, Quality-of-Service, Silicon Photonics

Memory Technologies

Memory Media, Parallelism, Controllers, Non-Volatile, Coherency, Scratchpads

Analytic Modeling

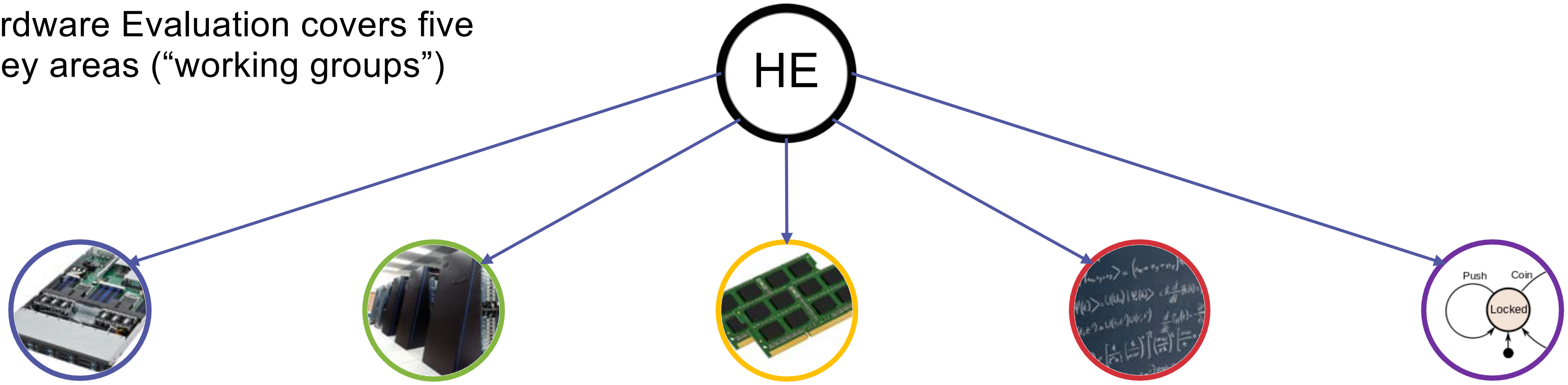
High-Level Architecture Balance, "Ops-to-Bytes", Instruction Mix, Branching, Vectorization

Abstract Machines and High-Level System Models

Outreach, Cross-ECP Communication, Non-NDA Models, High-Level Descriptions

Overview of Hardware Evaluation Project Capabilities

Hardware Evaluation covers five key areas ("working groups")



