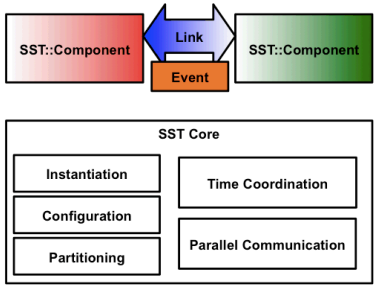
SAND2017-6128C



Sandia's ARM–centric Co-Design Strategy

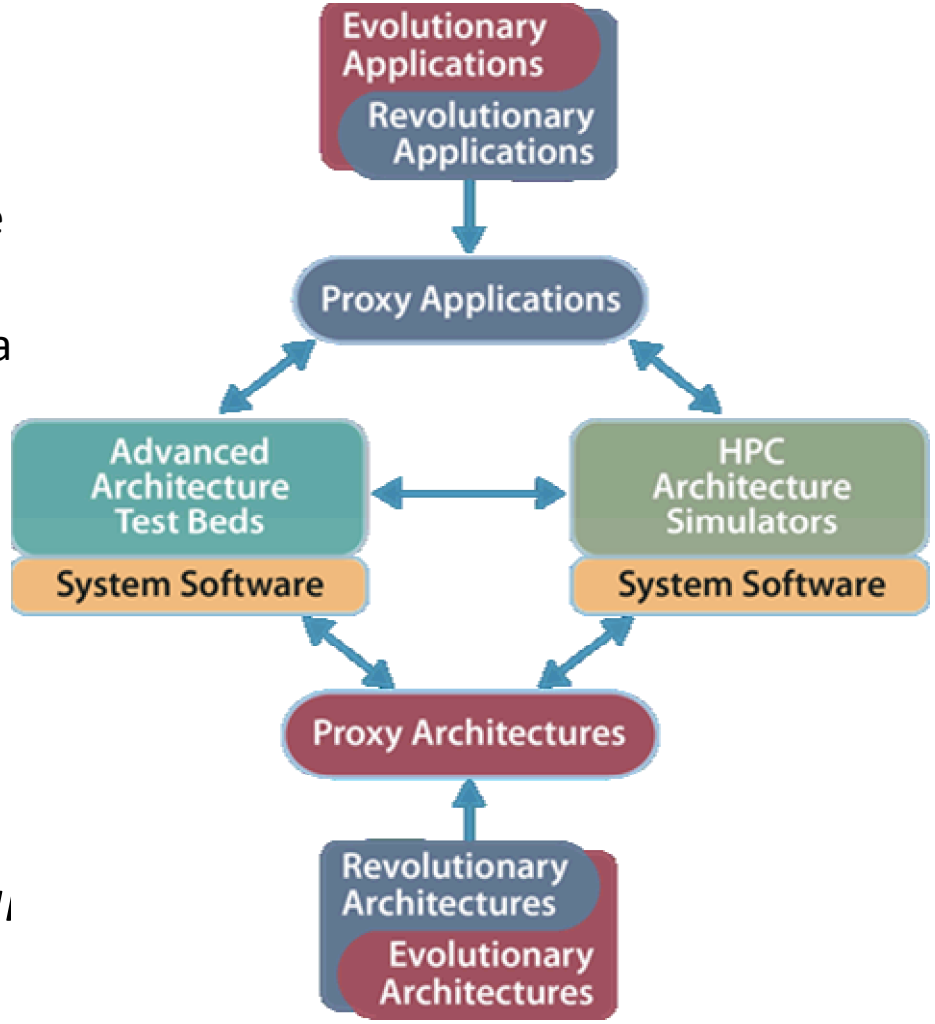
James A. Ang, Simon D. Hammond, Robert J. Hoekstra,
James H. Laros III, and Arun F. Rodrigues
Sandia National Laboratories
Center for Computing Research
Albuquerque, NM 87185-1319 USA

June 20, 2017
ISC 2017, Birds of a Feather #13,
Designing, Porting & Optimizing HPC Workloads for ARM Based Systems
Frankfurt, Germany

