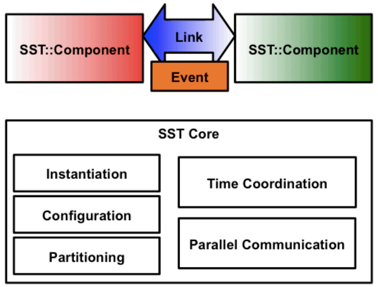
SAND2017-#### C

SAND2017-9713C



Sandia's ARM–centric Co-Design Strategy

Introduction to the NNSA/ASC Vanguard Project

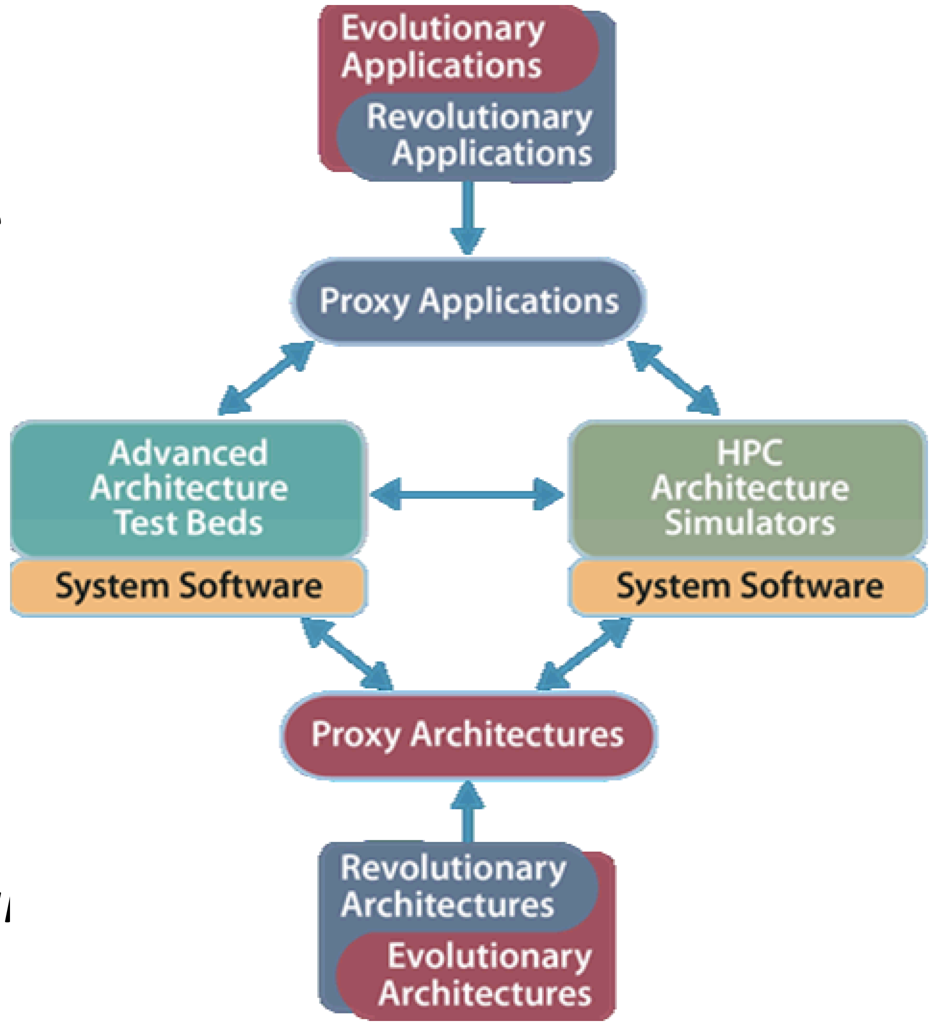
James A. Ang, Ron Brightwell, Simon D. Hammond, K. Scott Hemmert. Robert J. Hoekstra, James H. Laros III, Kevin Pedretti and Arun F. Rodrigues
Center for Computing Research
Sandia National Laboratories
Albuquerque, NM 87185-1319 USA

ARM Research Summit, Going ARM Workshop
Robinson College, Cambridge, UK
September 11-13, 2017

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.