HE

Node-Level Simulation

David Donofrio
LBNL

Interconnect Modeling

Scott Hemmert
SNL

Memory Technologies

Maya Gokhale
LLNL

Analytic Modeling

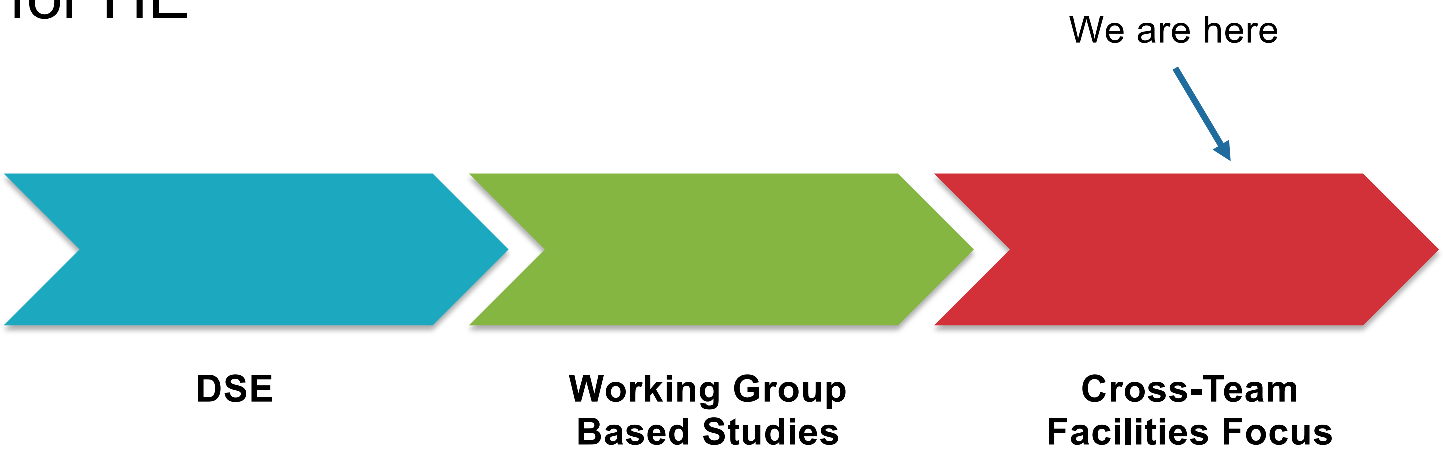
Scott Pakin
LANL

Abstract Machines and High-Level System Models

David Donofrio
LBNL

HE Team spans six DOE Labs

Timeline for HE



We are here

DSE

Working Group Based Studies

Cross-Team Facilities Focus

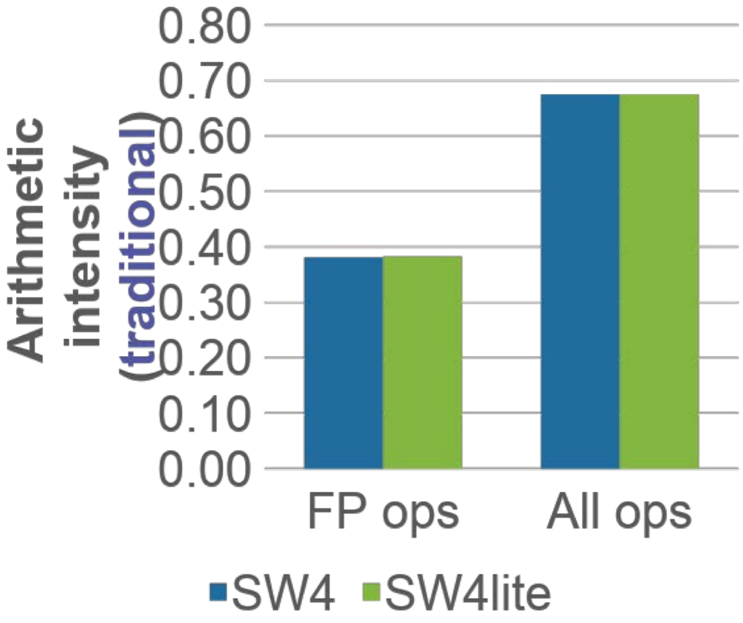
- Moving away from “Stove piped” groups
  - Focus on system level issues / challenges
  - Continue to provide guidance to DOE on future HW trends
- Facilities-based Studies
  - Driven to help support Exascale or post-Exascale technologies
  - Build on ECP Proxy Apps
  - Think about workflow, not just mini-apps and benchmarks