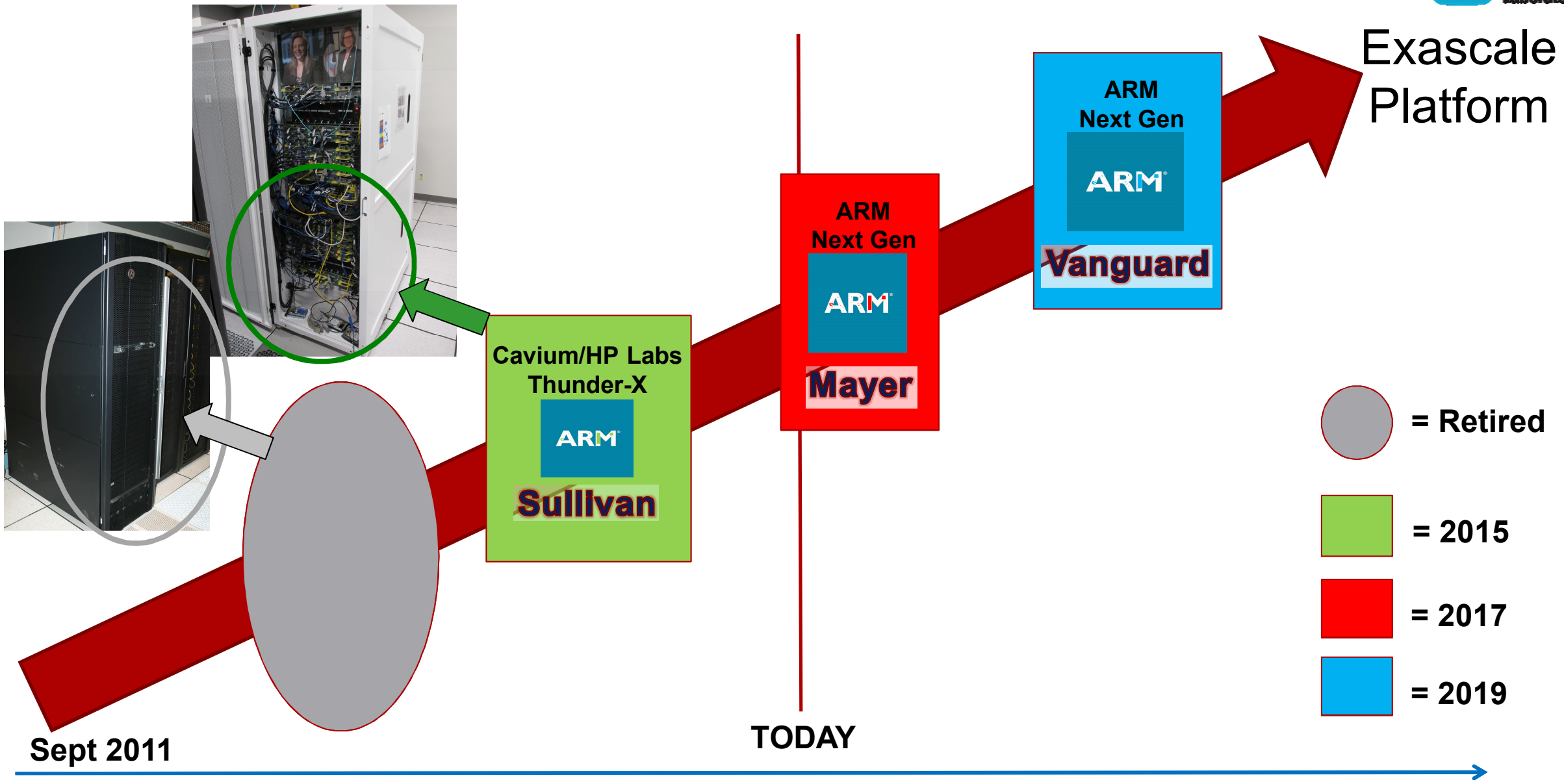
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Sandia ARM R&D Overview

- Sandia's ASC Advanced Architecture Testbed project
  - For co-design early access to new architectural features is critical
  - Positive ROI for the pain associated with early hardware & immature software
  - ARM testbed experiences – with Cavium and HPE's permission to sha Pre-production measurements
- Architectural Simulators
  - Structural Simulation Toolkit (SST), Open architectural simulation framework
  - Open Source, Open Development, catalyst for collaboration that can be proprietary
  - Links to Sandia/DOE's experience with application drivers
- Sandia is responsible for the NNSA/ASC's first *large sca* ARM platform
  - Overview of key requirements targeting 2019

