Emerging Computing Landscape and Opportunities

PSAAP Pre-proposal Meeting

David Richards
(with help from friends)

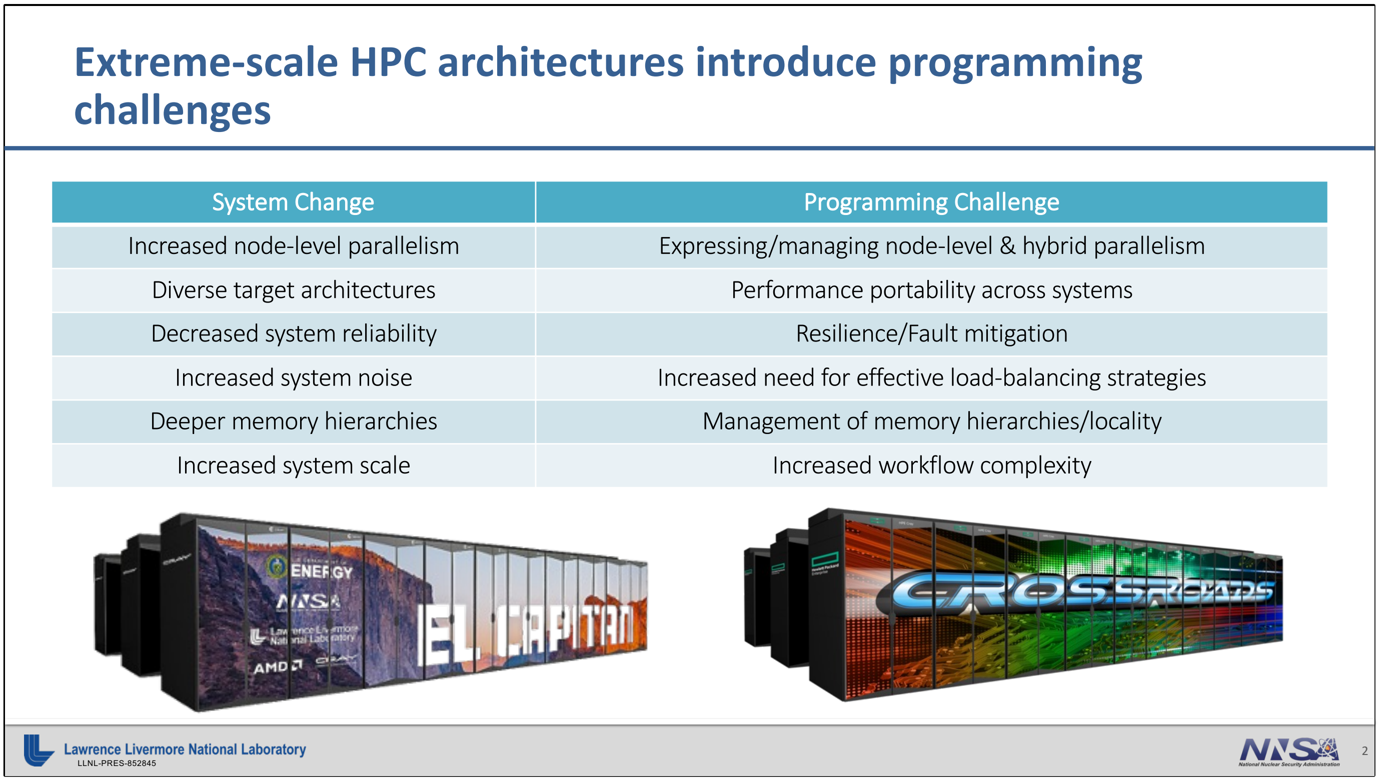
August 8-9, 2023

LLNL-PRES-852845
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

For the last several years I've been involved in helping HPC applications at LLNL prepare for advanced architectures: First, Sierra, and now El Capitan.

I've been asked to share with you an overview of some of the CS areas that will be coming out in the RFI. Additional details on some specific technical areas will be presented in follow-on presentations throughout the day.



Extreme-scale HPC architectures introduce programming challenges

System Change	Programming Challenge
Increased node-level parallelism	Expressing/managing node-level & hybrid parallelism
Diverse target architectures	Performance portability across systems
Decreased system reliability	Resilience/Fault mitigation
Increased system noise	Increased need for effective load-balancing strategies
Deeper memory hierarchies	Management of memory hierarchies/locality
Increased system scale	Increased workflow complexity

Today’s extreme scale architectures come with many programming challenges as shown here.

New node designs include manycore and multicore designs with numerous hyperthreaded processes. GPU nodes have thousands of effective cores of parallel processing. Both GPUs and CPUs continue to require more and more parallelism to use them efficiently. The enormous amount of compute capability per node and decreasing amounts memory per unit of compute are also challenging traditional MPI-only decompositions and load balancing techniques.

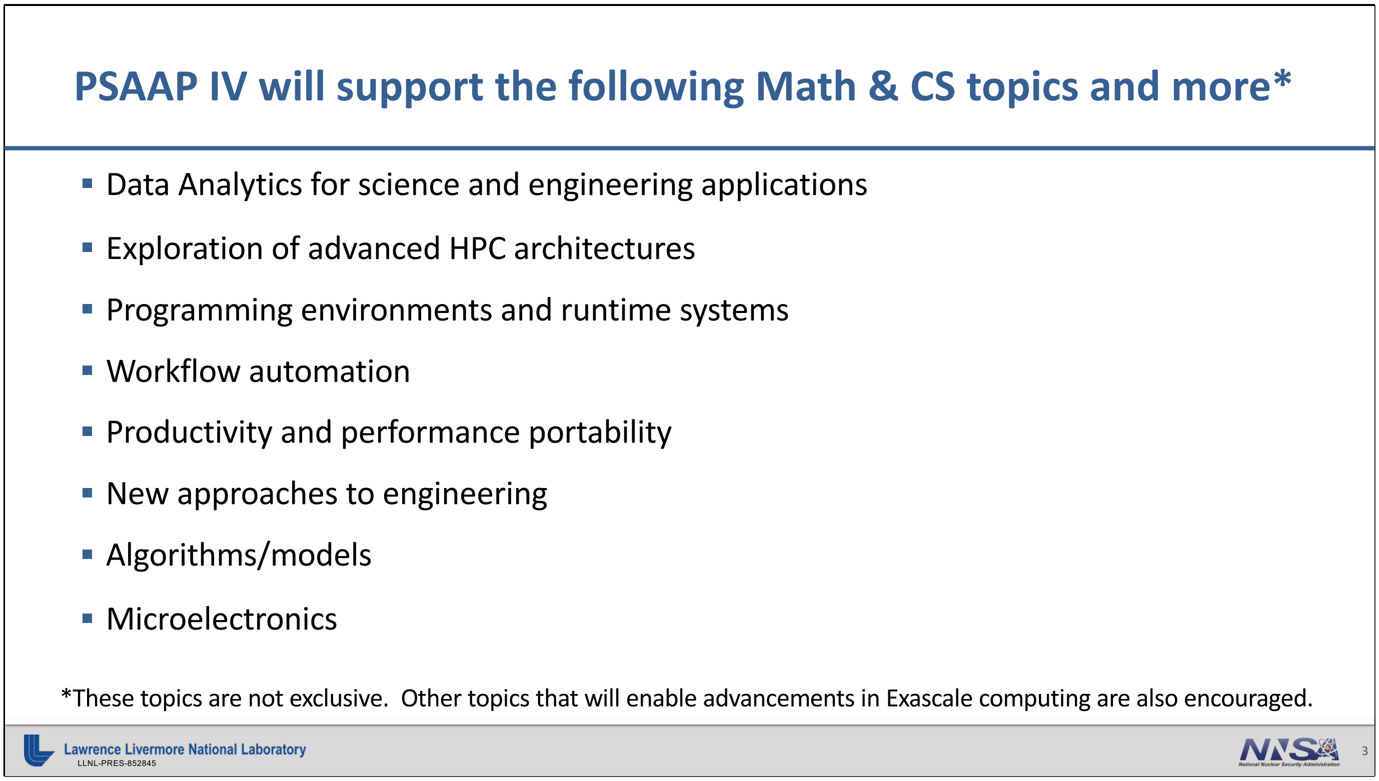
As a greater variety of CPUs, GPUs, and potentially novel, specialized, or disaggregated hardware become available, maintaining portability between systems becomes more and more difficult.