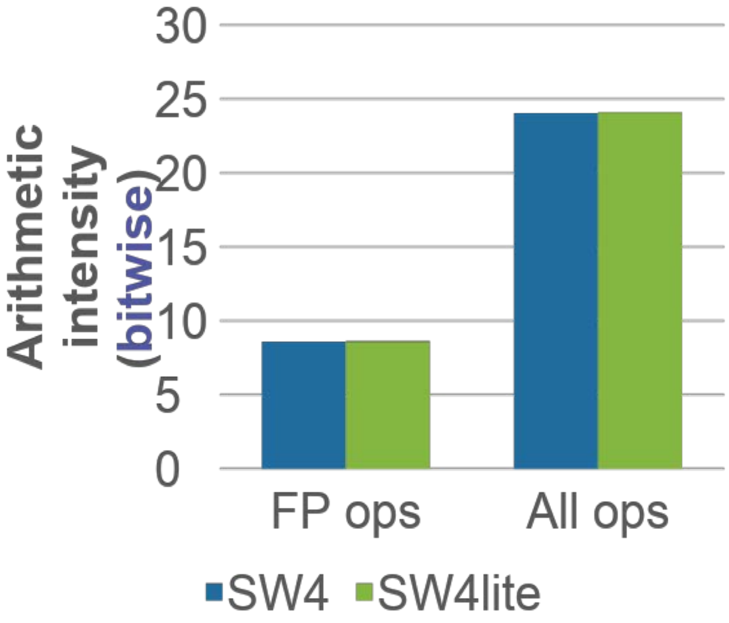
Example study…. A deep dive into SW4

SW4 and SW4lite Background

- SW4 is a seismic simulation code used by the EQSIM ECP project.
- Stand-alone SW4lite proxy application solves the elastic wave equation with limited seismic modeling capabilities.
- Current development focused on using the RAJA portability layer.
- 3-D heterogeneous material model with a curvilinear mesh near the free surface to represent the free surface boundary condition on a realistic topography
- Computational kernels
  - forcing function that simulates earthquake source
  - Stress calculation (RHS) that applies finite-difference stencils
  - Supergrid damping near far-field domain boundaries
  - Boundary conditions on free surface and far-field boundaries
  - Predictor-corrector procedure for updating displacements in time
- gaussianHill-rev.in input for SW4lite recommended by developers to represent performance of curvilinear kernels that take the most time in the larger problem.

Analyzing SW4 and SW4lite Computational Intensity with Byfl

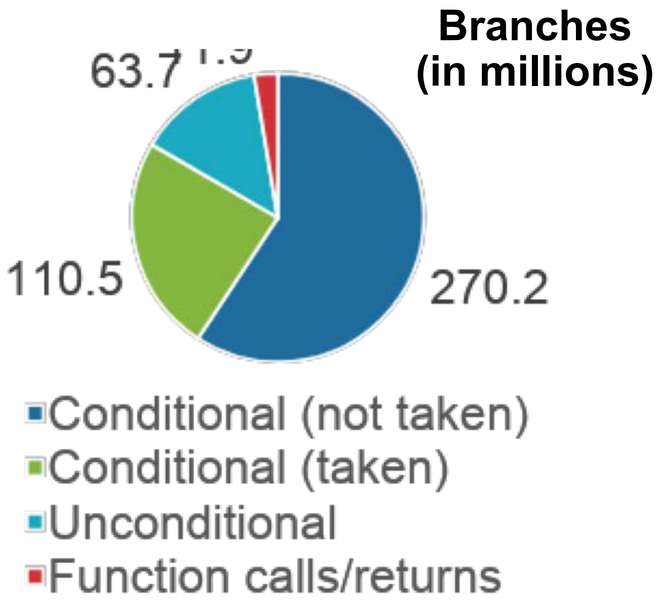




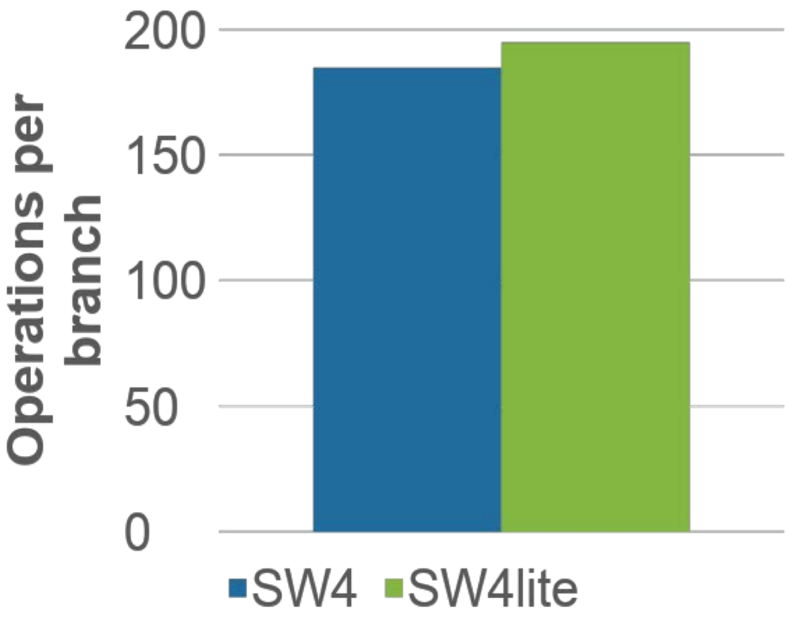
Initial Insight: High degree of memory-hierarchy traffic