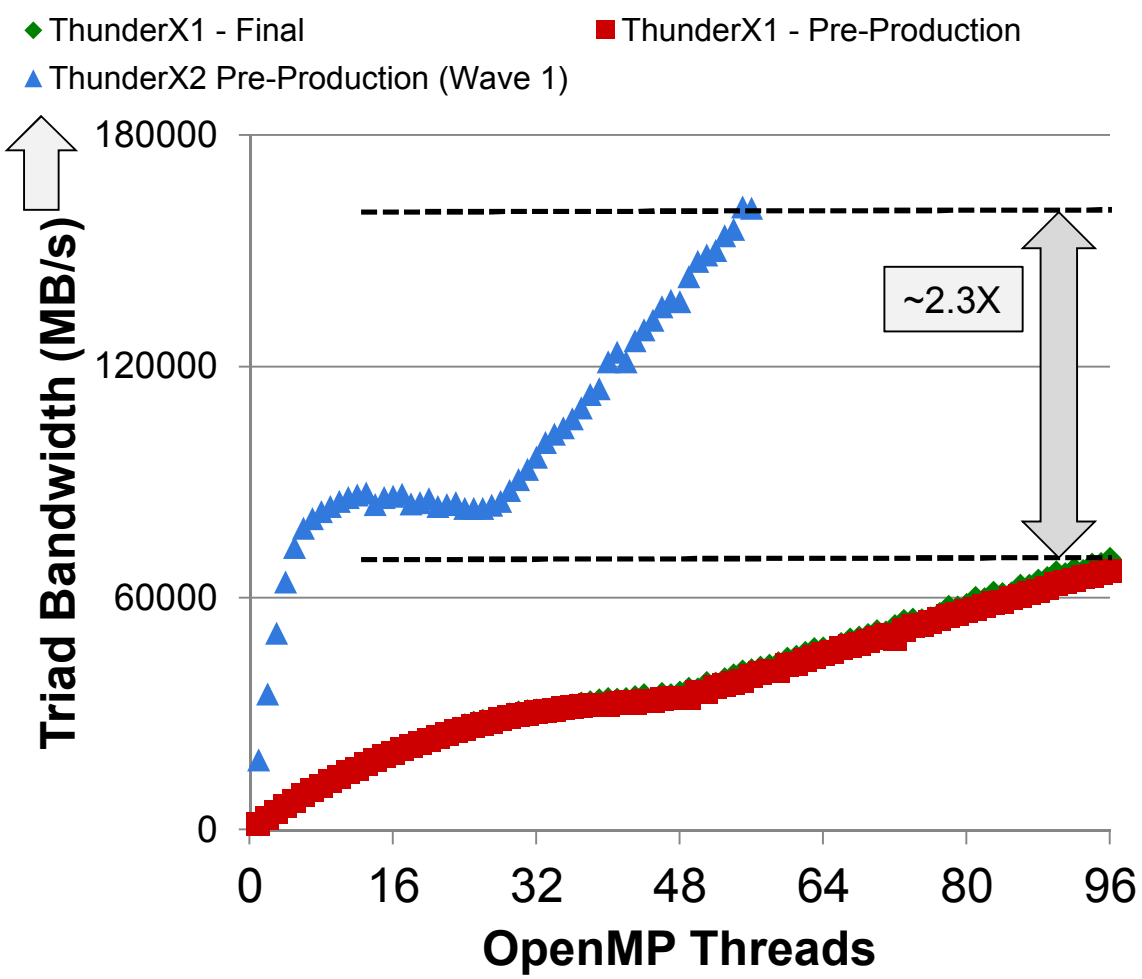


Sandia's NNSA/ASC ARM Testbeds

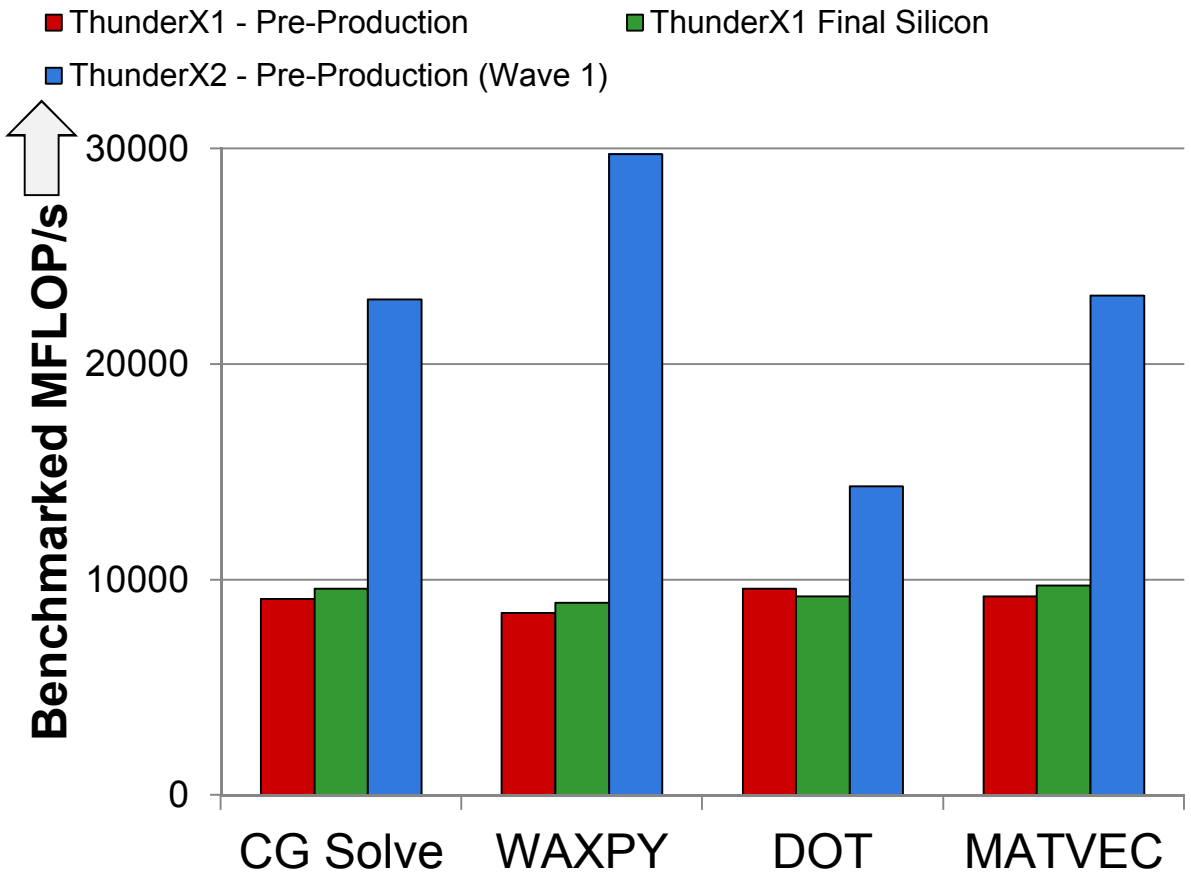


Memory Bandwidth Intensive Kernels

