Maximizing Investments in ASC Simulation Codes at Los Alamos

Salishan Conference on High-Speed Computing
Gleneden Beach, OR
April 26, 2018

Jerry S. Brock
Advanced Simulation & Computing
Deputy Program Director

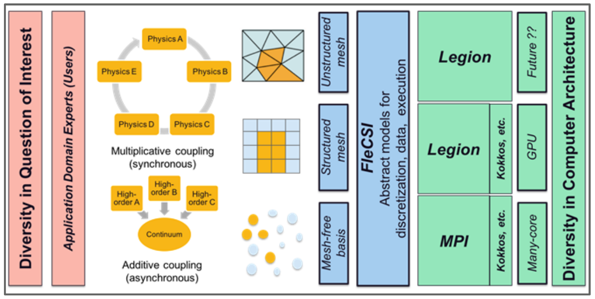
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Leverage Distant/Recent Past Investments & Target New Investments for Future Simulation

- **Distant Past: 75 Years of Computing & Simulation Code Evolution at LANL**
- **Recent Past: ASCI/ASC Enable HPC Simulation Science to Support US Stockpile**
- **Today: Leverage ASCI/ASC Code Investments onto Next-Generation Platforms**
- **Today: Target Investments in Codes & Technologies for Next-Generation Simulation**

First US Test: Trinity 1945

LANLs Next-Gen Physics Code

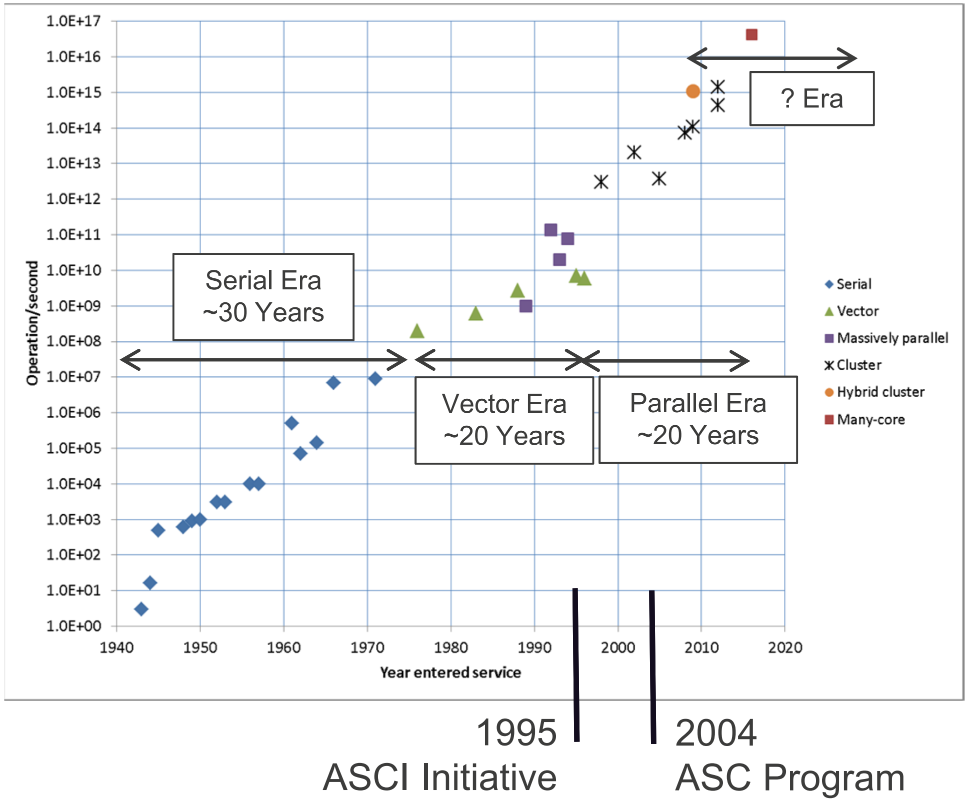


HPC Simulation Science Supporting the US Stockpile: ASCI 1995, ASC 2004

75 Years of Computing & Simulation Code Evolution at LANL

- LANL has Driven & Taken Advantage of Increased Computing Capability
- 16 Orders-of-Magnitude Computing Capability Increase Over 75 Years

Physics simulation codes have realized a similar leap in capability, & embody decades of knowledge. ASC must leverage past code investments & target new efforts for the physics simulation tools of the future.