… a closer look reveals that re-use better than expected, suggesting an emphasis on computer

Analyzing SW4 and SW4lite Branch Behavior with Byfl

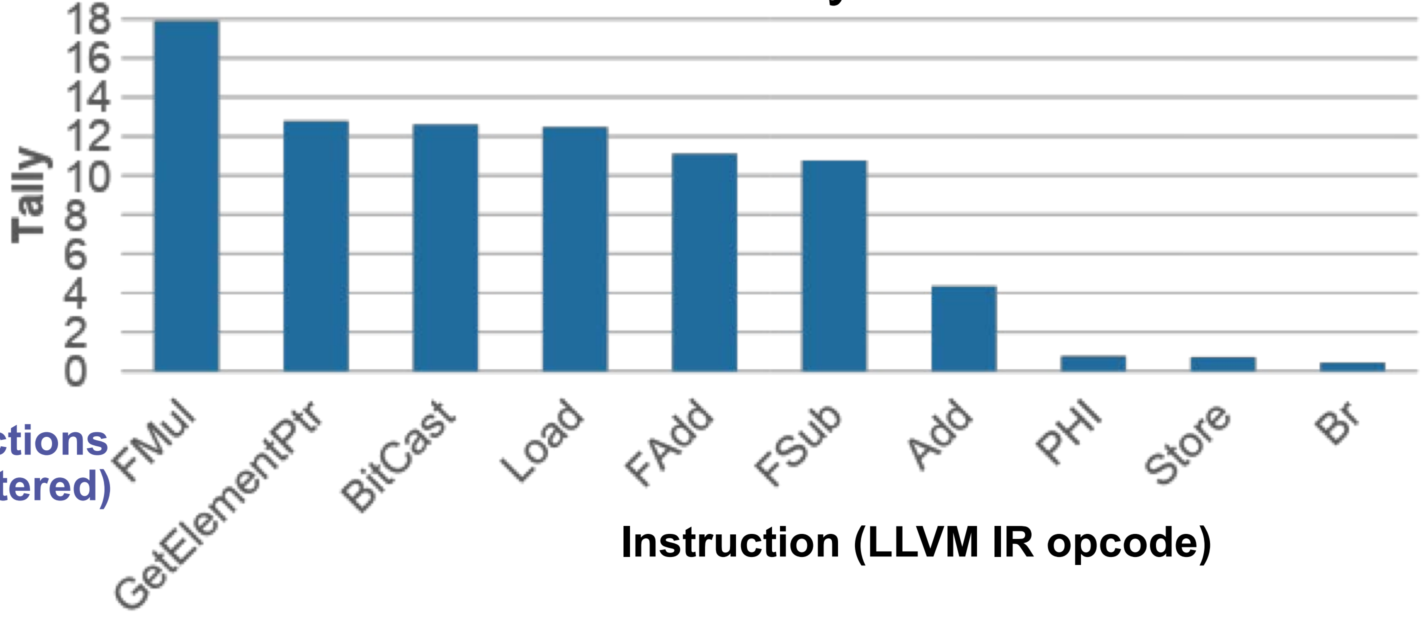


- Branch predictability (entropy) is 87%: highly predictable
- Good news!



- Very large average number of operations per branch
- SW4 able to take advantage of future hardware with proportionally high branch-misprediction penalties

