Los Alamos
NATIONAL LABORATORY

Modeling and Simulation in the Exascale Computing Project

Scott Pakin

10 August 2022

LA-UR-22-27977

Outline

- The Exascale Computing Project
- Modeling and simulation
- Retrospection

The Exascale Computing Project

EXASCALE COMPUTING PROJECT

- Ensure that the first US exascale supercomputers will be immediately productive
  - Provide applications, software, and user support, not hardware procurements
- Very large US Department of Energy initiative
  - US$1.8 billion spread over the years 2016–2023
  - Involves DOE labs, academia, and industry
- Goals
  - Advancing scientific discovery
  - Strengthening national security
  - Improving industrial competitiveness

Modeling and Simulation’s Place in ECP

Exascale Computing Project
2.0
Kothe (ORNL)

Project Management 2.1 Boudwin (ORNL)	Application Development 2.2 Siegel (ANL)	Software Technology 2.3 Heroux (SNL)	Hardware and Integration 2.4 Antypas (LLNL)

Modeling and Simulation's Place in ECP

- **Exascale Computing Project 2.0 Kothe (ORNL)**
  - **Project Management 2.1 Boudwin (ORNL)**
    - Project Planning and Management 2.1.1 Collins (ORNL)
    - Project Controls and Risk Management 2.1.2 Blair (ORNL)
    - Business Management 2.1.3 Milburn (ORNL)
    - Procurement Management 2.1.4 Keck (ORNL)
    - Information Technology and Quality Management 2.1.5 Wilson (ORNL)
    - Communications and Outreach 2.1.6 Bernhardt (ORNL)
  - **Application Development 2.2 Siegel (ANL)**
    - Chemistry and Materials Applications 2.2.1 Deslippe (LBL)
    - Energy Applications 2.2.2 Evans (ORNL)
    - Earth and Space Science Applications 2.2.3 Martin (LBL)
    - Data Analytics and Optimization Applications 2.2.4 Hart (SNL)
    - National Security Applications 2.2.5 Francois (LANL)
    - Co-Design 2.2.6 Germann (LANL)
  - **Software Technology 2.3 Heroux (SNL)**
    - Programming Models and Runtimes 2.3.1 Thakur (ANL)
    - Development Tools 2.3.2 Vetter (ORNL)
    - Mathematical Libraries 2.3.3 Li (LBL)
    - Data and Visualization 2.3.4 Ahrens (LANL)
    - Software Ecosystem and Delivery 2.3.5 Munson (ANL)
    - NNSA Software Technologies 2.3.6 Mohror (LLNL)
  - **Hardware and Integration 2.4 Antypas (LLNL)**
    - PathForward 2.4.1 de Supinski (LLNL)
    - Hardware Evaluation 2.4.2 Pakin (LANL) ← **ModSim**
    - Application Integration at Facilities 2.4.3 Parker (ANL)
    - Software Deployment at Facilities 2.4.4 Adamson (ORNL)
    - Facility Resource Utilization 2.4.5 Siddabathuni (ANL)
    - Training and Productivity 2.4.6 Barker (ORNL)

Why Include Modeling and Simulation in ECP?

- Forecast performance (and performance bottlenecks) to *future* architectures
  - Complement efforts by Application Development teams to analyze and improve performance on *current* platforms
  - Highlight opportunities and challenges for scientific applications
- Inform post-exascale system design
  - Identify features that could help or hinder the performance of ECP applications
- Provide deeper understanding of observed performance on current exascale testbeds
- Ensure modeling and simulation efforts emphasize applications and architectures of interest to DOE

ECP Modeling and Simulation Structure

Total of ~40 people and ~1% of ECP's budget

Hardware Evaluation
Scott Pakin, LANL

Memory Technologies
Maya Gokhale, LLNL

Simulate/analyze alternative memory types and configurations

Memory

Analytical Modeling and Node Simulation
Sam Williams, LBNL

Rapidly predict/explain application performance, even on future hardware, with particular focus on on-node performance

Node/system

Interconnect Simulation
Scott Hemmert, SNL

Simulate extreme-scale networks to estimate relative performance of future networks