With increased component count comes the potential for reduced reliability. Vendors have done an amazing job of keeping reliability high, but applications still must be prepared with fault mitigation strategies such as fast checkpoint/recovery schemes.

There is a potential for increased system noise. Most centers have been able to minimize these effects by dedicating cores or threads to system tasks, but effective load balancing is still a significant challenge for many applications.

Multi-level memory management is becoming the norm with high bandwidth memories and non-volatile memories becoming more popular.

And increased system scale will result in increase workflow and more complex multi-tasking schemes.



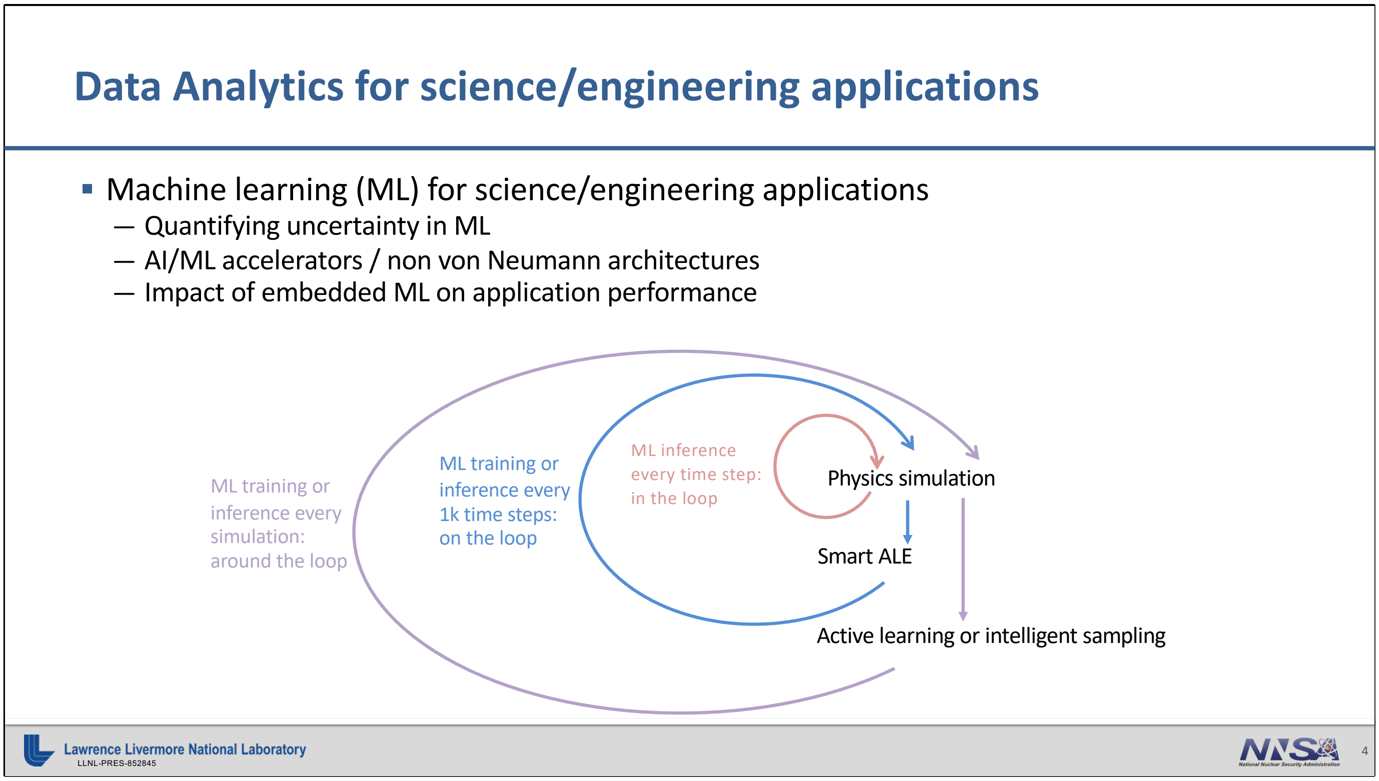
PSAAP IV will support the following Math & CS topics and more*

- Data Analytics for science and engineering applications
- Exploration of advanced HPC architectures
- Programming environments and runtime systems
- Workflow automation
- Productivity and performance portability
- New approaches to engineering
- Algorithms/models
- Microelectronics

*These topics are not exclusive. Other topics that will enable advancements in Exascale computing are also encouraged.

This is a list of many of the CS topics of interest for the PSAAP IV program. This list is not exclusive. We will entertain other topics that are in the spirit of advancing extreme-scale HPC. These topics are offered as examples of topics of interest to the Labs.

I will talk about each of these in more detail in the slides to follow. I will be giving examples of prototype work at the National Labs in these areas. This is not to say that these are all solved problems … because they are not. The examples are intended to give you a feel for the type of work that is needed to solve difficult national security problems.



Data Analytics for science/engineering applications

- Machine learning (ML) for science/engineering applications
  - — Quantifying uncertainty in ML
  - — AI/ML accelerators / non von Neumann architectures
  - — Impact of embedded ML on application performance

ML training or inference every simulation: around the loop
ML training or inference every 1k time steps: on the loop
ML inference every time step: in the loop
Physics simulation
Smart ALE
Active learning or intelligent sampling

Lawrence Livermore National Laboratory
LLNL-PRES-852845

4

Data analytics for science and engineering applications is an obvious topic of interest.

The pace of change in AL/ML is so rapid that it is hard to predict what kind of impact we will see on science and engineering applications. For PDE-based simulation codes that are very common in the NNSA we have identified at least three levels at which AI/ML techniques can be integrated into our modeling and simulation codes.

First, ML inference might be called “in the loop” one or more times every time step. Such inference might take the place of a more expensive physically-based model.