STREAM Triad



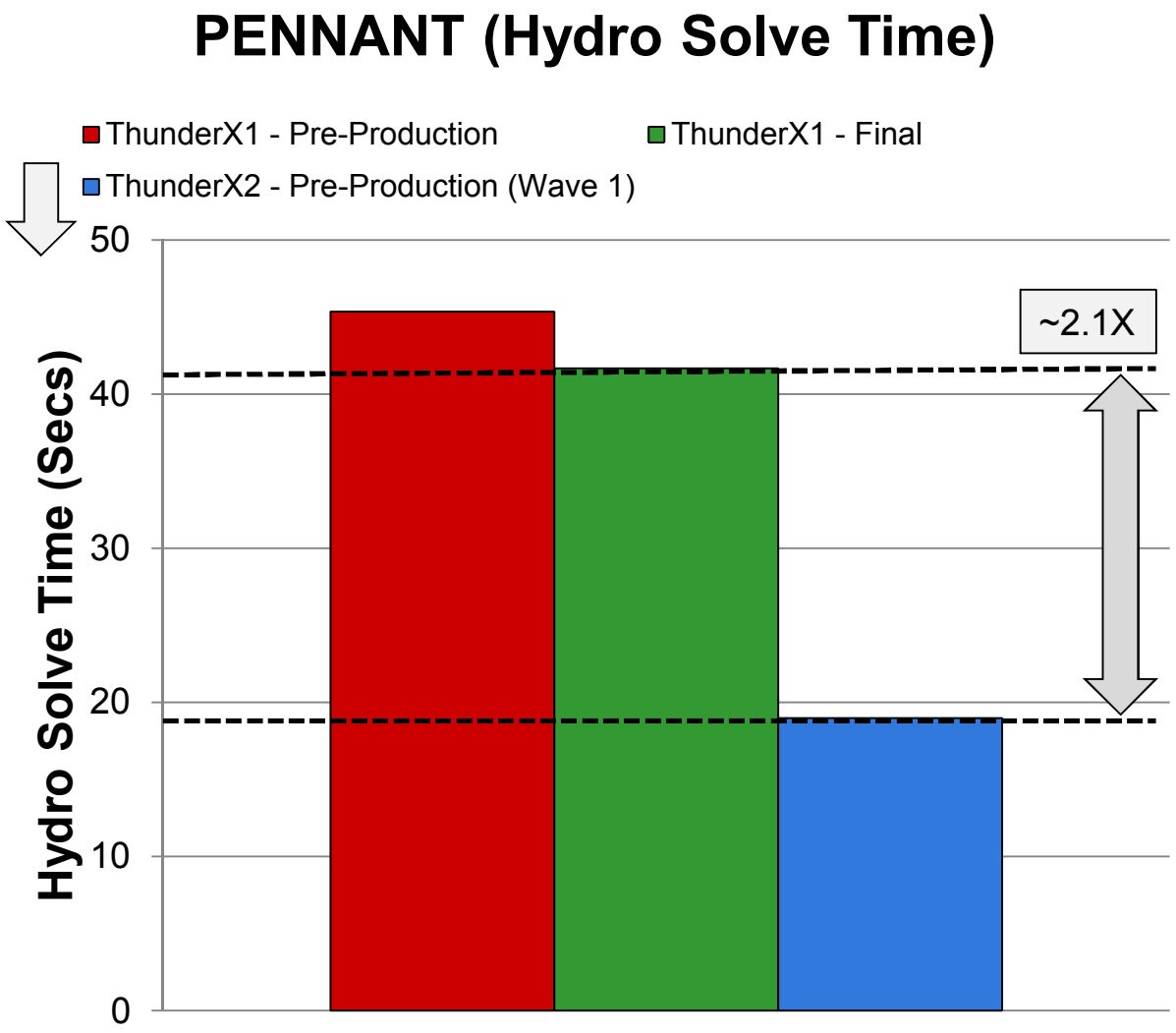
MiniFE CG Solve/Kernels



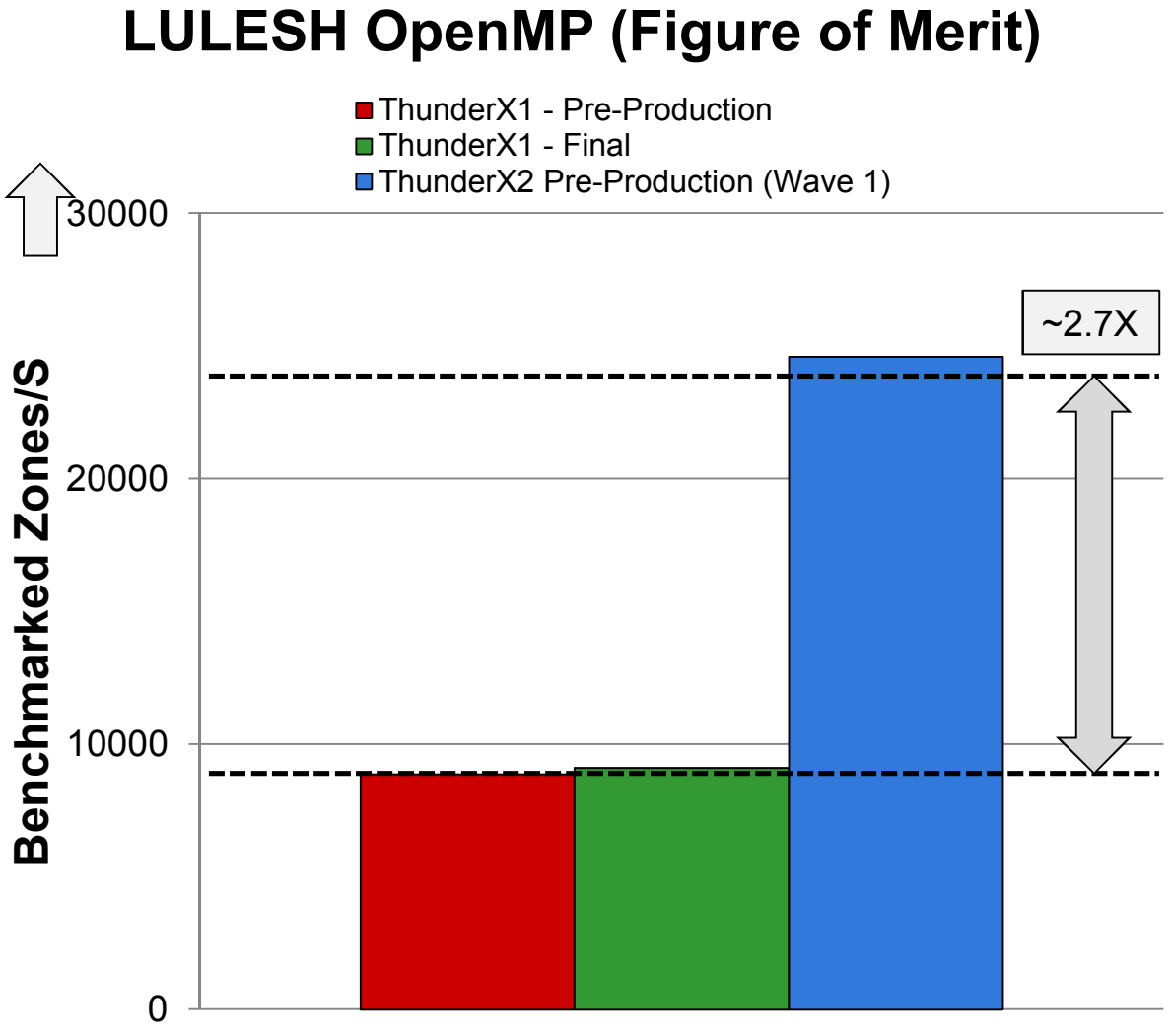
https://github.com/Mantevo/miniFE (Sandia National Labs)

Benchmarked on maximum number of cores (node-node comparison)

Compute Intensive Kernels



https://github.com/lanl/PENNANT (Los Alamos Nat. Lab)