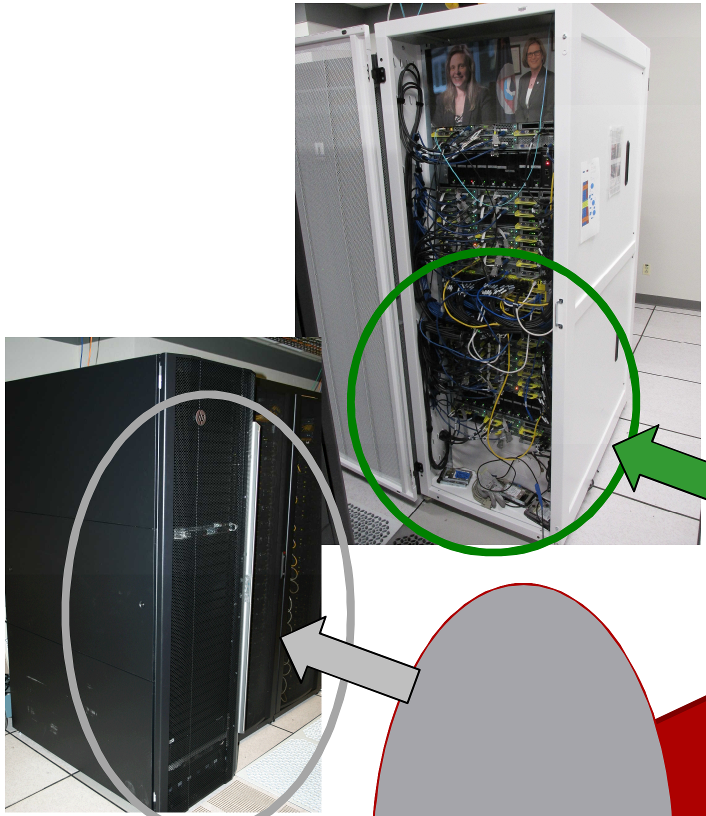
Sandia ARM R&D Overview

- 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's permission to share pre-production measurements of relative performance
- 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 scal* ARM platform
  - Overview of key requirements targeting 2019



Sandia's NNSA/ASC ARM Platforms

Sandia National Laboratories



Exascale Platform

ARM Next Gen
ARM
Vanguard "Project"

Install FY19 Target ThunderX3 (Triton)