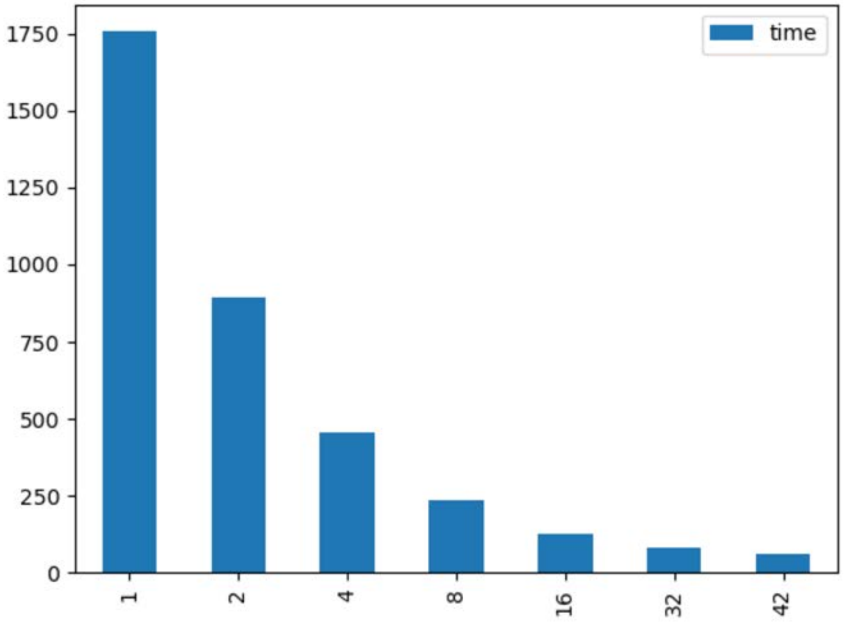
Analyzing SW4’s Instruction Mix with Byfl



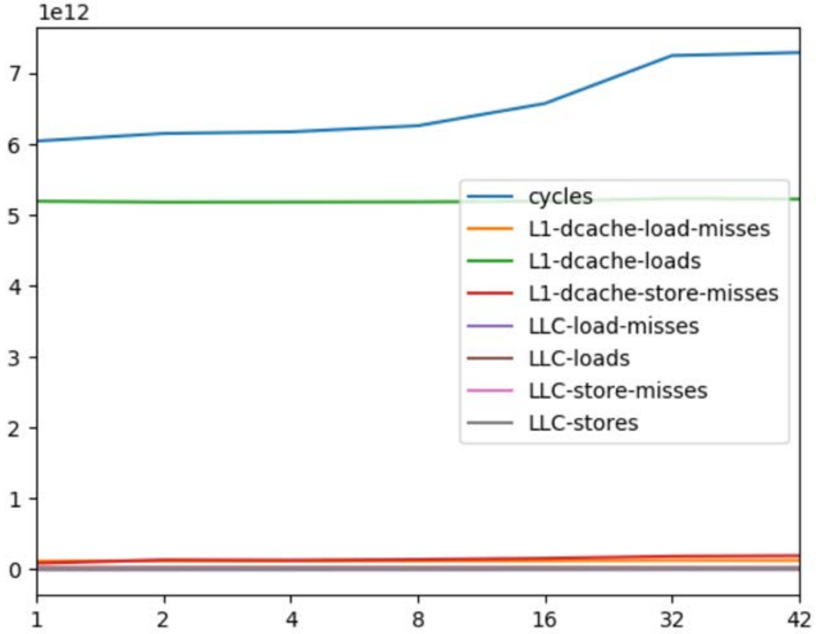
Top 10 instructions (of 37 encountered)

- Byfl uses LLVM’s internal representation as a canonical instruction set
  - Cleaner, more orthogonal, and with fewer special cases than a hardware instruction set
- Substantial number of operations (21%) are FP multiplies; indexing operations, loads, and FP add & subtract are also highly represented