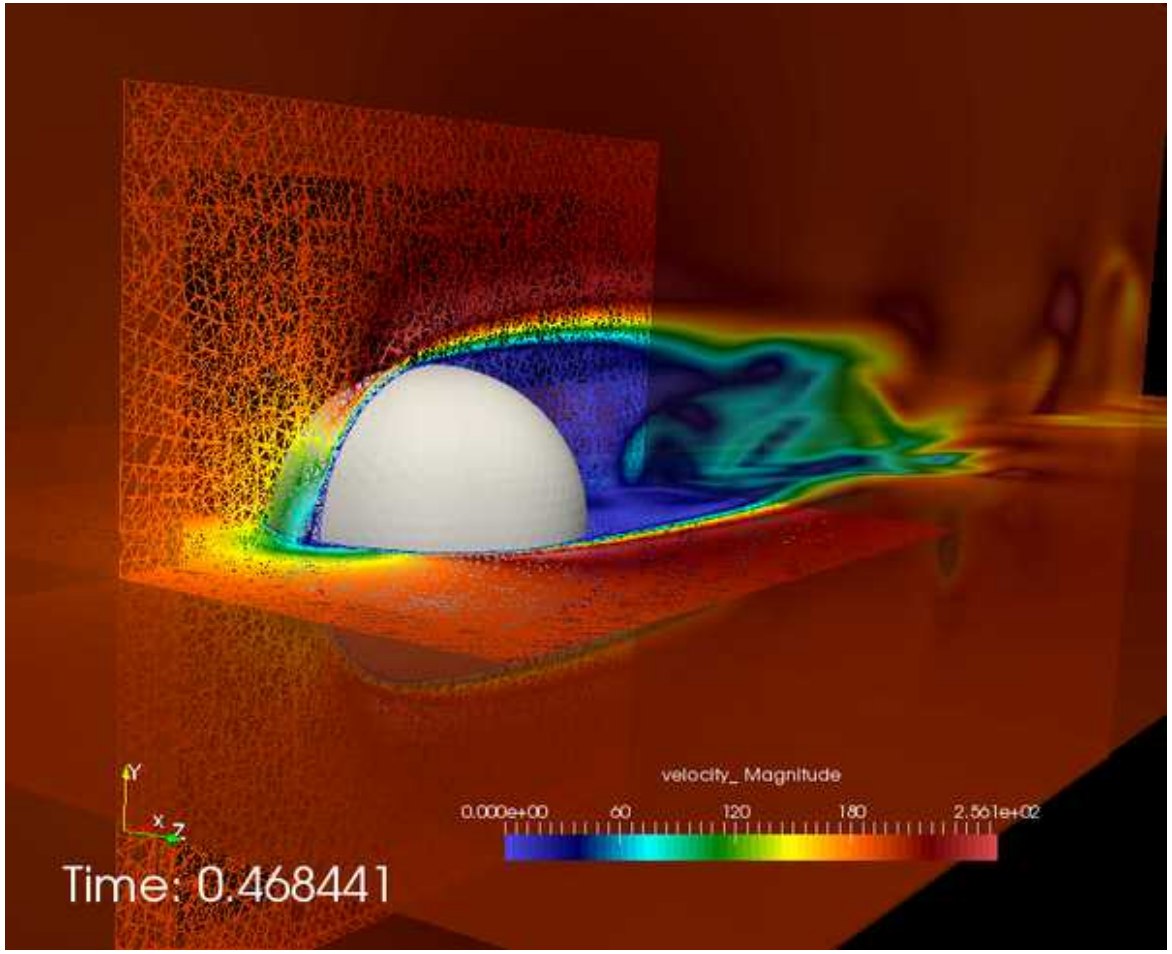


https://codesign.llnl.gov/lulesh.php (Lawrence Livermore Nat. Lab)

Full Engineering Applications on ARM

- Sandia has been actively working on libraries and packages for its ARM systems
  - Math Libraries/Kernels (Trilinos)
  - I/O Libraries (Exodus Mesh)
  - YAML

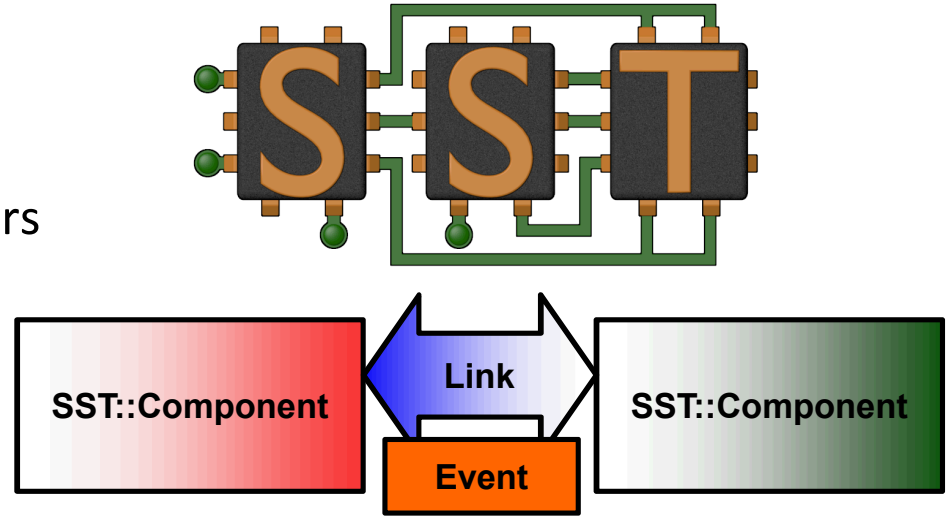
- Ported Sandia's open-source Trinity acceptance application to ARM
  - Representative of some SIERRA engineering applications
  - Complex mesh handling, load balancing, solvers *etc*