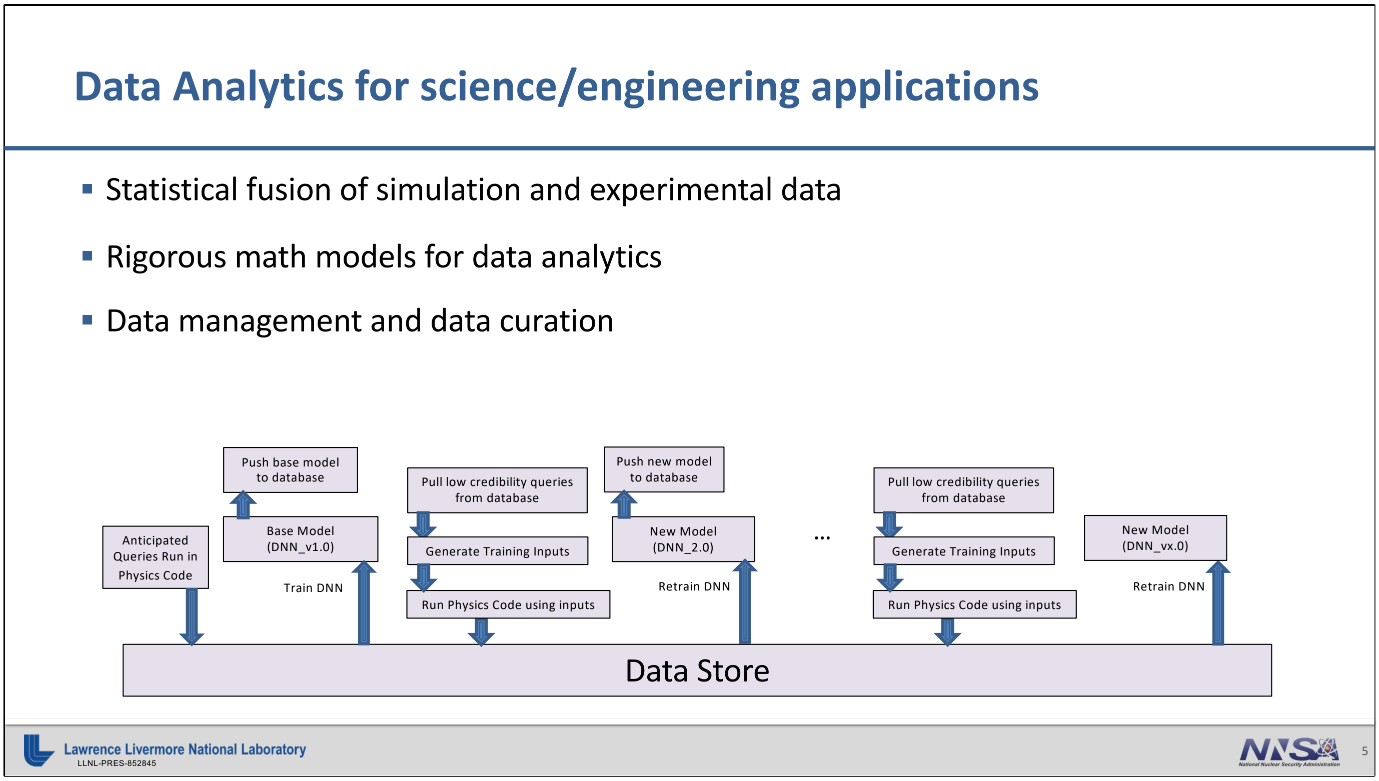
Second, ML training or inference might be called “on the loop” every 10^3 time steps or so. Such training might attempt to respond to the trajectory of the simulation. Although the cost of such training is amortized over many time steps, it would still have to e fairly low-cost to avoid substantially impacting the overall simulation performance.

Finally, ML training or inference might be called “around the loop” every simulation. This is the mode that is likely least sensitive to training performance.

For all of these modalities, it will be important to be able to quantify uncertainties.

Some work is currently going on in NNSA to explore offloading so me of this training and/or inference to dedicated special purpose accelerators. The latency imposed by the data motion for the offload will be a key metric in determining whether such accelerators can improve simulation performance.

Understanding how embedded ML impacts simulation performance is another area of active research.



Data Analytics for science/engineering applications

- Statistical fusion of simulation and experimental data
- Rigorous math models for data analytics
- Data management and data curation

Anticipated Queries Run in Physics Code

Push base model to database

Base Model (DNN_v1.0)

Train DNN

Pull low credibility queries from database

Generate Training Inputs

Run Physics Code using inputs

Push new model to database

New Model (DNN_2.0)

Retrain DNN

…

Pull low credibility queries from database

Generate Training Inputs

Run Physics Code using inputs

New Model (DNN_vx.0)

Retrain DNN

Data Store

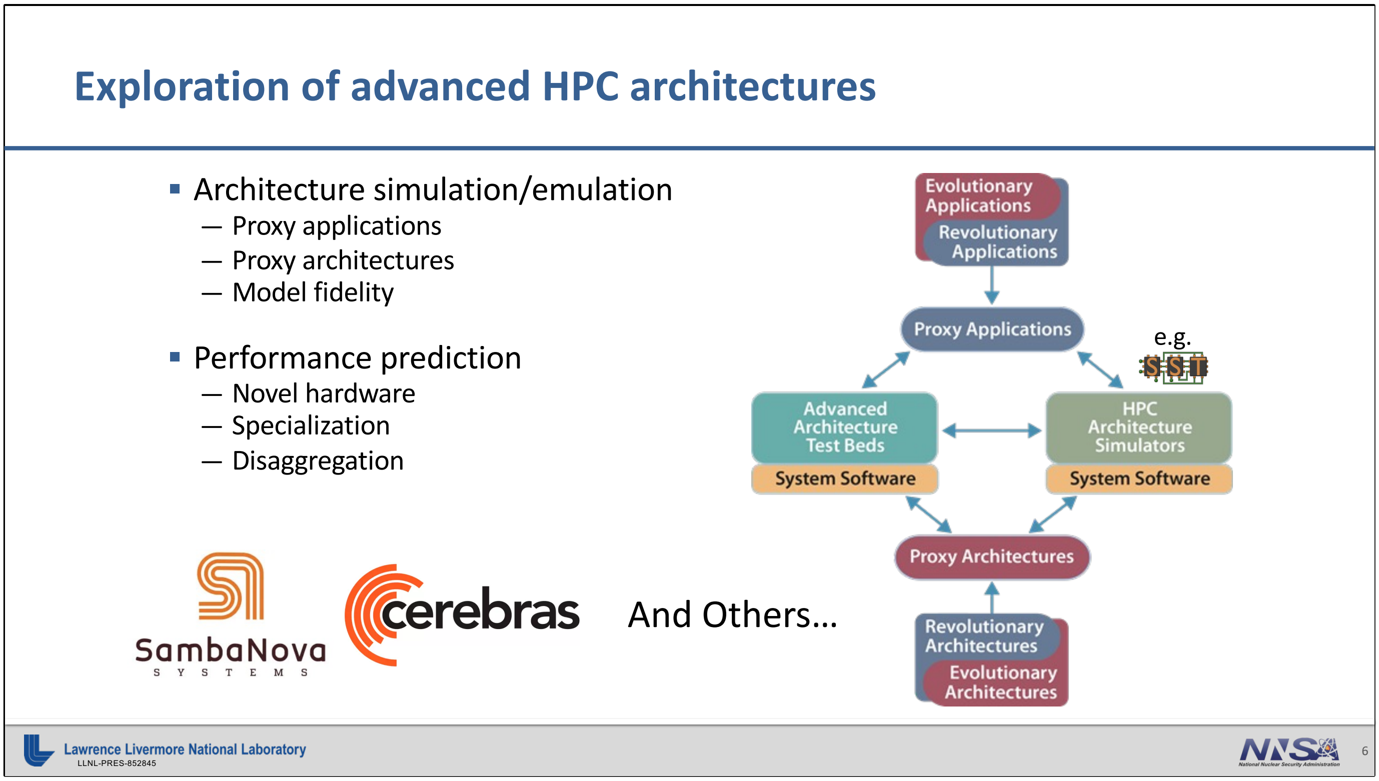
Lawrence Livermore National Laboratory
LLNL-PRES-852845

5

One of the areas of interest is combining or statistically fusing simulation and experimental data. The national laboratories have a role of stockpile stewardship where the goal is to certify nuclear weapons design without underground testing. Many of the components and subassemblies are individually tested, but some subassemblies and the entire system can not be tested and therefore, we must simulate and certify these systems using calibrated models.