Runtime for SW4Lite OpenMP – Utilizing PERF on Summit

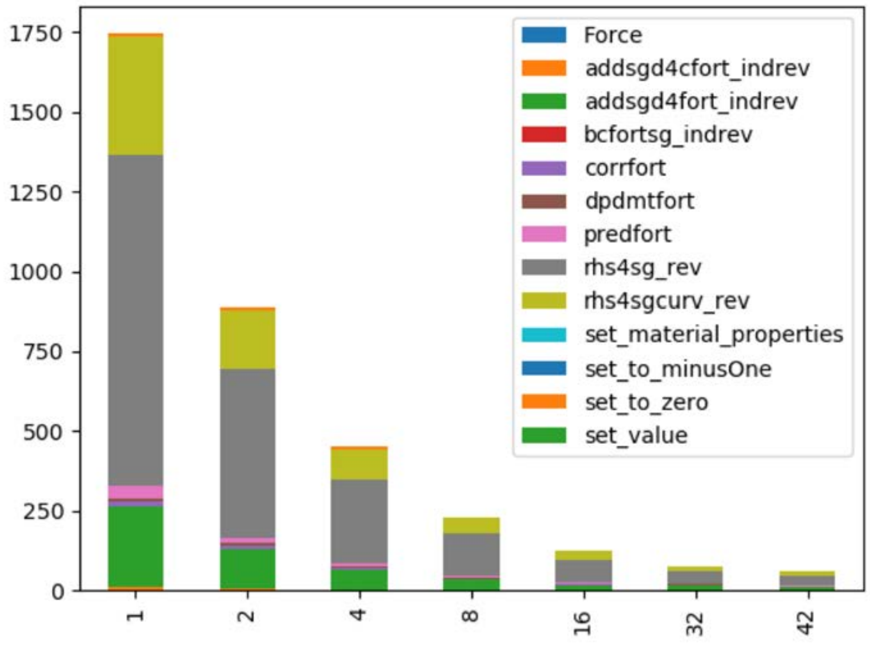


Time graph for SW4Lite with the gaussianHill-rev.in data set. Good scaling results across a single Summit node with OpenMP.



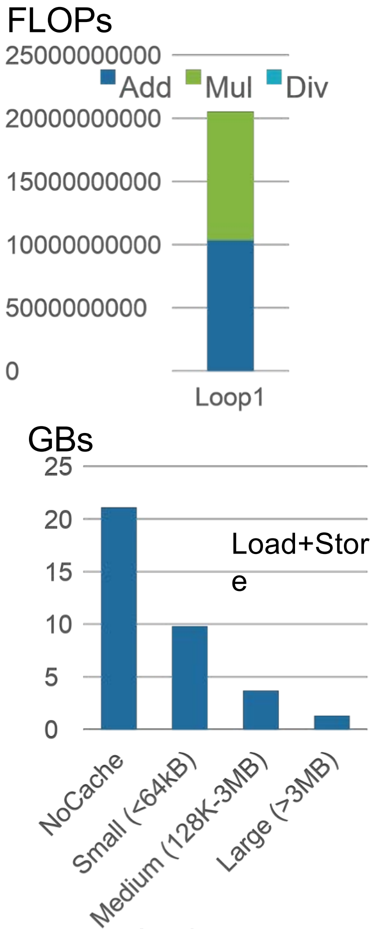
Performance counters using PERF for all threads. It showcases the memory hierarchy behavior in terms of misses. Behavior is similar across all threads.

Parallel Regions Breakdown per Function

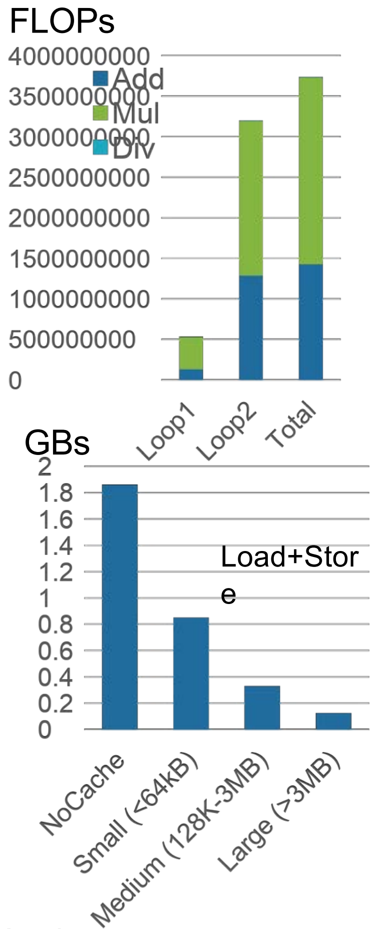


- Parallel regions timing per function. If a function has several parallel functions, their times are aggregated for the function.
- The top functions are rh4sg_rev, rhs4sgcurv_rev and addsgd4fort with the heaviest parallel region.
- **Purpose:** Identifies top functions for further study….