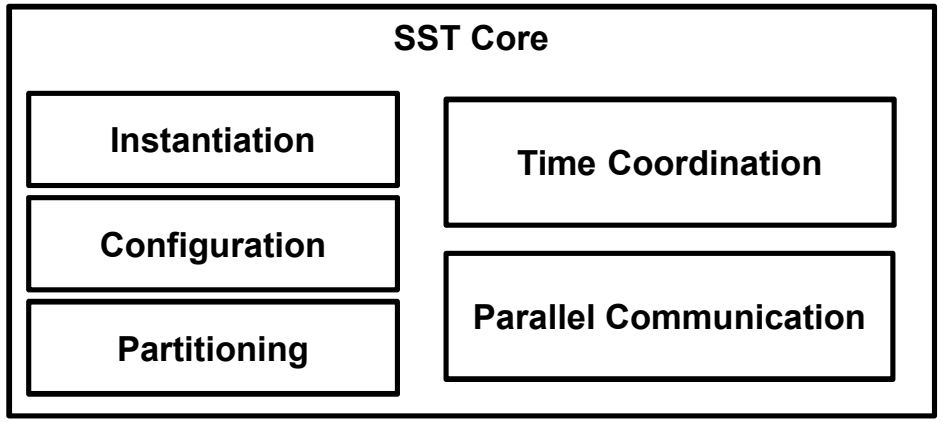


https://github.com/NaluCFD

SST in a Nutshell

- Use Supercomputers to Design Supercomputers
- Parallel Discrete-Event Simulator Framework
  - Flexible framework allows multitude of custom simulators
  - Demonstrated scaling to over 512 processors
- Comes with many built-in simulation models
  - Processors, Memory, Network
- Open API
  - Easily extensible with new models
  - Modular framework
  - (Non-Viral) Open-source core
- Time-scale independent core
  - Handles Micro-, Meso-, Macro-scale simulations
- "Best of Breed" – Bring together work from Labs, Industry, Academia





http://sst-simulator.org/

Example SST Element Libraries

Detailed Memory Models

- memHierarchy - Cache and Memory
- cassini - Cache prefetchers
- DRAMSim - DDR
- NVDIMMSim - Emerging Memories

Dynamic Trace-based Processor Model

- ariel - PIN-based Tracing

Cycle-based Processor Model

- m5C - Gem5 integration layer

High-level Program Communication Models

- ember - State-machine Message generation
- firefly - Communication Protocols
- hermes - MPI-like interface

Cycle-based Network Model

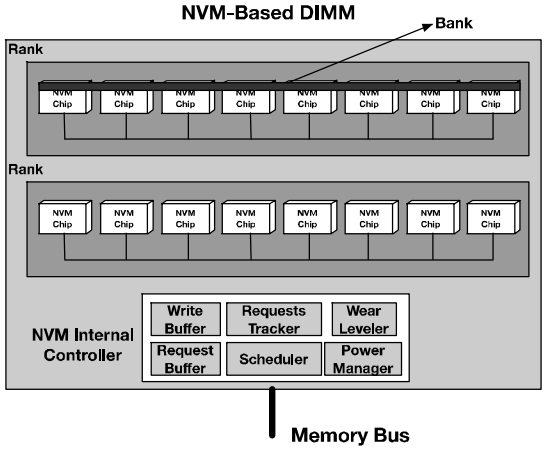
- merlin - Network router model and NIC

High-level System Workflow Model

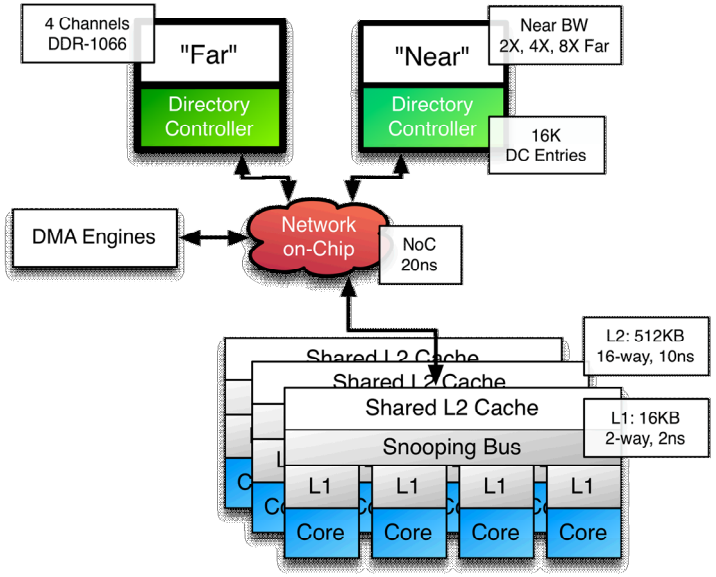
- scheduler - Job-scheduler simulation models