ASCI: Accelerated Strategic Computing Initiative
ASC: Advanced Simulation & Computing Program

Serial Era: 1943 – 1952

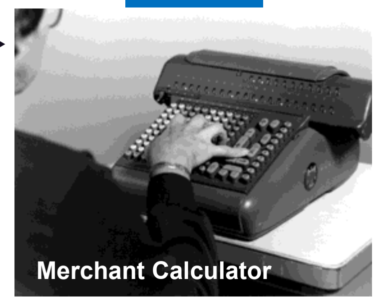
- **Mechanical 1943 - 1958**

- **Electro-Mechanical 1944 - 1950**
- **Electronic 1945 – 1948**

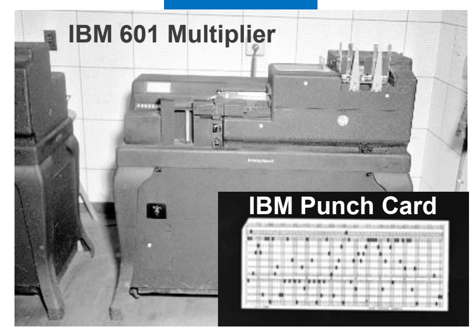
- ENIAC First Calculation 1945
- Monte Carlo Invented at LANL 1947
  Stan Ulam & John von Neumann
- Hydro Artificial Viscosity
  Invented at LANL 1948
  Richtmyer & von Neumann
- First Shock-Hydro Code
  Ran On IBM Prototype

3 Ops

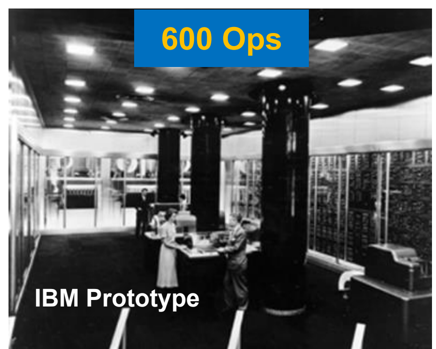
Computer →



16 Ops



600 Ops



13,000 Vacuum Tubes

500 Ops

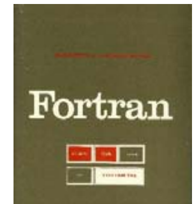
18,000 Vacuum Tubes; Patch Cords

ENIAC: Electronic Numerical Integrator & Calculator

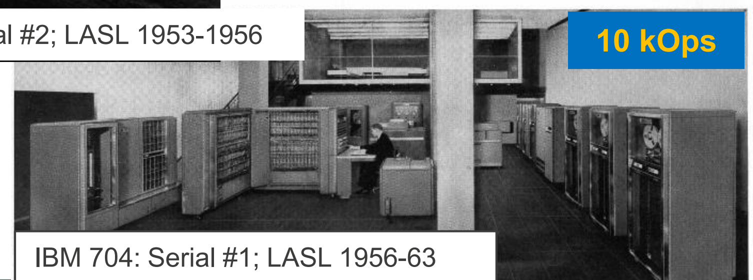
Serial Era: 1953 – 1975

- **Electronic 1953 - 1963**
  - IBM 701 & 704 were Revolutionary
    Programmed in Assembler
  - 701 Cathode-Ray-Tube Memory
  - 704 Floating Point Hardware
    & Magnetic Core Memory
- **Transistors 1964 – 1975**
  - CDC 6600: Fastest 1964-1969
  - CDC 7600: Fastest 1969-1976
  - Arbitrary Lagrangian Eulerian (ALE)
    Hydro Invented at LANL 1969

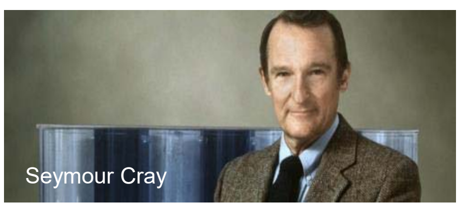
IBM 701: Serial #2; LASL 1953-1956

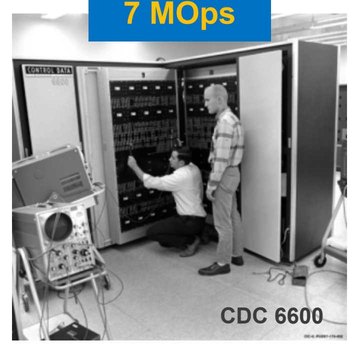


IBM 704, 1957



IBM 704: Serial #1; LASL 1956-63





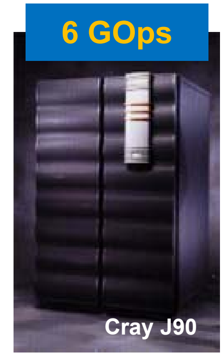
Vector Era: 1976 – Mid-1990s

First Parallel Calculations

- Same Operation on Multiple Data
- *Required Rewriting Weapons Codes*

Early Cray Machines at LANL

- Cray 1 Co-Designed with LANL
- Cray XMP 1983; Cray YMP 1988
- At LANL: T94 Until 2003; J90 Until 2004



These early efforts began a years-long relationship between Cray & LANL that continues today.