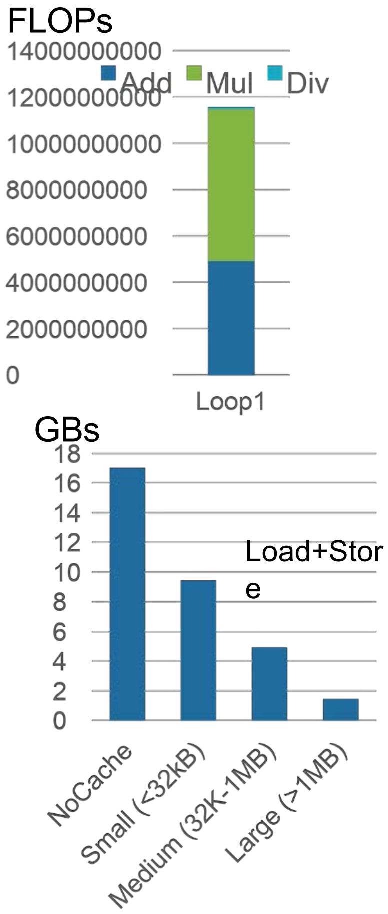
Compute/Memory Characteristics of Individual Kernels – Collected using ExaSAT



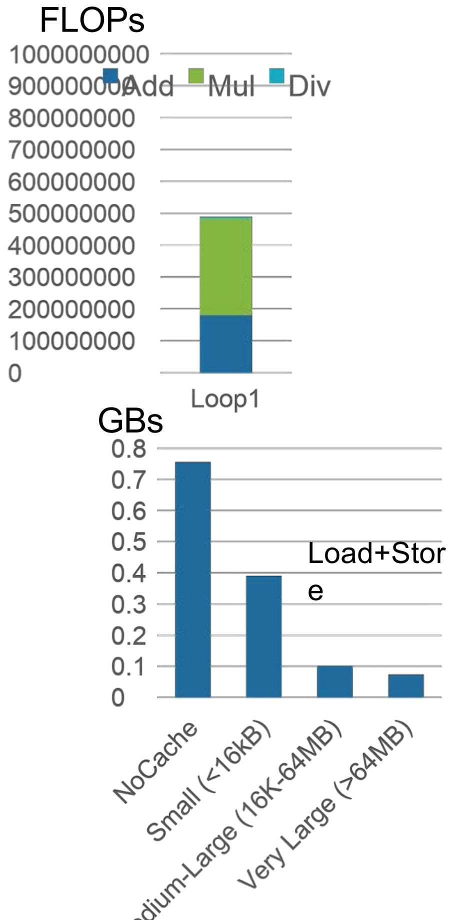
rhs4sg_rev



rhs4sgcurv_rev



addsgd4fort_indrev

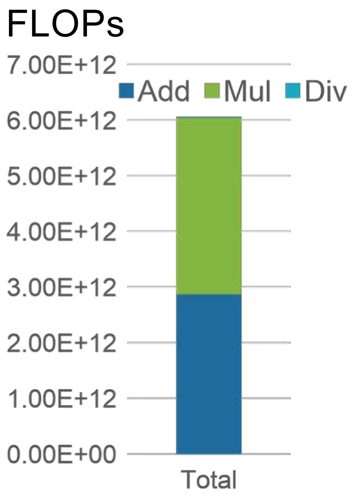


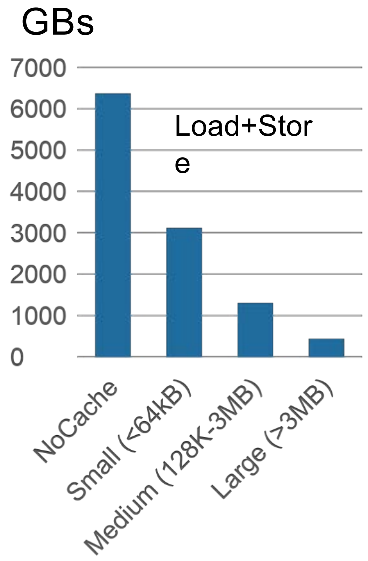
addsgd4cfort_indrev

EXASCALE COMPUTING PROJECT

Putting it All Together

- Input problem: gaussianHill-rev.in
- 200x rhs4sg_rev + 200x rhs4sgcurv_rev
- 100x addsgd4fort_indrev + 100x addsgd4cfort_indrev
- Application characteristics:
  - Add and multiply are key operations
  - Depending on the cache capacity, the application can be compute bound or cache-bandwidth or memory-bandwidth bound
  - On multi and many core processors, cores may simultaneously work on separate data sets. These core compete for memory bandwidth
  - Data tiling is an important optimization that reduces the data working set size so that the working sets of all processor cores fit on cache