SST Use Cases

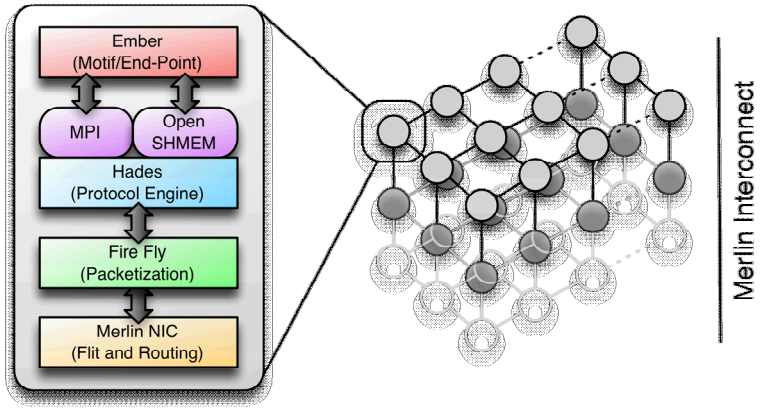
Emerging NV Memory Technologies



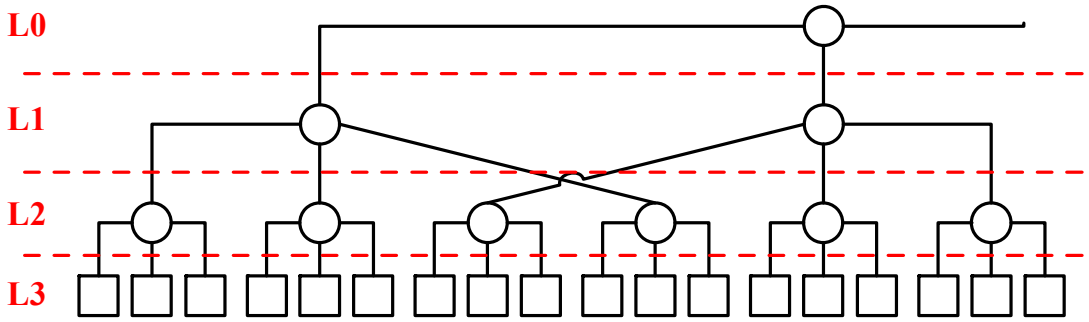
Multi-Level Memory (HBM+DDR+NV)



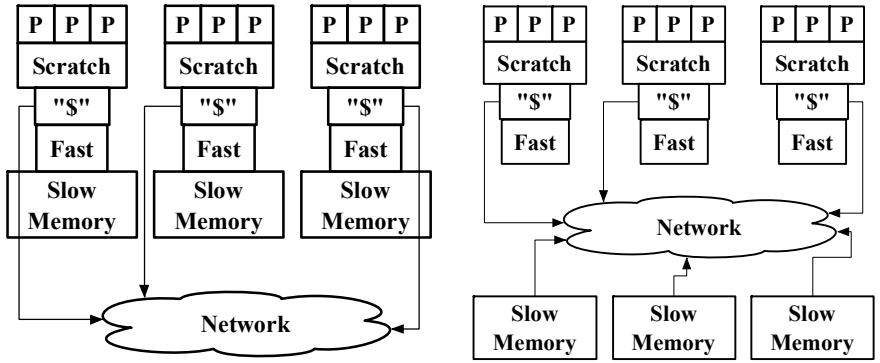
Communication/Interconnect Modeling



Photonic Network Topology & Routing



Disaggregated Memory



Goals of NNSA/ASC ARM Platform at Sandia

- Key Requirements for this 2019 Platform
  - Prototype for future leadership class DOE platform
  - Competitive HPC 64 Bit ARM processor technology
  - Integrated On-package Memory
  - First of a kind, Advanced Interconnect technology
  - Large focus on maturing the ARM software ecosystem
- Opportunities for *proof of concept* innovative test hardware
  - Logic in NIC to improve interconnection network performance electrical or optical
  - Logic in/near memory to support sparse linear algebra acceleration

Acknowledgements

- NNSA/ASC Program: Douglas Wade, Thuc Hoang
- Advanced Architecture Testbeds Operations Team: John Noe, Jim Brandt, Ann Gentile, Victor Kuhns, Nate Gauntt, Mike Aguilar
- SST Developers: K. Scott Hemmert, Branden Moore, Gwen Voskuilen, Amro Awad, Clay Hughes, Jeremy Wilke
- SST SQE team: Jon Wilson, Aaron Levine, John Van Dyke