Network

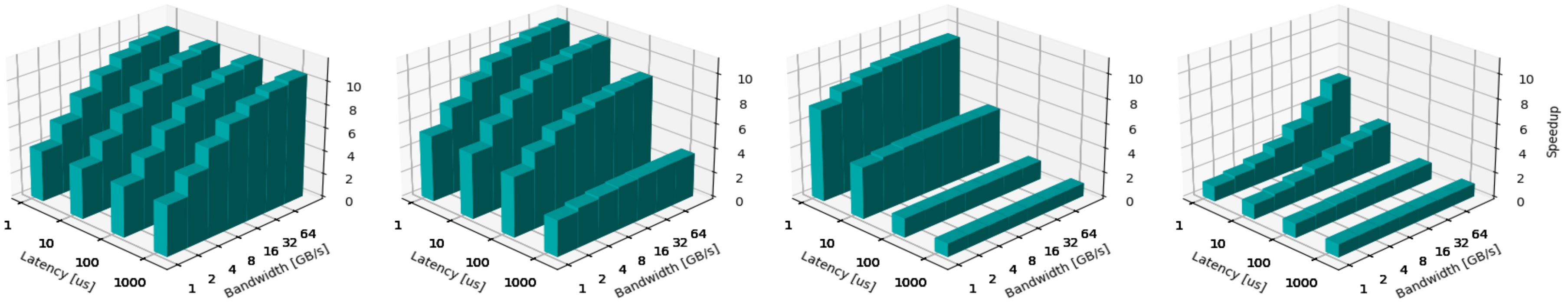
Outline

- The Exascale Computing Project
- Modeling and simulation
- Retrospection

Impact of Tighter CPU-GPU Integration

- What would be the effect of 10x faster communication between the CPU and GPU?
  - On-die versus over an I/O bus
- Baseline: 100μs kernel-launch latency and 8–16 GB/s PCIe bandwidth, including software overheads

Impact of Tighter CPU-GPU Integration (cont.)

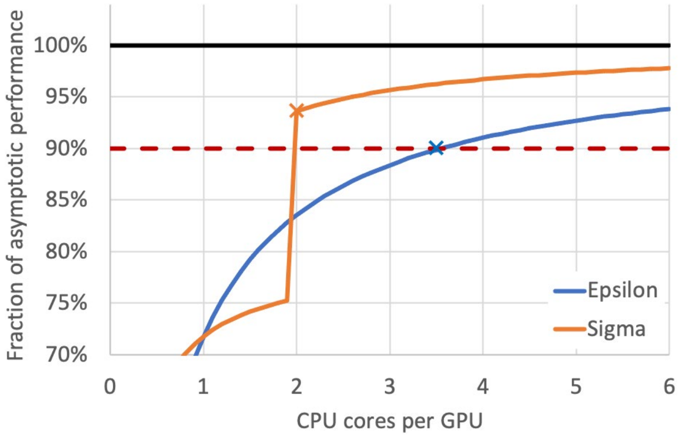


HACC LAMMPS OpenMC Rodinia NW

- **Findings**
  - Order-of-magnitude improvements in CPU-GPU integration will likely yield only a 30% increase in application-level performance relative to baseline (100μs, 16 GB/s) for most applications
  - Outliers approaching 2x
  - Advantage: No substantive application changes

Ratio of CPUs to GPUs

- Given limited chip area, what CPU:GPU ratio gives the best performance?
- Limit scope of study to number of Skylake-equivalent CPUs to V100-equivalent GPUs
- Find ratio that achieves 90% of potential application performance
  - Arbitrary but caps exponential gains from OpenMP, MPI GPU sharing, and communication artifacts
- **Findings**
  - The majority of applications studied require only one or two CPU cores per GPU
  - Makes practical single-socket integrated solutions