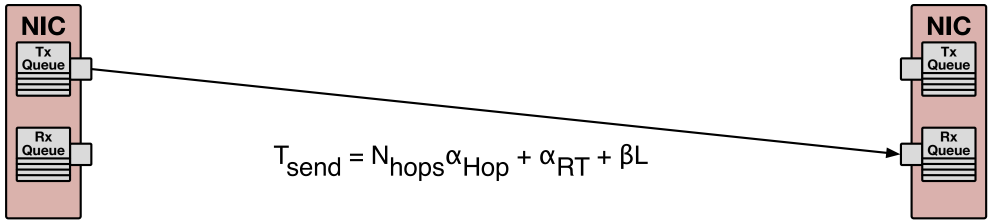




HE Models Not Limited to Node Level Questions

Range of efficient models to identify trends, quantify uncertainties in results

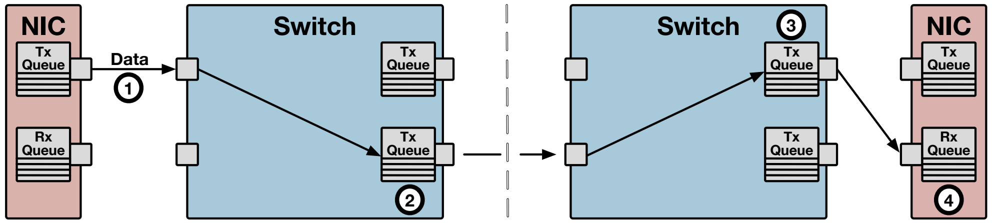
Lightweight simulation using analytic model



$$T_{send} = N_{hops}\alpha_{Hop} + \alpha_{RT} + \beta L$$

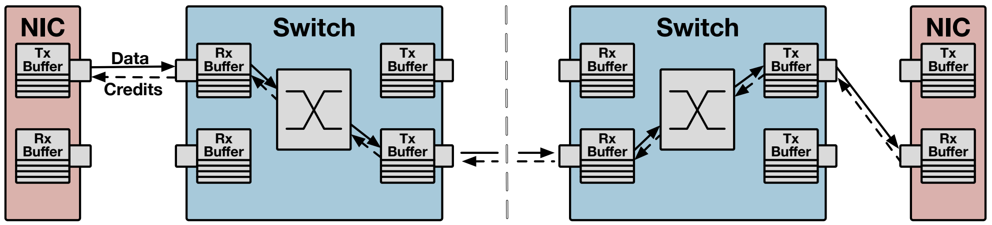
- Establishes *optimistic* upper-bound on performance in absence of contention
- Useful in validating software stack models for traffic patterns without contention
- Denoted LogP

Packet-based with simple congestion modeling



- Establish baseline performance of different routing/congestion control strategies
- Efficient execution, agnostic to router flow control details
- Denote SCULPIN

Packet-based with arbitration



- Best accuracy by tuning arbitration and token flow-control performance
- More complex and expensive
- PISCES, Merlin, CODES

What’s next for HE?

Focus on future systems

- Accelerators continue to drive performance
  - Prominent accelerators are currently GPUs
  - How can ModSim provide insights?
- Hardware evaluation team is focused on system level questions, two examples:
- “What important DOE application, benchmarks or kernels don’t work well on GPU accelerators?”
  - Utilize existing analytical modeling tools to gain deep insights into ECP Proxy Apps
  - Are there fundamental architectural features limiting performance?
  - Gather insights on existing architectures
- “When does coherency between the host and accelerator matter for application performance?”
  - Looking beyond GPUs, future nodes will include accelerators, what is the impact of various data movement methods on performance
  - Bring together existing memory modeling tools and gem5-based node-modeling tools
  - Model future architectures to provide insights into tradeoffs

Summary

- Bringing together suites of ModSim tools enables deeper insights
- Architectural insights from ECP HE ModSim activities allows vendors to prioritize features
  - Enables analysis *across* architectures rather than a single vendor
- Impacts of accelerators and increasing heterogeneity on application performance can be quantified by HE tool suite