Unlike conventional machine learning where you have large amounts of data, and it is okay to classify or identify objects with an accuracy in the 80-90% level, the DOE has high consequence applications for machine learning that will require 5 nines of reliability – only making a mistake 1 out of 10000 times or more. We need machine learning approaches that work on far less data, maybe taking advantage of Generative Adversarial Networks to generate synthetic data along with the real data. We need to quantify uncertainty in ML, and know when a machine learning algorithm is interpolating vs. extrapolating. We need a rigorous math model that goes with the ML to be able to explain the results.

Data management and data curation are also important topics. The diagram here shows a timeline from right to left and indicates data transactions to and from a data store. Data access must be low cost, but we also want maintain provenance of data and trained models, eg., for reproducibility.



Exploration of advanced HPC architectures

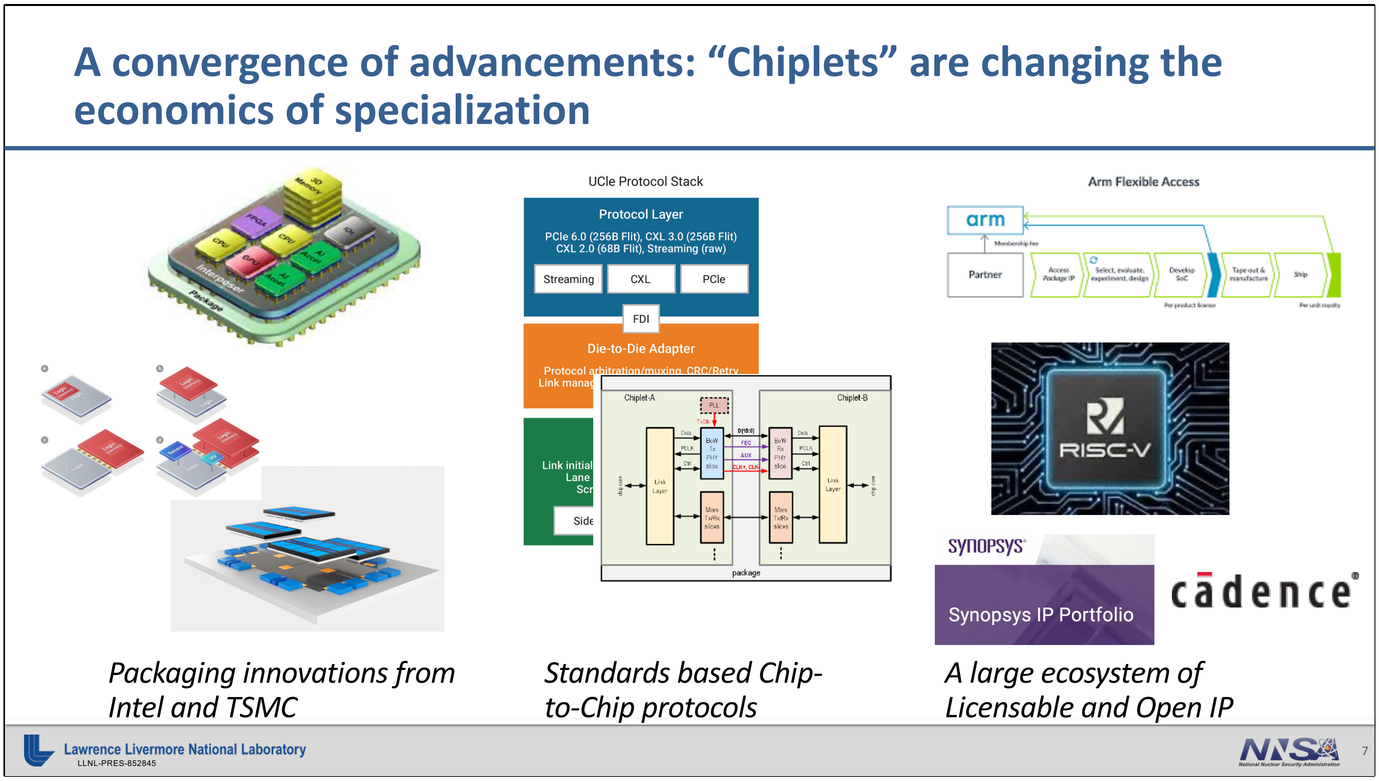
- Architecture simulation/emulation
  - Proxy applications
  - Proxy architectures
  - Model fidelity
- Performance prediction
  - Novel hardware
  - Specialization
  - Disaggregation

And Others…

SambaNova
SYSTEMS
cerebras
Evolutionary Applications
Revolutionary Applications
Proxy Applications
e.g.
SST
Advanced Architecture Test Beds
System Software
HPC Architecture Simulators
System Software
Proxy Architectures
Revolutionary Architectures
Evolutionary Architectures

Co-design of advanced HPC architectures is a topic of interest. As the figure shows, co-design is where evolutionary and revolutionary architectures and applications come together to create new HPC designs. This includes the design of memory, CPU/GPU configurations, and message passing and network protocols. Future HPC systems might also include various specialized hardware such as FPGAs, DSPs, network accelerators, AI accelerators, graph accelerators, etc. The ability to simulate HPC architectures using tools such as the Structural Simulation Toolkit (SST) is important when evaluating architectures. At the same time verifying and validating models on advanced architecture testbeds with the same or similar system software in necessary. The labs have a number of proxy applications that can be tested on these new architectures. The goal is to predict performance and perform design trade studies without building a full scale system.

Understanding the performance impact of memory hierarchies, and possibly disaggregated systems is also of interest.