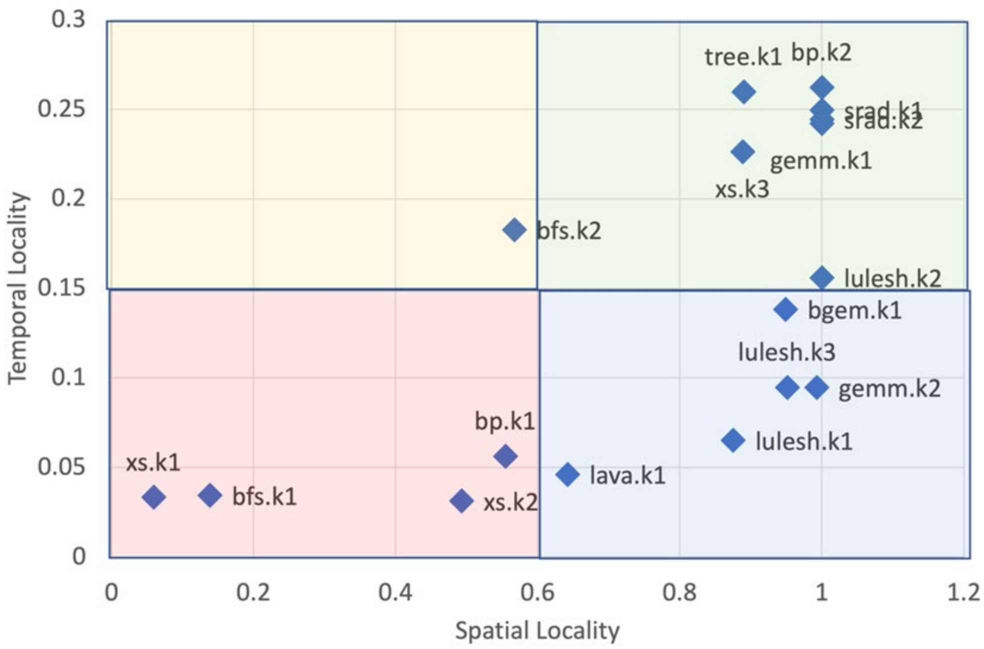


BerkeleyGW, Epsilon and Sigma modules

Los Alamos
NATIONAL LABORATORY

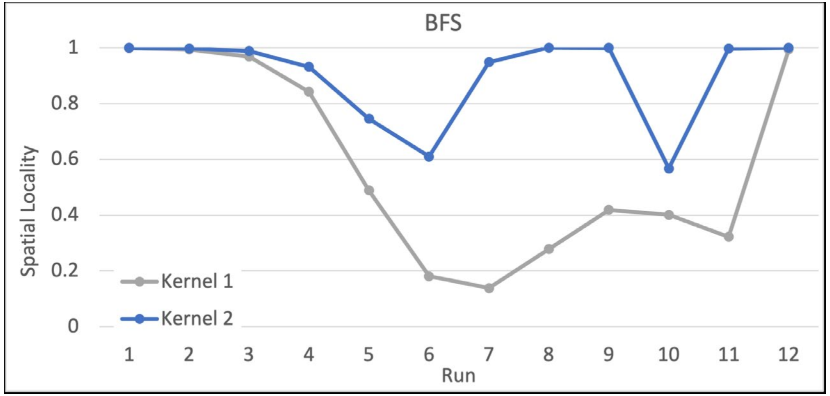
Memory-Locality Analysis

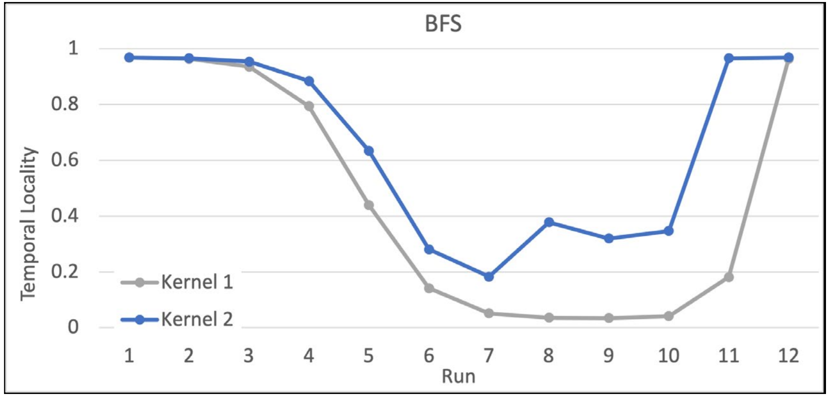
- MemLeap tool: Analyze spatial and temporal locality in GPU kernels
- Built atop NVIDIA's NVBit binary-instrumentation framework
- Captures accessed memory addresses by each thread in each warp then injects analysis code to measure architecture-independent locality metrics



Memory-Locality Analysis (cont.)