Parallel Era: Start Mid-1990s

Another Type of Parallelism
- Parallel Cluster Commodity Systems
- *Required Rewriting Weapons Codes*

1990s Decade of Great Change
- Fall of the Soviet Union 1991
- End of US Nuclear Testing 1992

Large-Scale Simulation Science
- Complements Theoretical & Experimental Science to Underpin US Stockpile
- Enabled Simulation-Based Certification of the US Stockpile

Simulation-based certification made possible by the ASCI/ASC HPC computers & predictive codes.

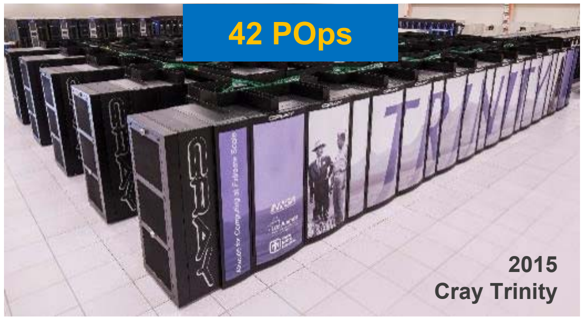
ASCI: Accelerated Strategic Computing Initiative
ASC: Advanced Simulation & Computing Program

Next Era: Start 2009 - Entering a New Era of Computing that is Still Being Defined

Another Type of Parallelism

- RoadRunner (RR): Accelerated Architecture
  - *First to Deliver PetaFlop Performance*
  - Precursor to Todays Accelerated Computers
- Cielo & Trinity: Many-Core Systems
- *Required Rewriting Weapons Codes*

LANL is one of the leaders of computing change with RoadRunner & Trinity.



Sustain & Extend ASCI/ASC Application Investments Made During Post-Testing Era

Multi-Physics Codes

- Eulerian Application Project (EAP) Codes
  Direct Eulerian / Adaptive Mesh Refinement (AMR)
- Lagrangian Application Project (LAP) Codes
  Lagrange / Arbitrary Eulerian-Lagrangian (ALE)

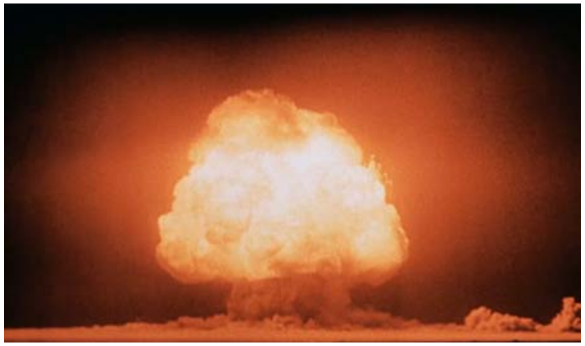
Single-Physics Codes

- Transport Codes: PARTISN & IMC

Weapons Science Codes

- Plasma Kinetic Code: VPIC

First US Test: Trinity 1945



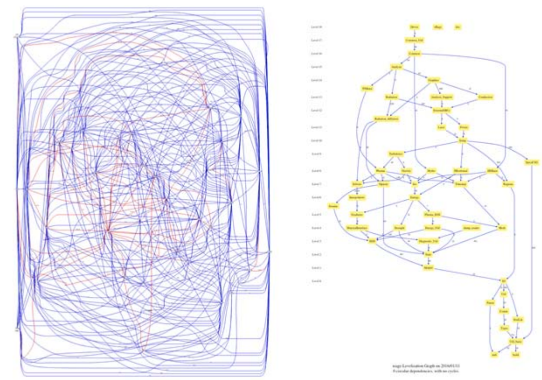
Last US Test: Divider 1992

Restructure EAP Code Base for Maintainability, Performance & Portability

Balance Ongoing Investments

- *First ~20 ASCI/ASC Years Focused Almost Exclusively on Adding Physics*
- Sustained Simulation Capability Requires
  - Code Maintenance & User Support
  - New Code Physics & Capabilities
  - Performance Upgrade & Modernization