ARM Next Gen
ARM
Mayer

Pre-production Cavium ThunderX2

Small Pre-Vanguard early access system – FY18

Cavium/HP Labs ThunderX1
ARM
Sullivan

Cavium ThunderX1

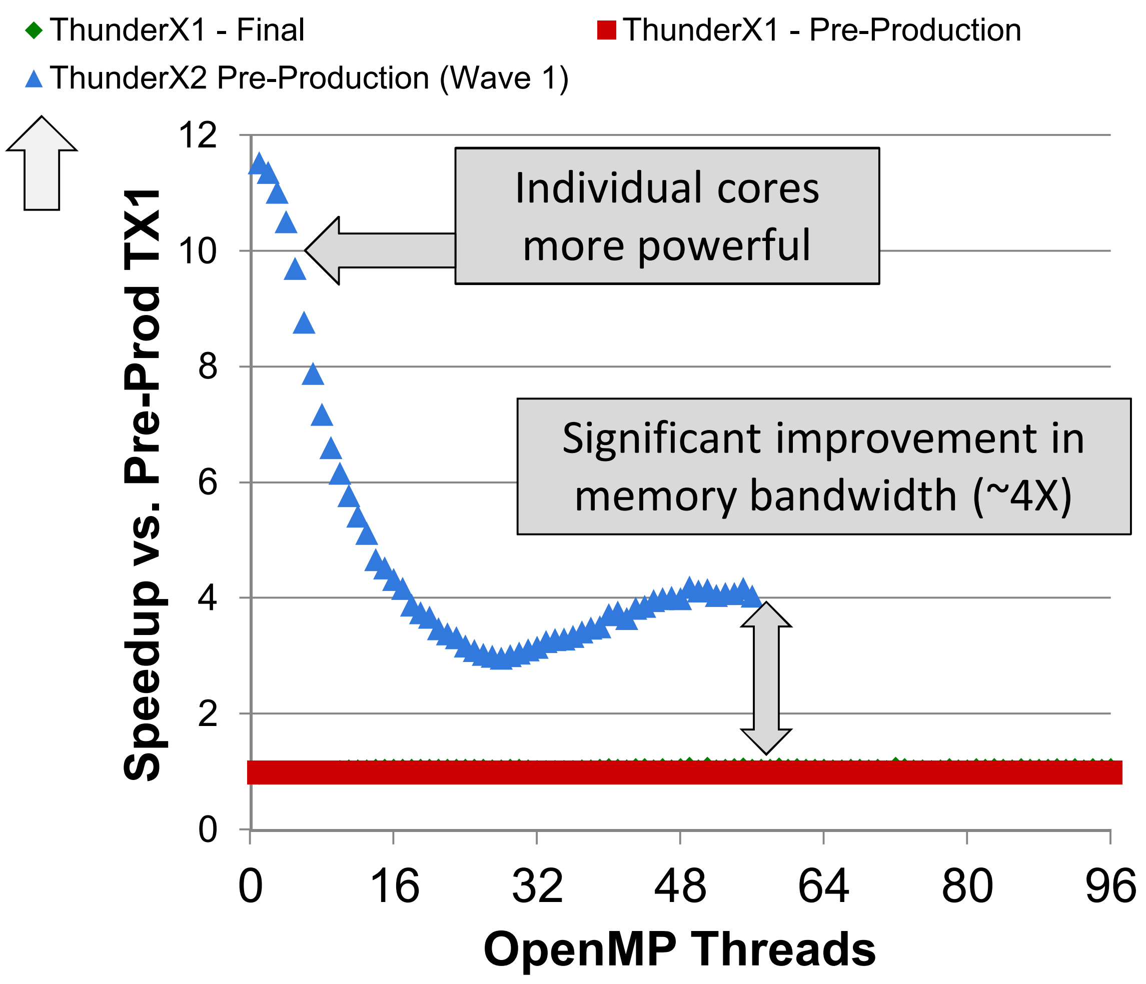
TODAY

Sept 2011

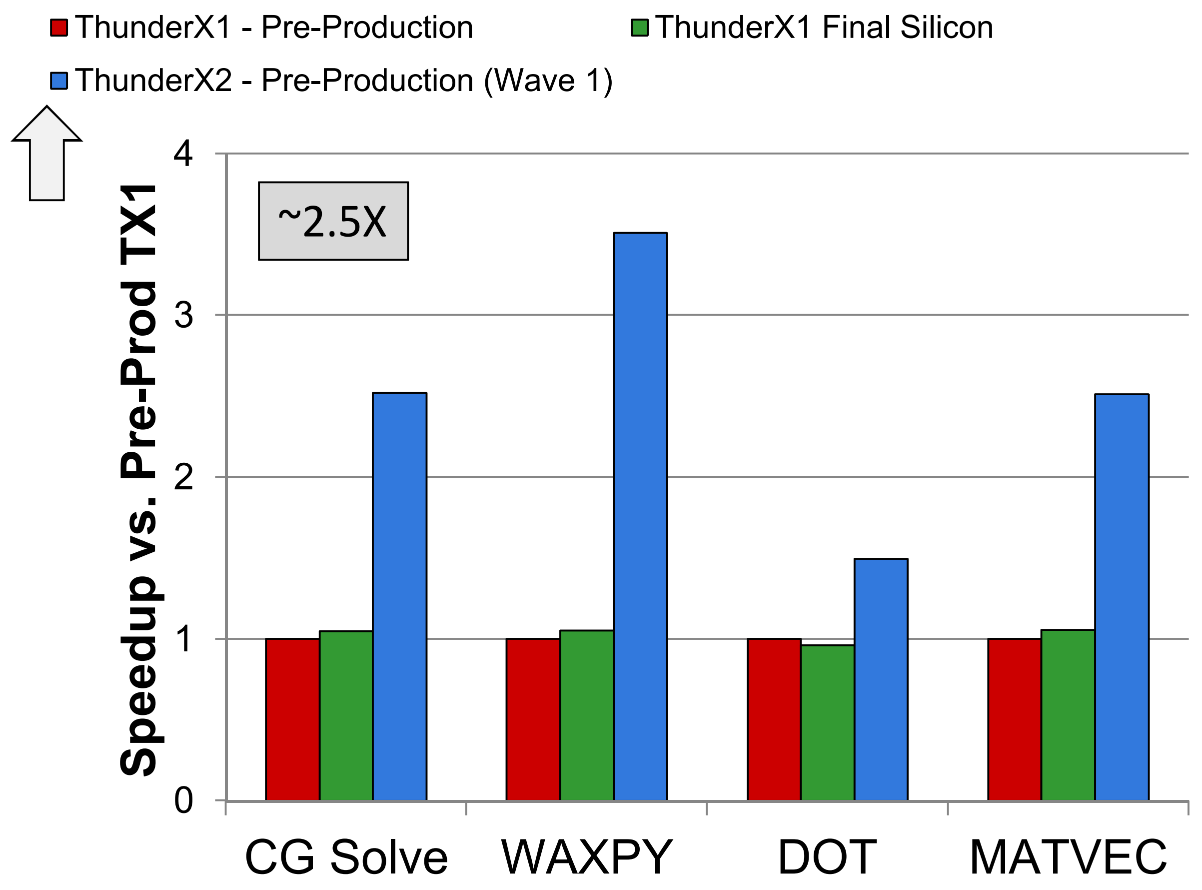