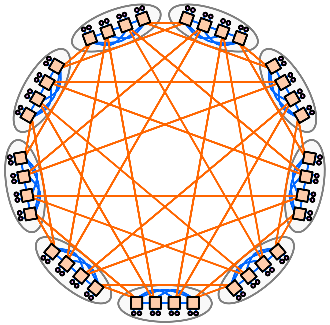
- **Finding**: Spatial and temporal locality can vary substantially even across multiple invocations of the same kernel (on different inputs)
- The following graphs plot spatial locality (left) and temporal locality (right) of 12 iterations two highly input-dependent kernels from a breadth-first search
- Implications for how well a program can exploit the memory subsystem



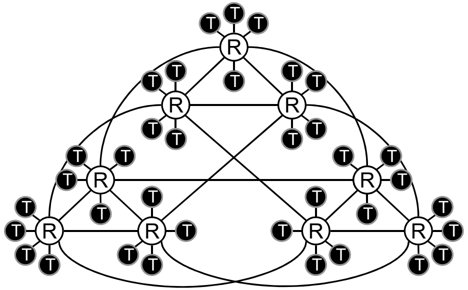


Performance Sensitivity to Failed Network Links

- All modern networks can route around failed links
- Comes at a performance cost
  - Less aggregate bandwidth is available
  - Previously independent flows may contend for certain links
  - More router hops may be required for a message to get to its destination
- Studied two network topologies: Dragonfly and HyperX
  - Both simulated at 30,000+ nodes
- Studied three communication patterns: 27-point stencil, KBA sweeps, and LQCD communication



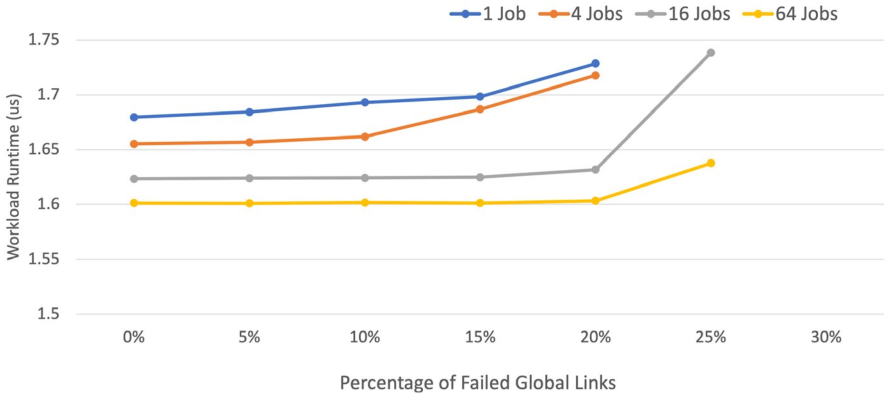
Dragonfly topology
Image credit: Wikipedia



HyperX topology
Image credit: Ahn et al., SC'09

Performance Sensitivity to Failed Network Links (cont.)

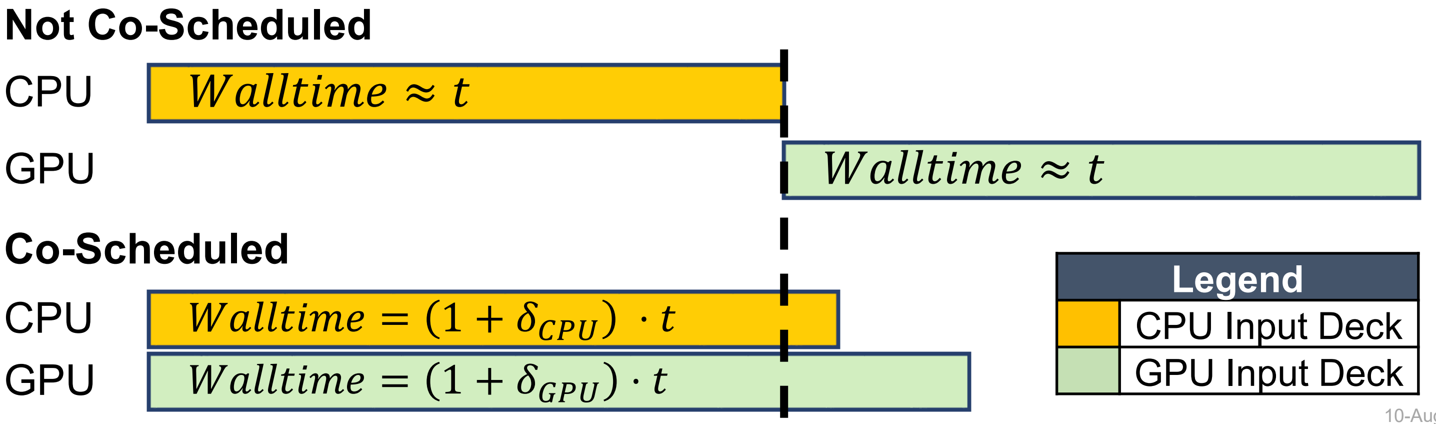
- **Findings**
  - Only the 27-point stencil showed sensitivity to failed links
  - HyperX is more sensitive to failed links than Dragonfly (may be based on some pessimistic assumptions, though)
  - Systems can withstand at least 5% link failure before the throughput of the machine is noticeably affected, even for the 27-point stencil
  - Larger jobs are more sensitive to link failures than smaller jobs
  - Lower global/bisection bandwidth increases sensitivity to failed links