A convergence of advancements: “Chiplets” are changing the economics of specialization

Packaging innovations from Intel and TSMC

Standards based Chip-to-Chip protocols

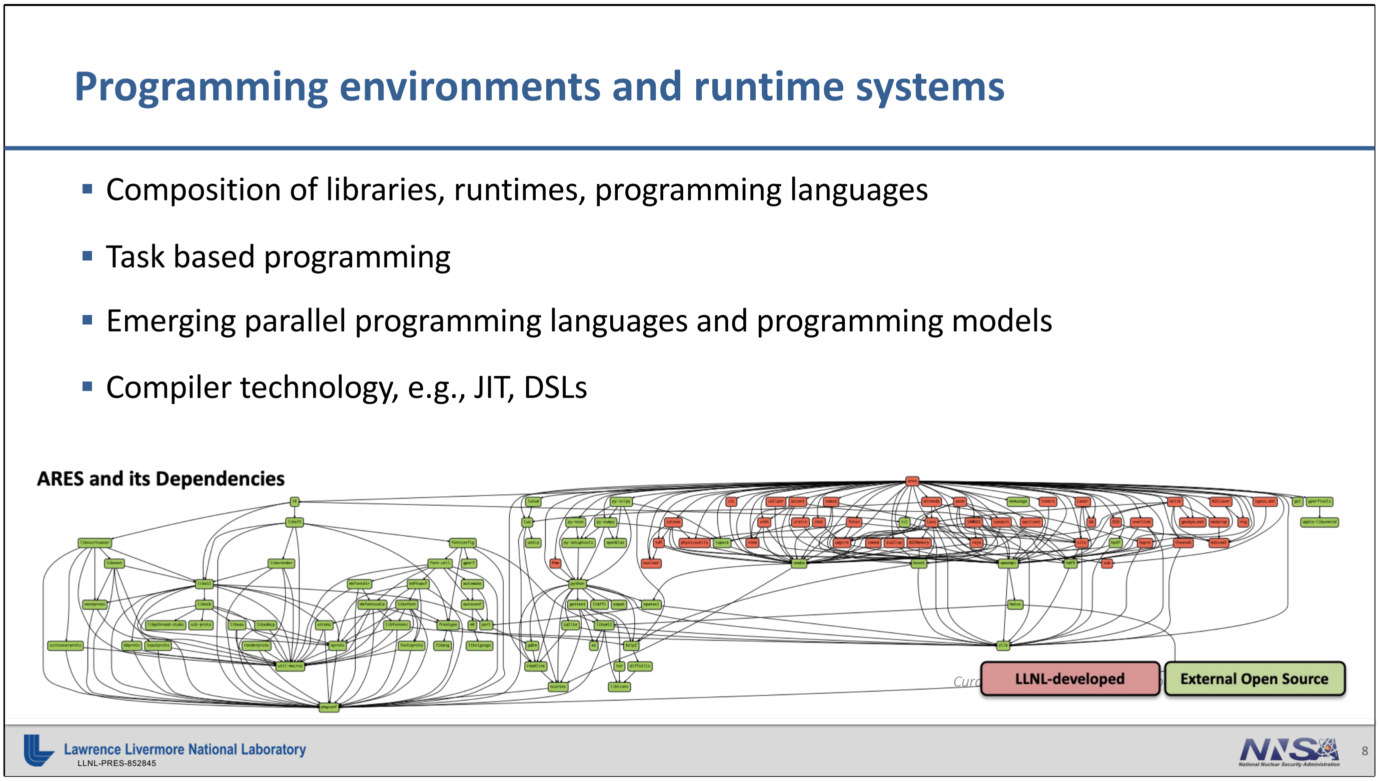
A large ecosystem of Licensable and Open IP

Lawrence Livermore National Laboratory
LLNL-PRES-852845

NNSA
National Nuclear Security Administration

7

The rise of chiplet technology may be an enabling technology for greater availability of specialized hardware.



Programming environments and runtime systems

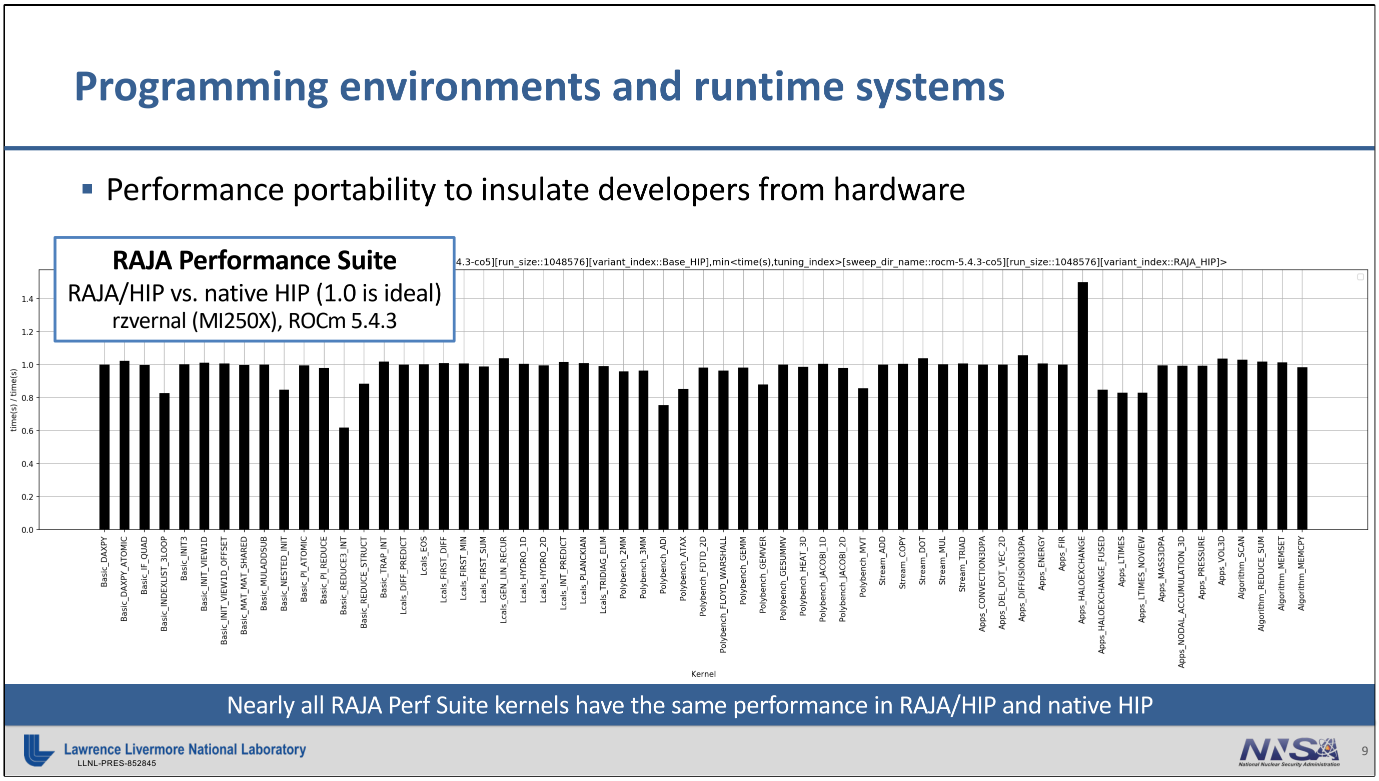
- Composition of libraries, runtimes, programming languages
- Task based programming
- Emerging parallel programming languages and programming models
- Compiler technology, e.g., JIT, DSLs

ARES and its Dependencies

🟥 LLNL-developed
🟩 External Open Source

Composition of programming environments and runtime systems is an important topic of interest. The diagram below shows the library dependency graph of one of the Multiphysics codes at LLNL. This gives some idea of the complexity of these applications, and the number of libraries, both lab-developed, and external open source, that they rely on. These codes rely on multiple languages and programming models, and any programming environment and runtime systems must interoperate in these kinds of complex builds.

The labs also have 10s of millions of lines of legacy, validated codes, and technologies that can integrate with those legacy code bases are especially welcome.