= Retired
= 2015
= 2017
= 2019

Memory Bandwidth Intensive Kernels

STREAM Triad



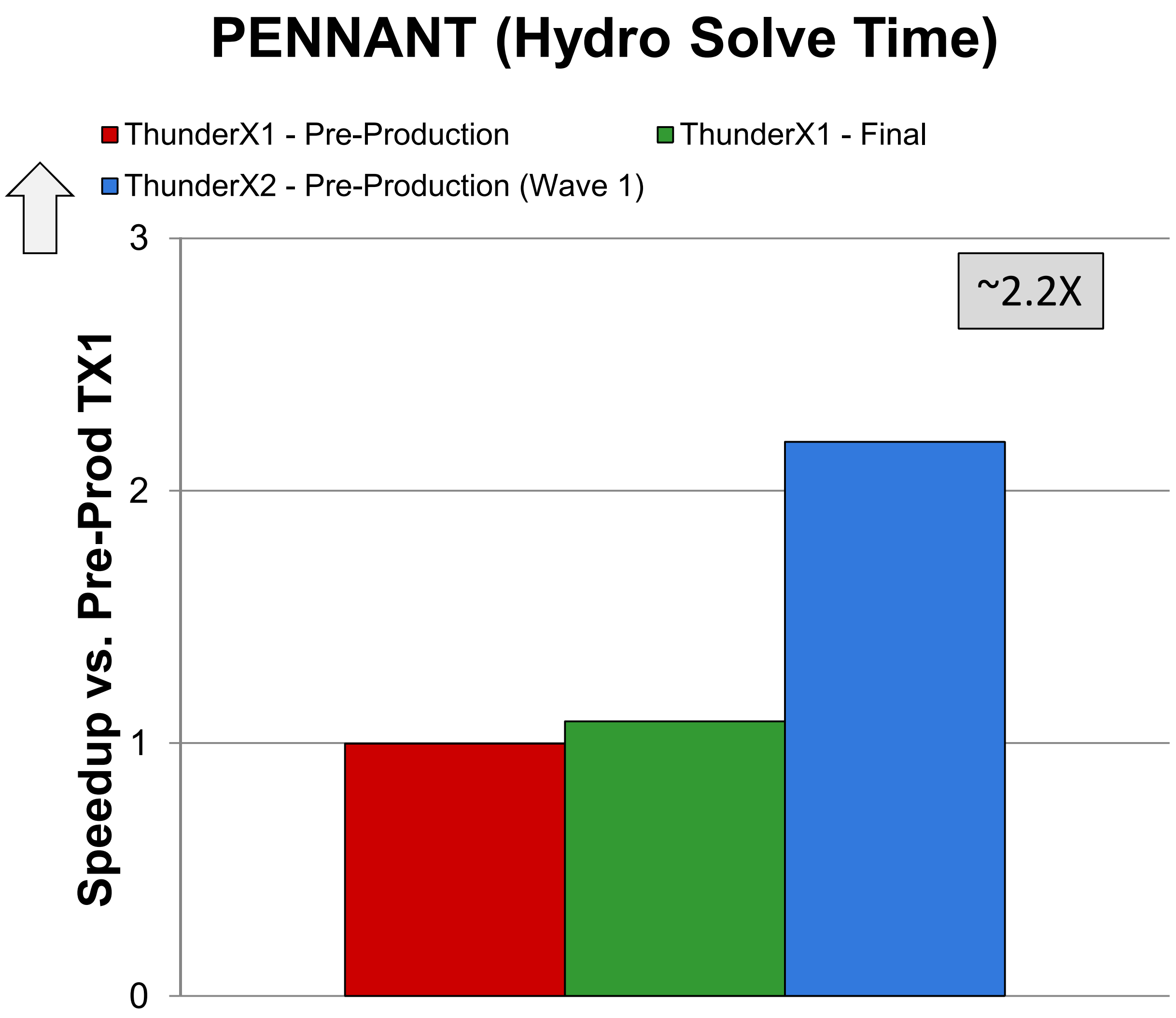
MiniFE CG Solve/Kernels



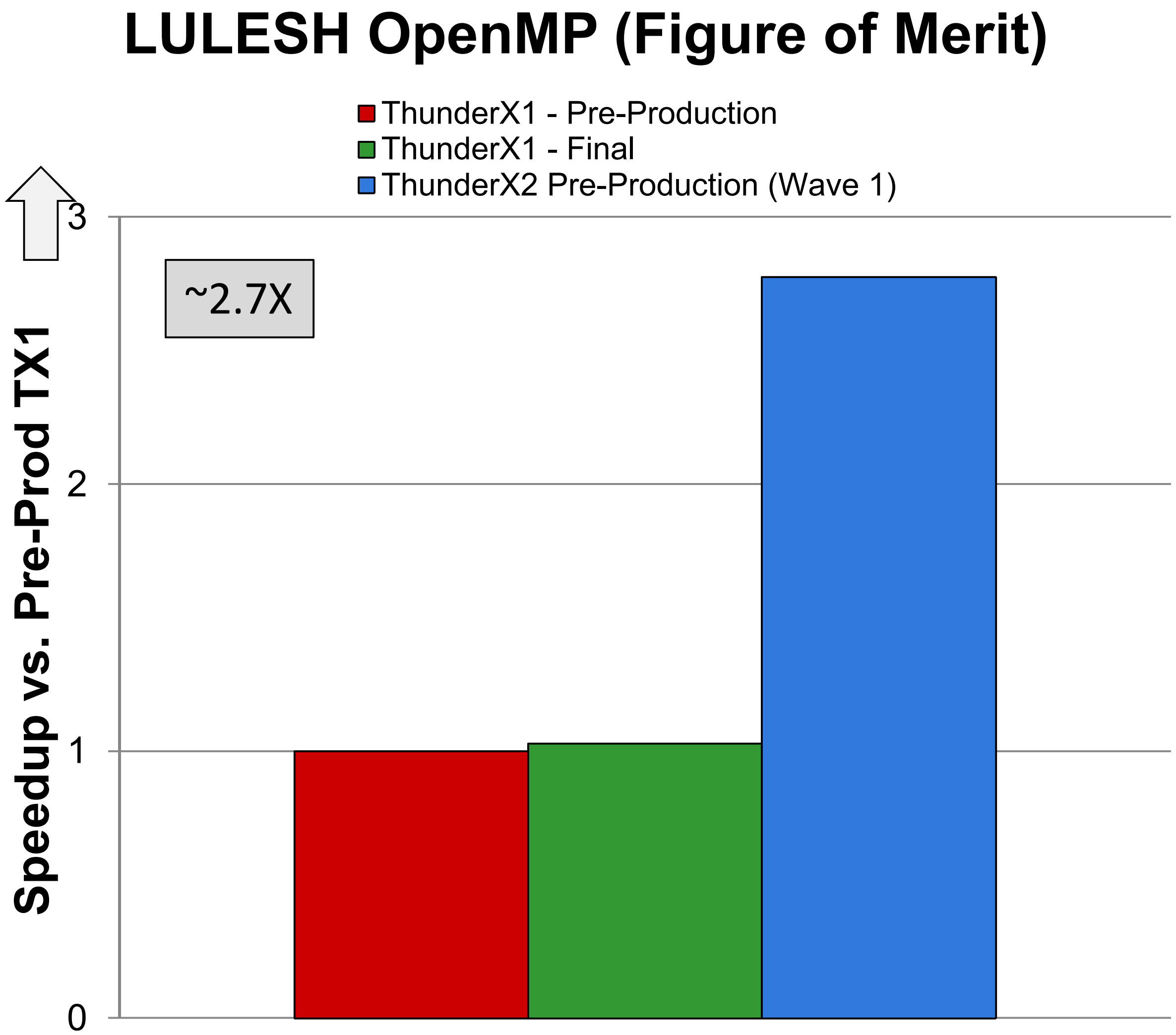
https://github.com/Mantevo/miniFE (Sandia National Labs)