27-point stencil on a half-bisection-bandwidth HyperX

Co-scheduling CPU and GPU Jobs

- Architectural model
  - Integrated CPUs and GPUs sharing high-bandwidth memory
- Hypothesis
  - Neither CPU codes nor GPU codes alone can saturate HBM bandwidth
  - Co-scheduling CPU-intensive applications and GPU-intensive applications can increase throughput



Co-scheduling CPU and GPU Jobs (cont.)

- Slowdown calculated by normalizing to standalone run times
- CPU input decks ran on 36 ranks, and GPU input decks ran on 2 GPUs
- Applications experienced 1–52% slowdown when co-scheduled

	GPU: Laghos		Quicksilver		PeleC		SW4Lite		Castro		WarpX	
CPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
Laghos	1.18	1.01	1.19	1.31	1.21	1.20	1.18	1.04	1.24	1.15	1.20	1.24
Quicksilver	1.39	1.01	1.40	1.31	1.41	1.21	1.40	1.04	1.42	1.14	1.39	1.25
PeleC	1.38	1.01	1.40	1.32	1.42	1.21	1.39	1.05	1.43	1.16	1.40	1.27
SW4Lite	1.51	1.01	1.52	1.33	1.51	1.23	1.51	1.06	1.51	1.18	1.52	1.27
Castro	1.17	1.01	1.18	1.33	1.19	1.26	1.17	1.07	1.19	1.18	1.18	1.29
WarpX	1.31	1.01	1.33	1.32	1.36	1.23	1.32	1.06	1.39	1.16	1.32	1.26

Example: Sw4Lite on the CPU co-scheduled with WarpX on the GPU.

SW4Lite slowed by 52% and WarpX slowed by 27%.

Legend		
green	0.00	1.10
light green	1.10	1.20
yellow	1.20	1.30
orange	1.30	1.40
red	1.40	1.50
dark red	1.50	5.00

Outline

- The Exascale Computing Project
- Modeling and simulation
- Retrospection

Successes

- Good potential for identifying future performance opportunities and bottlenecks
  - Follow hardware trends and use ModSim to analyze application impact
- Cross-laboratory teams support different approaches to answering technical questions
- Lots of interesting analyses and findings
  - Starting to form a picture of how scientific applications may perform on future hardware
  - Helpful that ECP includes a proxy-applications component
- Tool development/enhancement
  - Almost all of which is now open-source

➔ Having smart people work on challenging problems generally yields positive outcomes

Struggles

- Better integration across the Memory, Node, and Network teams would have been nice
  - Ideal would be to have everyone approach the same performance question from different angles and using different tools and methodologies
  - In practice, cycle-accurate simulators and analytical performance models, for example, handle different application scales, work at different levels of accuracy, and answer different performance questions
  - Some challenges herding cats played a role, too

Struggles (cont.)

- ModSim analyses not widely valued by the application teams
  - "There's no point in our altering our applications based on your predictions. We optimize for today's platforms, and if anything changes, we simply re-optimize for that."
  - Similar story for ModSim explanations of current performance: hardware counters are ground truth; everything else is based on potentially untrustworthy assumptions
  - A bit more appreciation came from the DOE supercomputing facilities in the context of procurement decisions and hardware configuration

Summary

- The Exascale Computing Project has included a ModSim component, called "Hardware Evaluation", for a number of years
- Pull together expertise in memory, node, and network modeling and simulation from across the DOE complex
- Examine potential impacts of hardware trends on ECP application performance
- Successes include analyses and recommendations based on analytical modeling and various types of simulations
- Struggles include integration across memory/node/network components and garnering trust in our findings

➔ Important for DOE to include ModSim in large research efforts because there is always a "next" supercomputer for applications to target