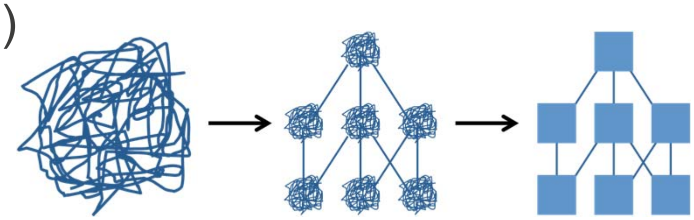
xRage Dependency Graphs



October 2014 January 2016

EAP Code Restructure/Packagization

- Refactor Code; Organize into Packages (Done)
- Upgrade Build/Test System to Support Packages (Done)
- Upgrade Packages: Internal & Interfaces (Ongoing)
- Enhance Package Performance (Ongoing)



Restructure Code; Upgrade Packages

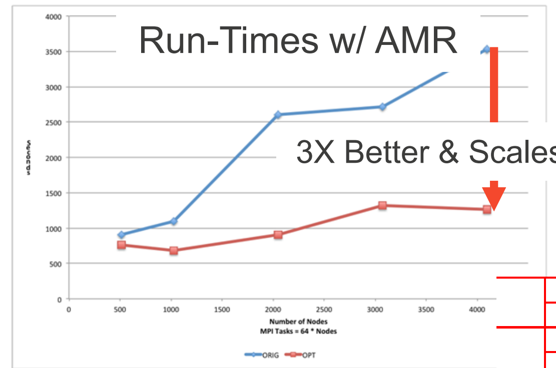
EAP Performance Improvements on Trinity Knights Landing (KNL) Architecture

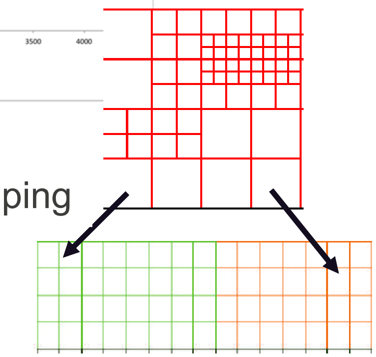
AMR Communication
- RMA All-to-All and Caching Strategy
- 3X Overall Performance; Good Strong Scaling

Mesh-Mesh Mapping
- Better Techniques for Mesh Intersections
  At Scale was Approaching >50% of Run-Time
- 7.5X Improved Mesh Intersection

Data Input & Output
- Burst-Buffer Technology (Cray DataWarp)
- Hierarchical I/O (HIO) for Trinity & Beyond
- 15X Improved I/O (App Write >300 GB/s)

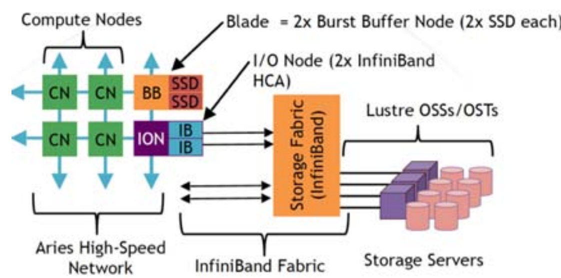
Routinely run at 4K KNL nodes with demonstrations across full Trinity partition.





Mesh-Mesh Mapping

Burst-Buffer Infrastructure



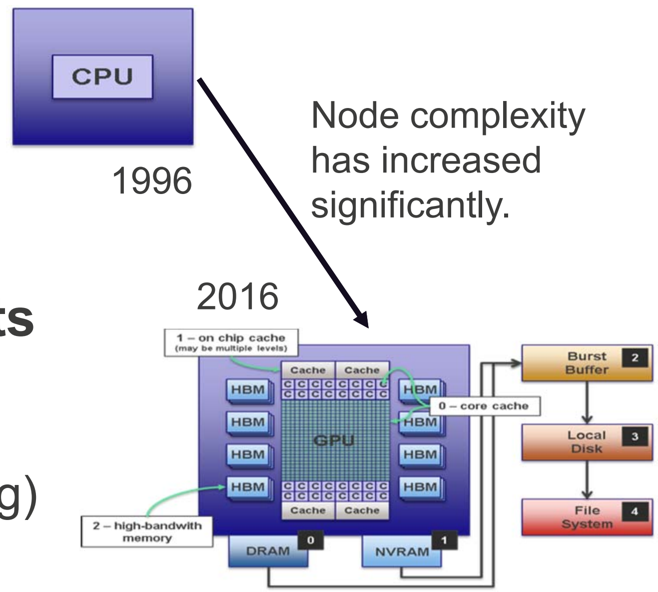
Portable Performance on Emerging Architectures is Also Important for EAP Codes