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

Compute Intensive Kernels



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

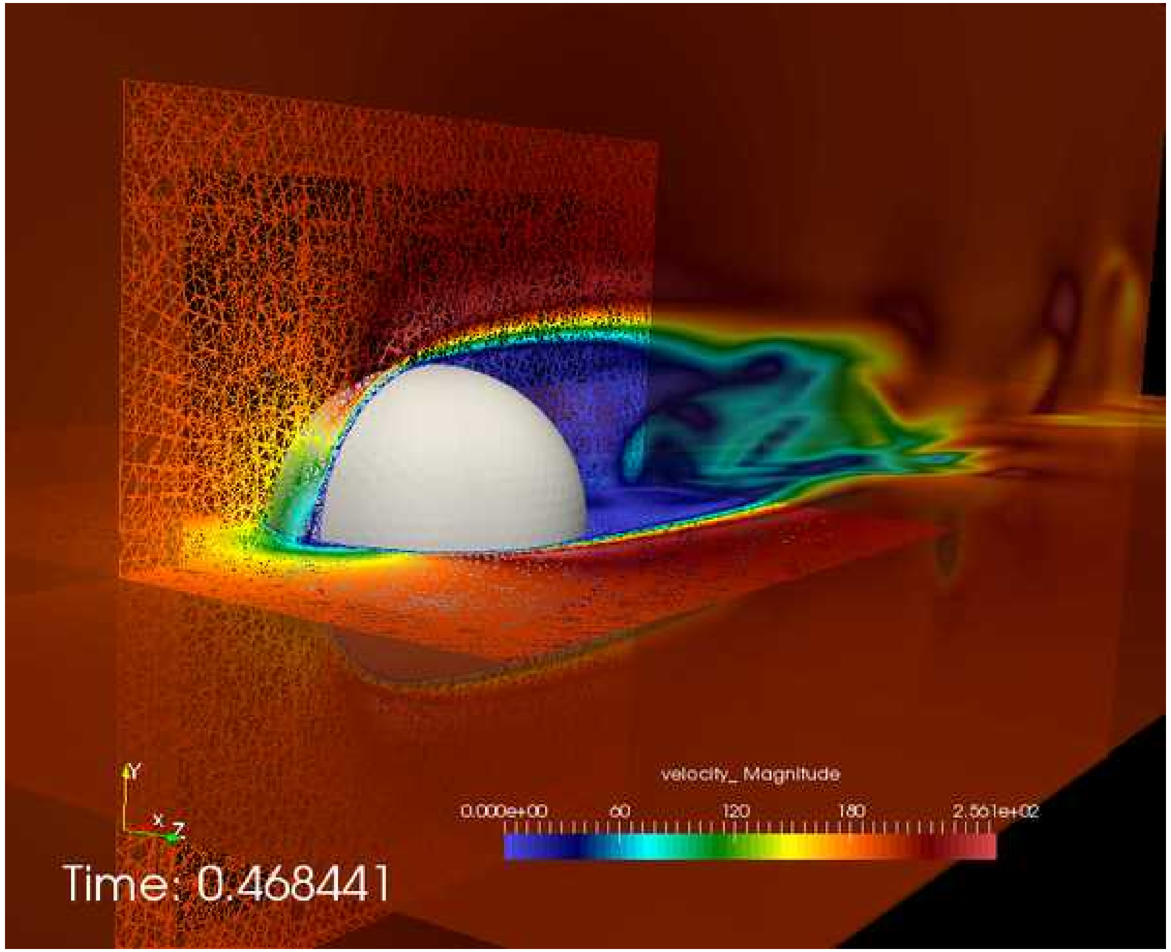


https://codesign.llnl.gov/lulesh.php (Lawrence Livermore Nat. Lab)

Full Engineering Applications on ARM

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

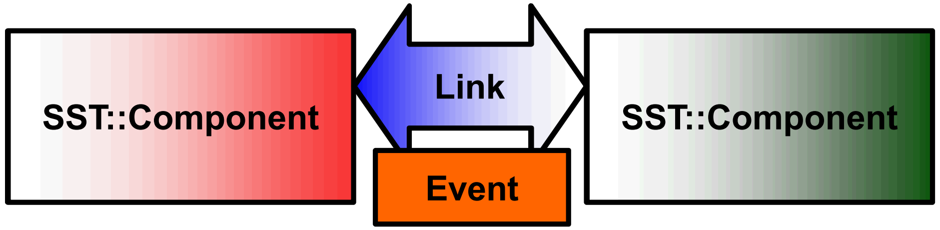
- Ported Sandia's open-source *Trinity* acceptance application to ARM
  - Representative of some SIERRA engineering applications
  - Complex mesh handling, load balancing, solvers *etc*