Programming environments and runtime systems

- Performance portability to insulate developers from hardware

RAJA Performance Suite
RAJA/HIP vs. native HIP (1.0 is ideal)
rzvernal (MI250X), ROCm 5.4.3

Nearly all RAJA Perf Suite kernels have the same performance in RAJA/HIP and native HIP

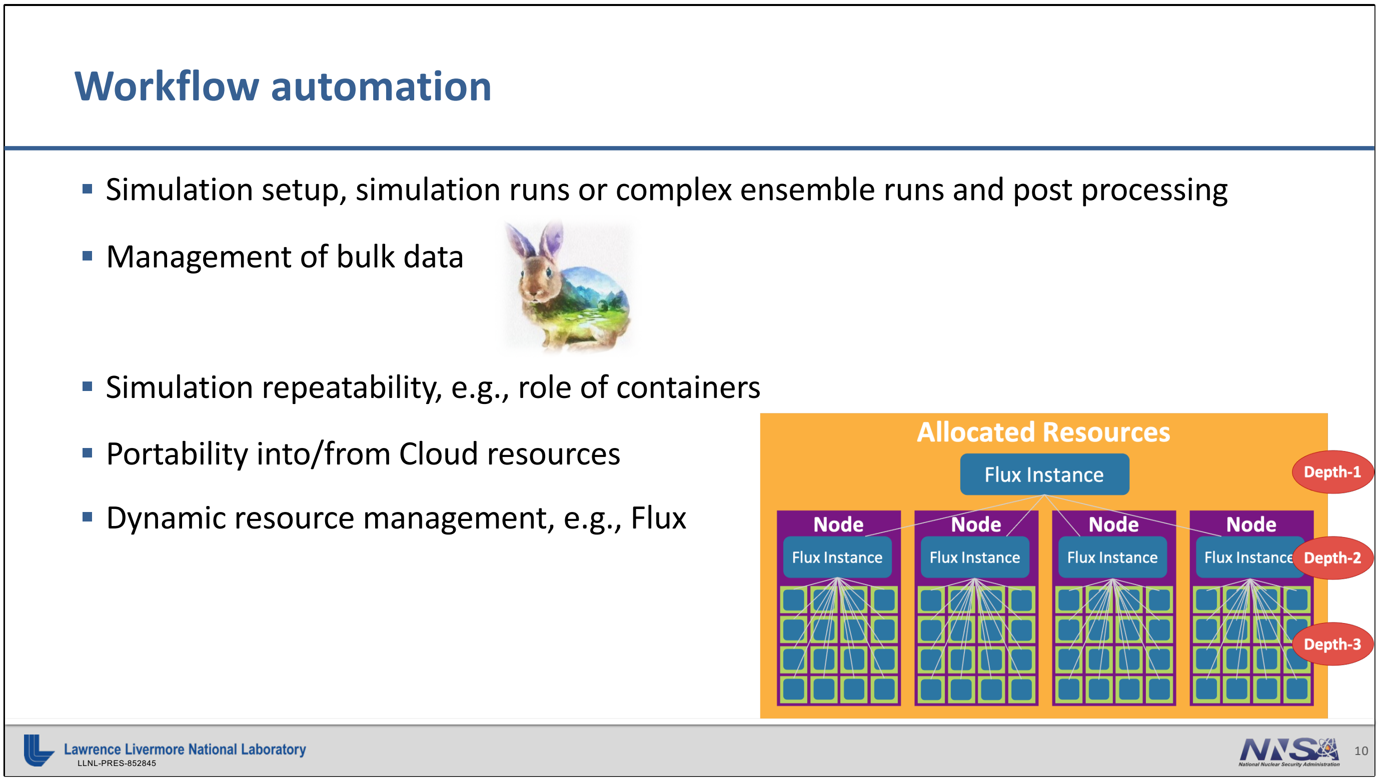
Lawrence Livermore National Laboratory
LLNL-PRES-852845

National Nuclear Security Administration

9

Portability abstractions that insulate developers from hardware and allow them to write code that performs well on multiple hardware platforms are important to NNSA.

Raja has proven to be a very effective portability abstraction. This graph shows the performance of test kernels in the RAJA performance suite as implemented in RAJA and compiled with the HIP back end for AMD GPUs and implemented in native HIP. The performance is nearly identical for most kernels producing a ratio of 1.0. Note that similar comparisons for RAJA/CUDA on Nvidia or for Kokkos on AMD or Nvidia would produce very similar results.



Workflow automation

- Simulation setup, simulation runs or complex ensemble runs and post processing
- Management of bulk data
- Simulation repeatability, e.g., role of containers
- Portability into/from Cloud resources
- Dynamic resource management, e.g., Flux

Workflow automation is another topics of interest. Simulation setup is still difficult where ensemble runs are typically the norm, with a post processing step that looks at the analysis.

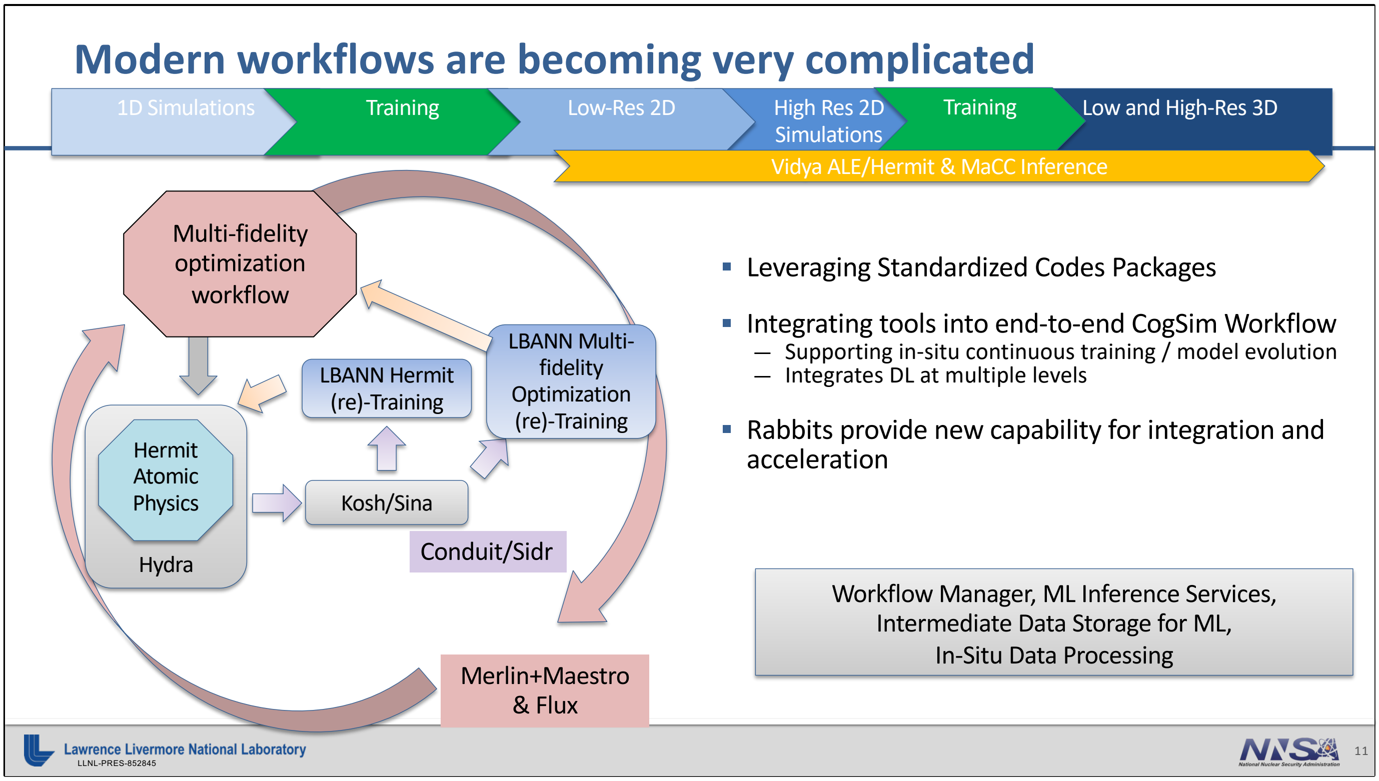
Management of bulk data is another area of interest. NNSA has been developing various data warehouse strategies that allow in-situ passing of data between applications.