Portability Across Diverse Architectures

- Integrate Flexible, Multi-Language Support
  - Supports Portability Layers, e.g., Kokkos/Raja
  - Supports Package Performance Enhancements
- Platform Targets: Trinity (KNL) & Sierra (Power+GPU)

General Node-Level Performance Improvements

- Also Supports Portability for Emerging Architectures
- Identify & Document Parallel Iteration Patterns (Done)
- Enhance Vectorization: Assist/Force Compiler (Ongoing)
- Leverage Threading in Key Packages (Prototyping)



EAP codes are undergoing a significant transformation, e.g., restructuring & targeted performance enhancements, to improve viability and performance on various emerging architectures.

Sustain & Extend Investments in LAP Code Base for Future Simulation Capability

Historical & Ongoing Investments

- *First ~20 ASCI/ASC Years Focused Almost Exclusively on Adding Physics*
- LAP Born Levelized Together with a Database & Source-Source Translator
- Extend Investments via Upgrades; Target New Capabilities

Code Enhancements via Translation

- Abstraction Between Developer & Executable
- Updated Source-Source Translator: Rewrite
  - Enables Application-Specific Analysis
  - Awareness of Specific Iteration Patterns
  - Optimize Parallelism for Each Architecture

LAP check-out to run procedure including Rewrite translator.

LAPs historical adoption of a database & source-source translator provide a spring board for agile code modernization.

LAP is Making Key Investments in Code Portability & Code Performance at Scale

Better LAP Code Portability

- Manage Dependencies Across Platforms & Architectures
  Integrate Spack to Manage External Libraries
- Package Code & Dependencies via Containers
  Isolate Execution Environment from System Changes
- Early Targets: LANL BW/HW/KNL & Sierra (Power+GPU)

Enhance Performance Especially at Scale

- Significant Algorithmic Improvements
  Multi-Material ALE, Mesh-Mesh Mappers, etc.
- Modernize I/O, Visualization & Analysis
  In Situ: Avoid Large-Scale I/O Where Possible
  Integrate ParaView Catalyst for Images/Movies
- Exploit Burst-Buffer Technology
  HIO Library Under HDF5 for Trinity & Beyond

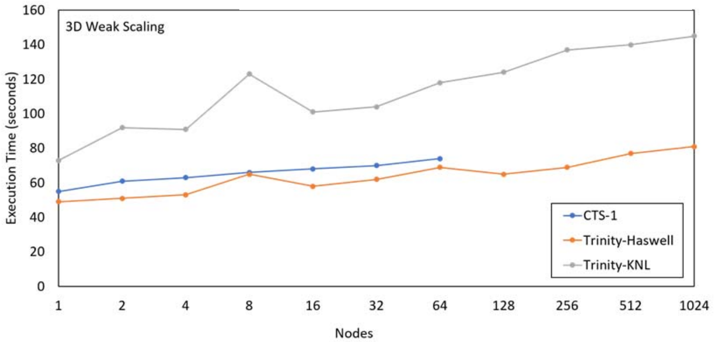
Little Boy

Previous & Ongoing Investments in Single-Physics Codes: PARTISN

PARTISN

- Time-Dependent, Deterministic (Sn) n/γ Transport Multi-Group Energy, Structured Meshes
- Parallelism Aligned with Physical Characteristics

3D Weak Scaling
MPI + OpenMP Threads



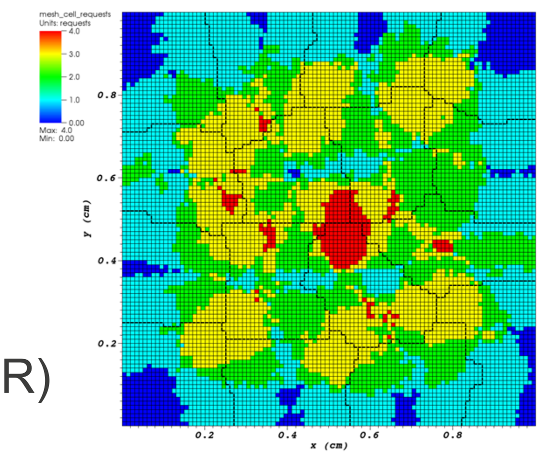
Past Performance Efforts

- Distributed Memory (MPI) Over Space & Energies
- SIMD (Vector Units) Over Angles (RoadRunner)
- Hybrid Domain Decomposition / Replication (Cielo)
- Shared Memory (OpenMP) Over Energy Groups (Trinity)