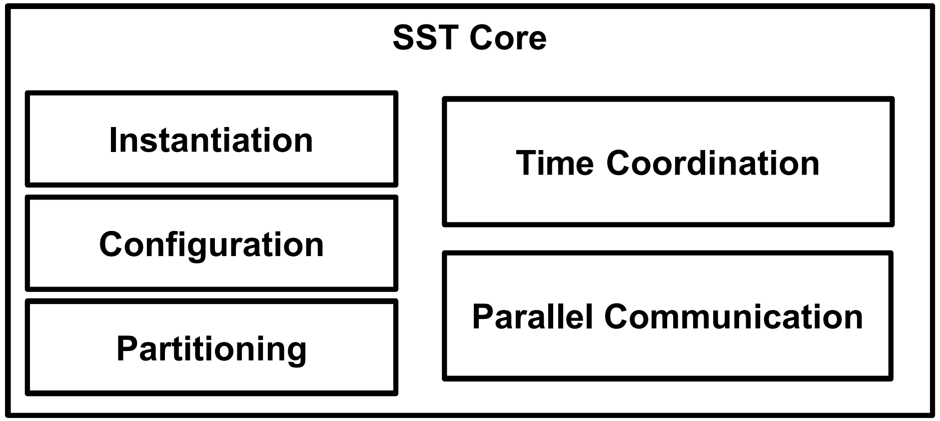


https://github.com/NaluCFD

Architectural Simulation Framework: SST

- 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





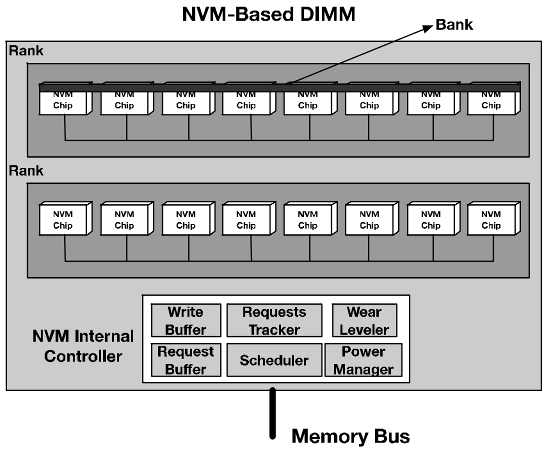
http://sst-simulator.org/

Example SST Element Libraries

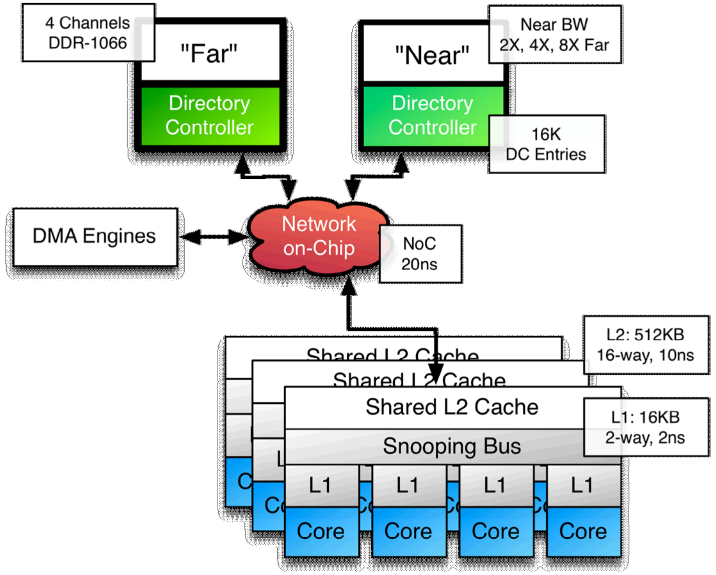
Detailed Memory Models	memHierarchy	- Cache and Memory
	cassini	- Cache prefetchers
	DRAMSim	- DDR
	NVDIMMSim	- Emerging Memories
Dynamic Trace-based Processor Model	ariel	- PIN-based Tracing
Cycle-based Processor Model	m5C	- Gem5 integration layer
High-level Program Communication Models	ember	- State-machine 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