The role of containers in HPC is an area of interest. Can containers be used to store not only the simulation executable but also the data, so that you have a history and a provenance associated with they runs.

Interoperability and portability between cloud resources and NNSA HPC centers is of interest, including how compute cloud services contribute to workflows.

Dynamic management of resources is of interest, especially as we contemplate systems with specialized or disaggregated hardware.

Rob Neely will have more to say about workflows later today.



Modern workflows are becoming very complicated

1D Simulations → Training → Low-Res 2D → High Res 2D Simulations → Training → Low and High-Res 3D

Vidya ALE/Hermit & MaCC Inference

Multi-fidelity optimization workflow

LBANN Hermit (re)-Training

LBANN Multi-fidelity Optimization (re)-Training

Hermit Atomic Physics

Hydra

Kosh/Sina

Conduit/Sidr

Merlin+Maestro & Flux

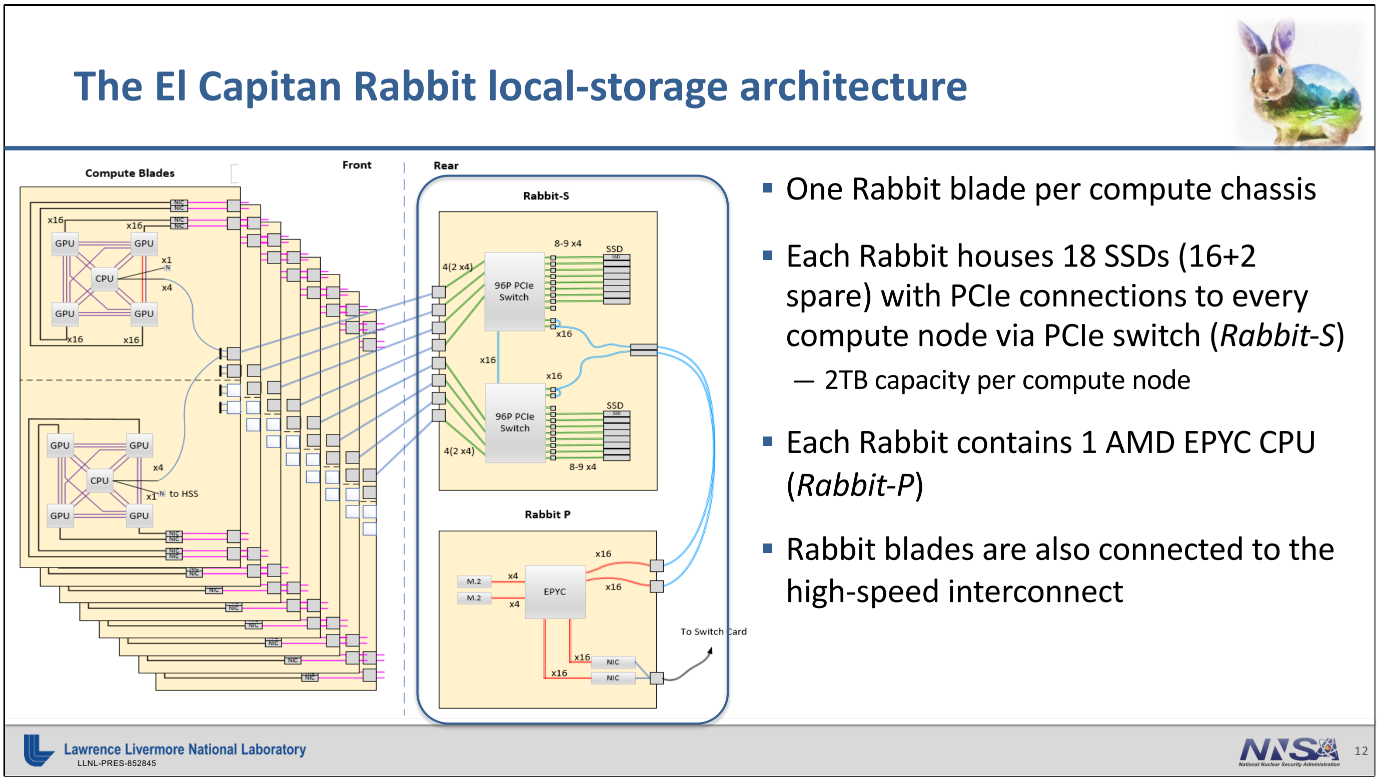
- Leveraging Standardized Codes Packages
- Integrating tools into end-to-end CogSim Workflow
  - Supporting in-situ continuous training / model evolution
  - Integrates DL at multiple levels
- Rabbits provide new capability for integration and acceleration

Workflow Manager, ML Inference Services,
Intermediate Data Storage for ML,
In-Situ Data Processing