Current Performance Investments

- Cache Management for Threading & Vectorization (KNL)
- Asynchronous Communication to Hide Latency (Trinity)
- Early Plans/Efforts for GPU Porting (Sierra)

Previous & Ongoing Investments in Single-Physics Codes: IMC



Particle Mesh-Cell Queries
Hybrid DD/Rep Algorithm

IMC (Implicit Monte Carlo)

- Time-Dependent, Monte Carlo Thermal Radiation Transport
  Multi-Group Energy, Structured & AMR Meshes
- Approach: Algorithmic, Load Balance & Run-Time Efforts

Past Performance Efforts

- Distributed Memory (MPI) Spatial Domain Decomposition
- Hierarchical Parallelism for Heterogeneous Architectures (RR)
- Low-Order Acceleration & Improved Load Balance (Trinity)

Current Performance Investments

- Hierarchical, Asynchronous Computing
  On-Node, Many-Core Threading (Trinity KNL)
  GPU Threads, Overlap Communication/Computation (Sierra)
- Hybrid Domain Decomposition/Replication Algorithm

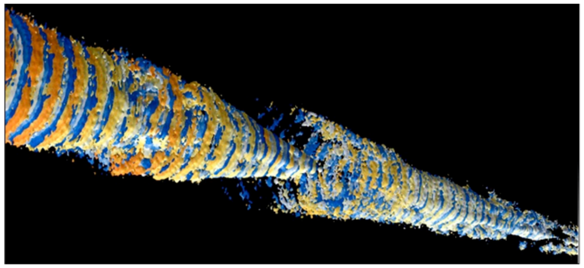
Weapons Science Codes Often Lead the Way To Performance on Emerging Architectures

VPIC (Vector Particle-in-Cell)
- 3D Explicit, Relativistic, Charge-Conserving Electromagnetic Particle-in-Cell (PIC) Code
  - Single Precision, Structured Meshes

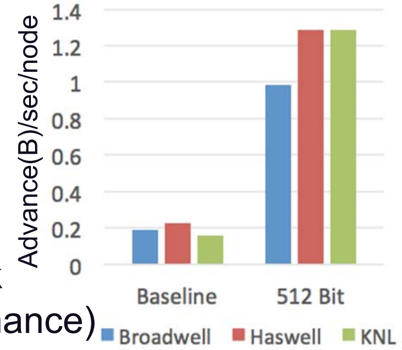
Past Performance Efforts
- Distributed-Memory Parallel (MPI)
- Shared-Memory Parallel (OpenMP or Pthreads)
- Explicit Short/Wide Vectors using Hardware Intrinsics

Current Performance Investments
- New Data Layout for Memory Locality
  - Intermediate Results at 7% Peak Performance
  - Expect to Exceed 15% Peak Performance (Trinity KNL)

Self focusing electron wave from laser-plasma interaction in ICF hohlraum.