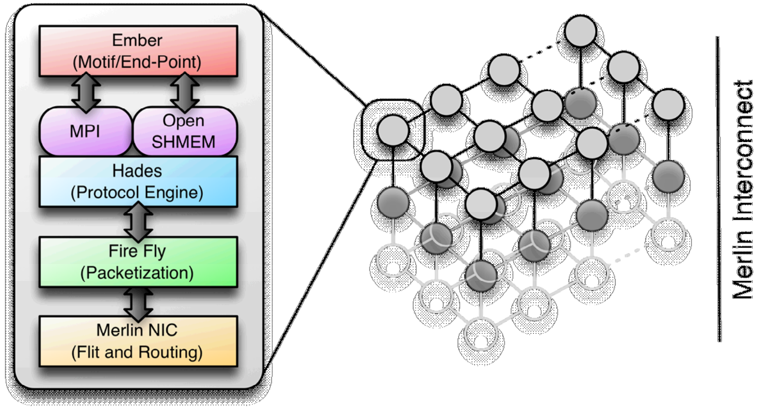
Emerging NV Memory Technologies



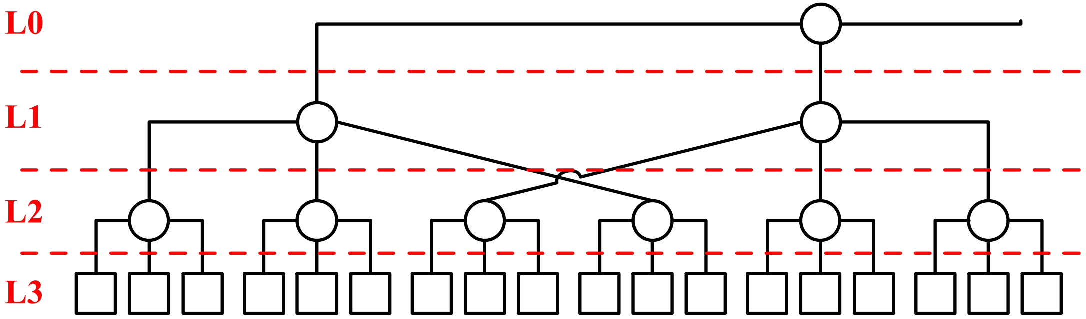
Multi-Level Memory (HBM+DDR+NV)



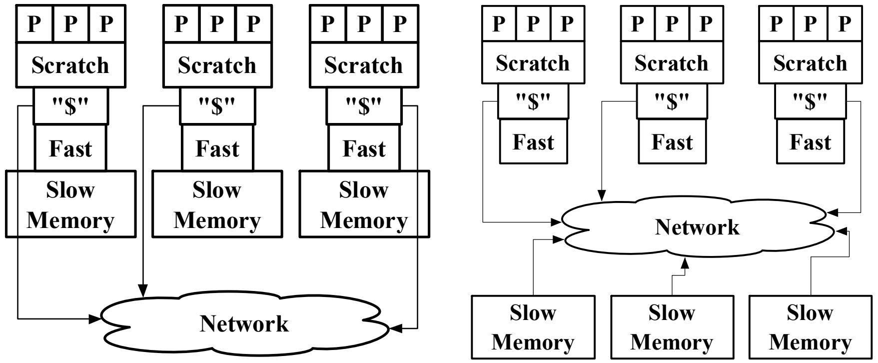
Communication/ Interconnect Modeling



Photonic Network Topology & Routing



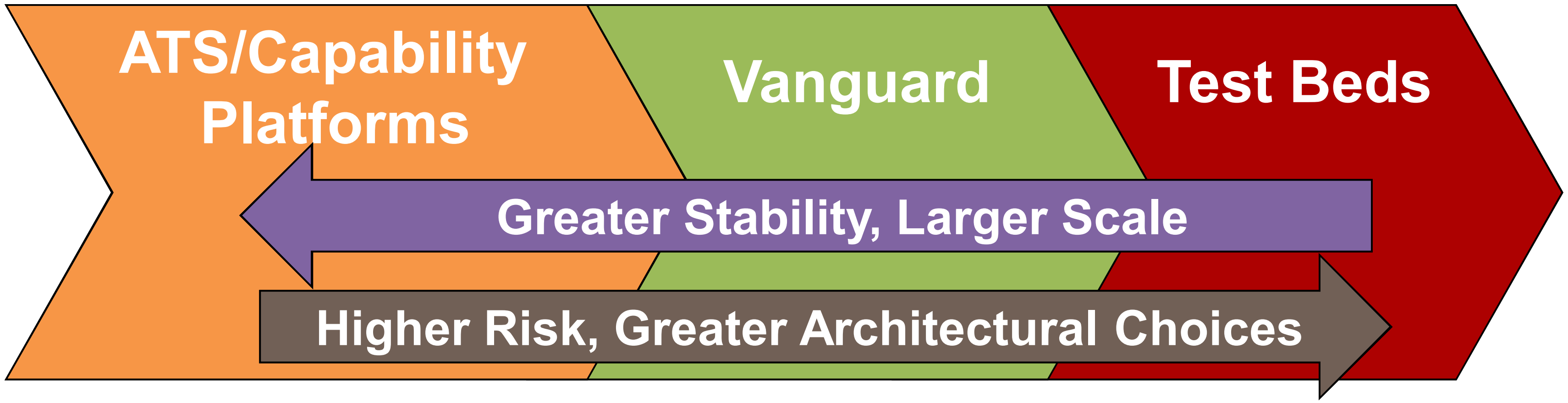
Disaggregated Memory



Goals of NNSA/ASC Vanguard Project

- **Expand HPC ecosystem** by developing ARM-based system necessary to enable a **credible** ARM-based ATS offering for ASC post-2022
  - On-package high-bandwidth memory
  - Advanced HPC Interconnect
  - **Mature ARM HPC ecosystem**
    - Software stack and toolchain (OS, compilers, scalable MPI, runtime, system management, development tools, I/O, …)
  - Influence Supplier/Integrator community to accelerate and optimize ARM technologies for leadership HPC, and foster HPC community confidence
  - **Ultimate success for this platform will be running multi-physics production codes at scale**
- Leverage the open ARM architecture platform to enable innovations in processor, interconnect, memory, specialized acceleration and packaging technologies that ***focus on performance and scalability of our multi-physics production codes*** and could impact 2025 systems

The Vanguard Project Approach



- Bridge the gap between what we can accomplish with Testbed project and fielding an advanced technology system platform
- Vanguard is a project, not a single platform
- Accelerate maturity of emerging technologies for ASC program

Vanguard Approach (cont.)

- **Appropriate** Scale systems
  - Must be large enough to serve as proof of concept for future advanced technology systems
  - Also large enough to be of interest to mission multi-physics application teams
- **Appropriate** Investment
  - Gain and maintain vendor and collaborator attention
- **Appropriate** level of risk
  - Goals target mission workloads but not turn-key production mission support
- Delicate balance
- Might be different balance based on target technology and other factors

Software Environment Plan - Overview

- Goal: Accelerate maturity of ARM ecosystem for ASC computing
- Need an integrated software stack for the 2019 ARM Prototype to enable application development and optimization
  - Programming environment (compilers, math libs, tools, MPI, OMP,…..)
  - Low-level OS (optimized Linux, I/O, network, containers + VMs, PowerAPI, …)
  - Job Scheduling and management (WLM(Slurm?), app launcher, user tools, ….)
  - System Management (boot, monitoring, image mgt., rapid reprovisioning, ….)
- Focus Areas
  - Integration and robustness, overall user experience
  - Specifically address known weaknesses: compilers, libs, and tools
  - Increase modularization and openness of system software stack; seek “plugin” capability for externally developed components.

VANGUARD

Software Environment Collaboration Roles

- System Vendor
  - Deliver and support core elements of the software environment necessary for a viable integrated system (part of system contract)
- Tri-lab team
  - Integrate system into our computing environment
  - Identify and resolve SW issues in collaboration with system vendor
  - Contribute tools and other capabilities to fill gaps and improve the overall computing environment
- Other Eco-system Vendors/Stakeholders
  - Linaro, OpenHPC
  - ARM/Allinea
  - . . .

Drivers for ARM Software Stack

1. Focus on NNSA/ASC's unique requirements
   - Many programs are already exploring HPC on ARM (e.g., Mont-Blanc, Post-K)
   - Need to demonstrate full multi-physics applications running at scale on ARM
2. Build an integrated team
   - Integrate work across tri-labs and vendors, managed as single project
   - Focus on strengths of each laboratory, support technology deployment
3. Provide a robust "traditional" HPC software stack
   - Meet user expectations, provide familiar environment
   - ARM is just an ISA; much of system software stack should "just work"
4. Engage with users
   - Enable low-effort issue reporting, rapidly resolve, get and provide feedback
   - Build community through talks, training, hackathons, workshops, etc.
5. Look forward
   - Improve support for real-world workflows and data-centric computing
   - Seek to reduce friction points for users when moving between platforms
   - Leverage software technology from Exascale Computing Project

Technology Development Areas

- 2019 ARM platform is **not** a production system – increased latitude for experimentation and R&D activities
  - Need to address options for scheduling and types of access to the system to provide for full scale projects, alternate environments, etc.
- Plan to pursue several technology development areas, possibly:
  - On-package high-bandwidth memory
  - Advanced HPC Interconnect
  - Compilers, math libraries, tools, task-based runtimes, ...
    - Vendor framework that OpenHPC components get plugged into
    - OpenHPC framework that vendor components get plugged into
  - Experimental OS/R stacks
    - Lightweight kernels (Hobbes/Kitten, Intel mOS, RIKEN McKernel)
    - KVM support for cyber "simulate the Internet" frameworks
    - Support for HPC + Analytics (including AI and ML) workflows and compositions
    - More interactive IaaS-style usage models

Required Vendor Support

- Modular system software environment architected to facilitate site modifications and additions.
- Documented APIs for integrating new software components with vendor stack (e.g. RAS APIs, Network, I/O, Health of subsystems,…)
- Ability to partition the system and manage multiple OS images
  - Carve out dedicated resources for R&D activities
  - Boot new or experimental software environments
- Buildable source – High Priority
  - Ability for labs to debug and fix system software issues
    - For example, be able to debug MPI, PMI, Lustre, Burst Buffers, app load, …
  - Ability to rebuild vendor Linux kernels with modified config options
    - Requires vendor to provide source for all “binary-only” kernel drivers and all non-standard Linux kernel patches
  - Ability to build and evaluate experimental OS/R stacks

Acknowledgements

- **NNSA/ASC Program:** Mark Anderson, 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
- **Cavium:** Gopal Hegde, Rishi Chugh, Larry Wikelius
- **Cavium (former Broadcom):** David Hass, Giri Chukkapalli
- **ARM:** Nigel Paver, Eric Van Hensbergen, Geraint North
- **HPE:** Mike Vildibill, Nic Dube, Kelly Pracht
- **Cray:** Duncan Roweth, Dan Ernst, Larry Kaplan, Heidi Poxon

Goals of NNSA/ASC Vanguard Project

- Key Requirements for this 2019 Platform
  - Proof of Concept 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 to *prototype* innovative test hardware
  - Logic in NIC to improve interconnection network performance electrical or optical
  - Logic in/near memory to support sparse linear algebra acceleration
  - Test and evaluation of other advanced processor architectures

Current NNSA Architecture Strategy