VPIC Trinity Performance
Large Production Run



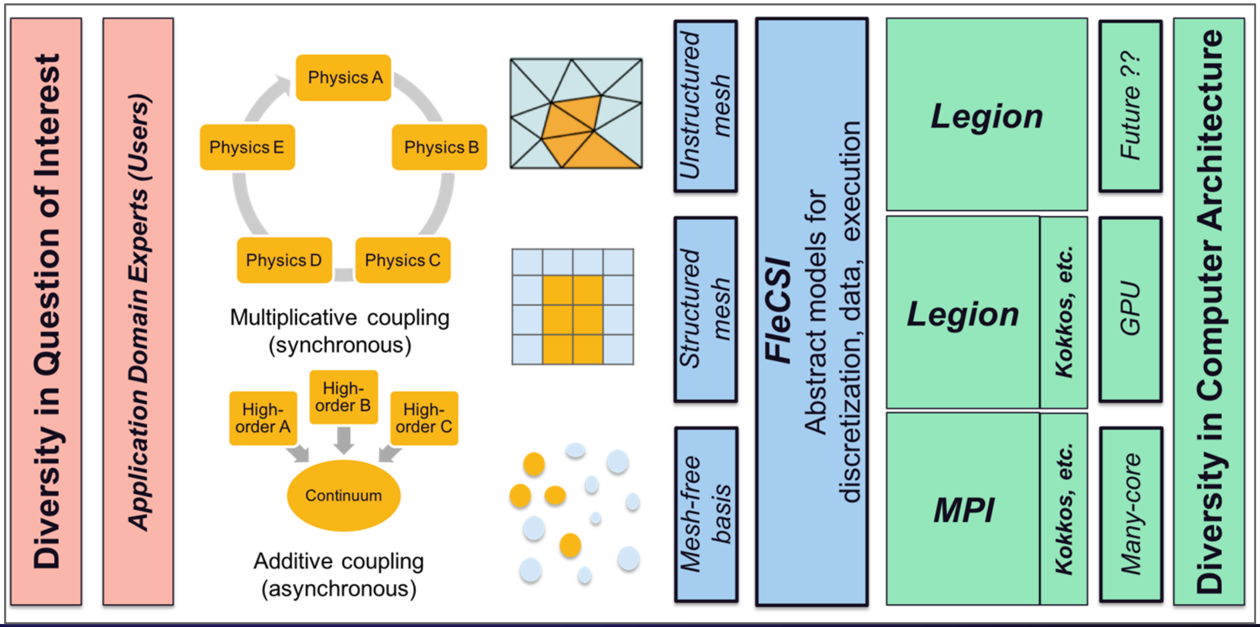
0.35TF/s/node (7% Peak Single-Precision Performance)

Target New Investments in Next Generation Application Codes for Next-Era Architectures

Ristra: LANLs New Multi-Physics Code

- Clean-Sheet Start; Short-Term Effort; *High-Risk, High-Reward*
- Annual Investment Comparable to EAP & LAP
- Separate Physics from Computer Architectural Concerns

Rely on trusted physics algorithms & mesh topologies; place risk in code structure & novel run-time approaches.



Early Ristra Efforts Have Produced a Number of Hierarchical Physics Capabilities

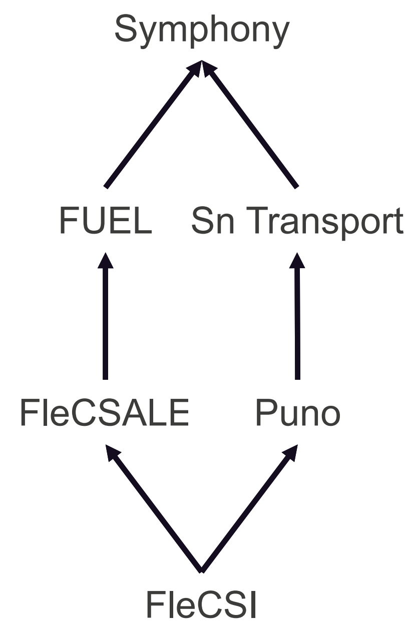
FleCSALE
- Simple, Single-Material (Gas) Hydrodynamics Code

FUEL
- Complex, Multi-Material Hydrodynamics Code
- Gas & Solid Materials with Strength with Realistic EOS

FleCSPH
- Smooth-Particle Hydrodynamics (SPH) Code
- Tree-Based Search for Near Neighbor Information

Puno
- P1 Thermal Radiation Code
- Low-Order Portion of LO-HO Multiscale Algorithm

Symphony (Recent Start)
- Coupled-Physics, Radiation Hydrodynamics Code



The Ristra Strand

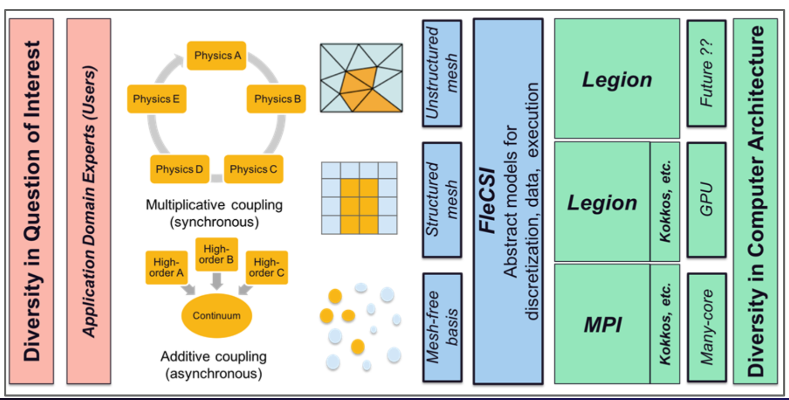
Ristra Places Risk in Multiple, Novel Run-Time Approaches Enabled by New Abstraction Layer

Ristra Run-Time Approaches

- Message Passing Interface (MPI) for Many-Core
- Legion: Data-Centric, Task-Based Programming Model
  Asynchronous Execution for Compute Efficiency

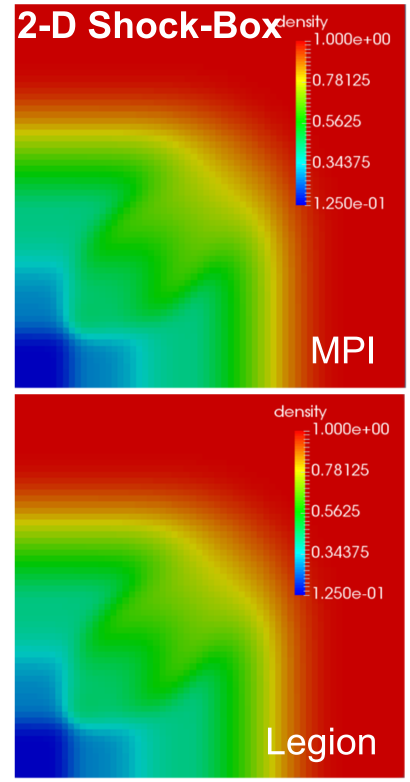
Ristra Abstraction Layer

- Flexible Computer Science Infrastructure (FleCSI)
- Enables Separation of Physics from Computer Architectures
  Write Physics Once, Specialize Run-Time as Needed



Continuous co-design development paradigm between FleCSI/algorithms & FleCSI/run-time libraries.

FleCSALE



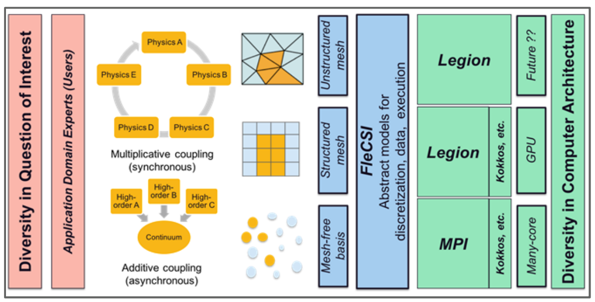
LANL Strategy for Future Simulation Capability

Leverage Distant/Recent Past Investments

- Sustain/Extend EAP & LAP Investments

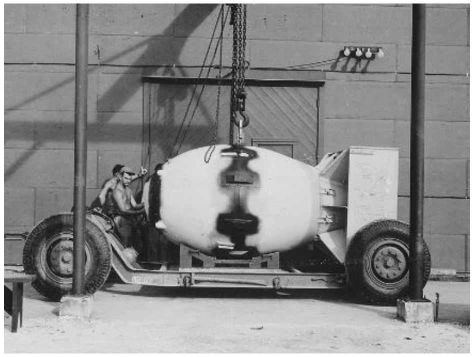
Target New Investments for Future Simulation

- Ristra: Next-Generation Physics Code



First US Test: Trinity 1945

Fat Man



HPC Simulation Science Supporting the US Stockpile: ASCI 1995, ASC 2004

Questions?

CRAY
TRINITY
Alliance for Computing at Extreme Scale
NNSA
Los Alamos
Sandia National Laboratories

Abstract

Early in the nuclear era, the US relied mainly upon theory and experiment to design, demonstrate and deploy with confidence its nuclear weapons stockpile. From the Manhattan Project to the end of the 20^{th} century, simulation science played an important, but limited role in shaping the US stockpile. At the cessation of the integrated-testing era in the early 1990s, the US elevated the role of simulation to complement theoretical and experimental sciences in support of its nuclear security enterprise. The ASCI/ASC Program was the manifestation of this new investment in simulation science supporting DOE/NNSA, providing novel computing platforms together with multi-physics simulation tools. Today, the ASC Program is challenged to retain the physics fidelity and modeling approaches embodied in the existing multi-physics codes in a new generation of simulation tools. Moreover, these new tools must be adaptable and performant on the many novel architectures expected during the ongoing evolution of high performance computing platforms. This presentation will discuss the challenges in leveraging past and current investments for LANL's future multi-physics, large-scale simulation capability.