Lawrence Livermore National Laboratory
LLNL-PRES-852845

NNSA National Nuclear Security Administration 11

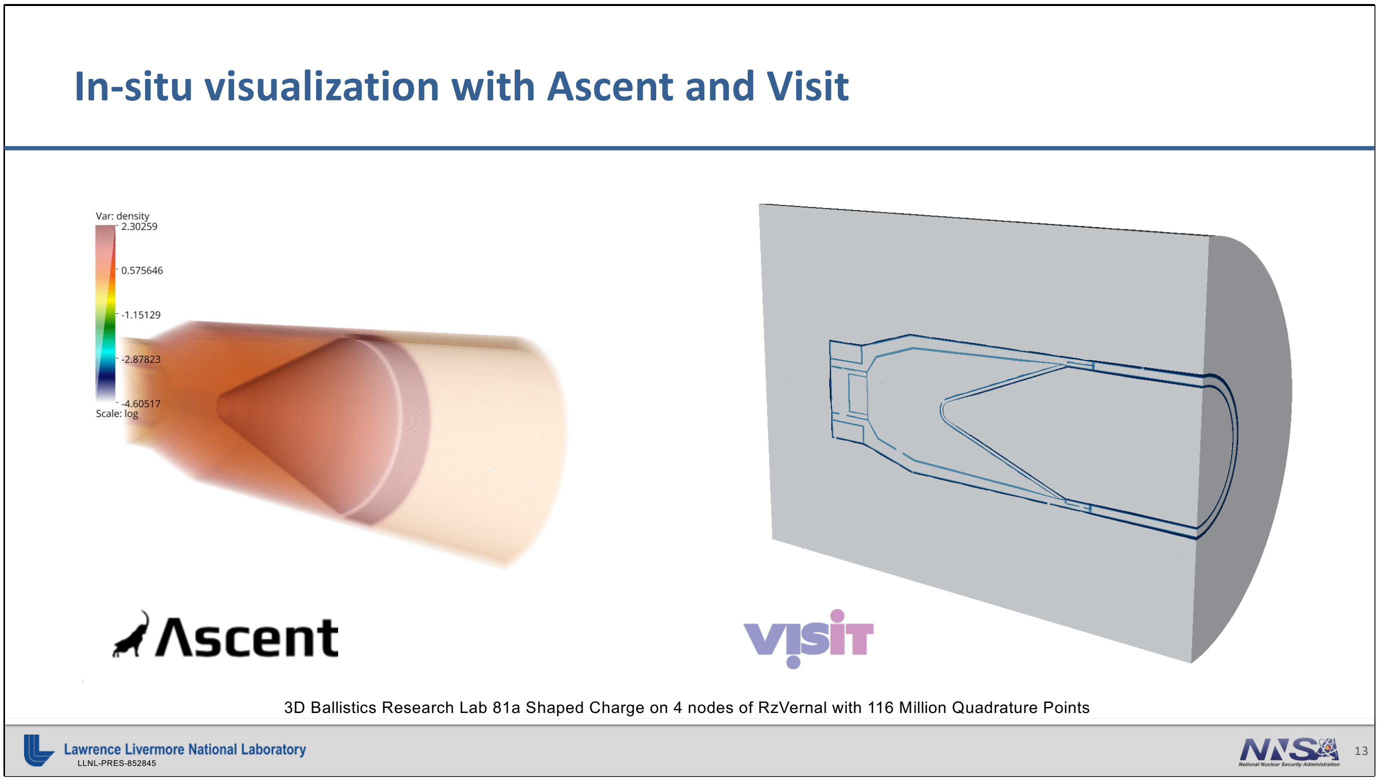
As system scale increases, workflows are becoming more and more complicated. It is now common to see workflows with multiple simulation and modeling tools, frequently operating at multiple fidelities in 1D, 2D, and 3D. Workflows also include ML/AI optimization loops with components that are trained from simulation results and help steer simulation ensembles as they are trained.



The El Capitan Rabbit local-storage architecture

- One Rabbit blade per compute chassis
- Each Rabbit houses 18 SSDs (16+2 spare) with PCIe connections to every compute node via PCIe switch (*Rabbit-S*)
  - — 2TB capacity per compute node
- Each Rabbit contains 1 AMD EPYC CPU (*Rabbit-P*)
- Rabbit blades are also connected to the high-speed interconnect

El Capitan will feature a near-node local storage architecture called rabbits. Each rabbit blade will make direct PCI connections to 8 compute blades and provide 2TB of SSD non-volatile storage per compute node. The Rabbit SSDs can be used as burst buffers or as low-latency local file systems. Each Rabbit also has its own CPU processor that can run arbitrary containerized applications such as in-transit analysis. Rabbits are also connected to the high speed network fabric.



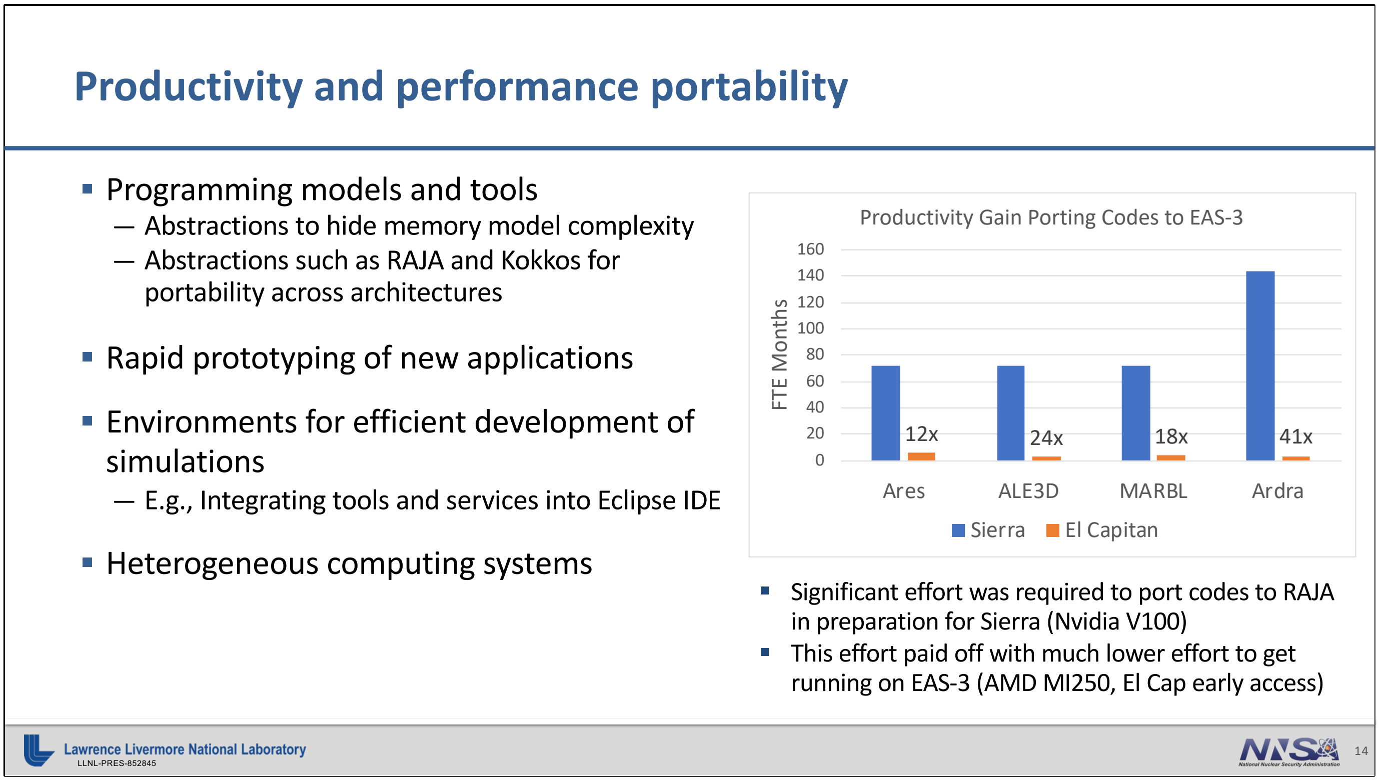
In-situ visualization with Ascent and Visit

3D Ballistics Research Lab 81a Shaped Charge on 4 nodes of RzVernal with 116 Million Quadrature Points

Lawrence Livermore National Laboratory
LLNL-PRES-852845

13

Exascale and post-exascale systems can produce data at such high rates that saving all data for later analysis is difficult or impossible. In-situ or in-transit analysis and visualization, perhaps using dedicated hardware such as rabbits, is of great interest.



Productivity and performance portability

- Programming models and tools
  - Abstractions to hide memory model complexity
  - Abstractions such as RAJA and Kokkos for portability across architectures
- Rapid prototyping of new applications
- Environments for efficient development of simulations
  - E.g., Integrating tools and services into Eclipse IDE
- Heterogeneous computing systems

- Significant effort was required to port codes to RAJA in preparation for Sierra (Nvidia V100)
- This effort paid off with much lower effort to get running on EAS-3 (AMD MI250, El Cap early access)

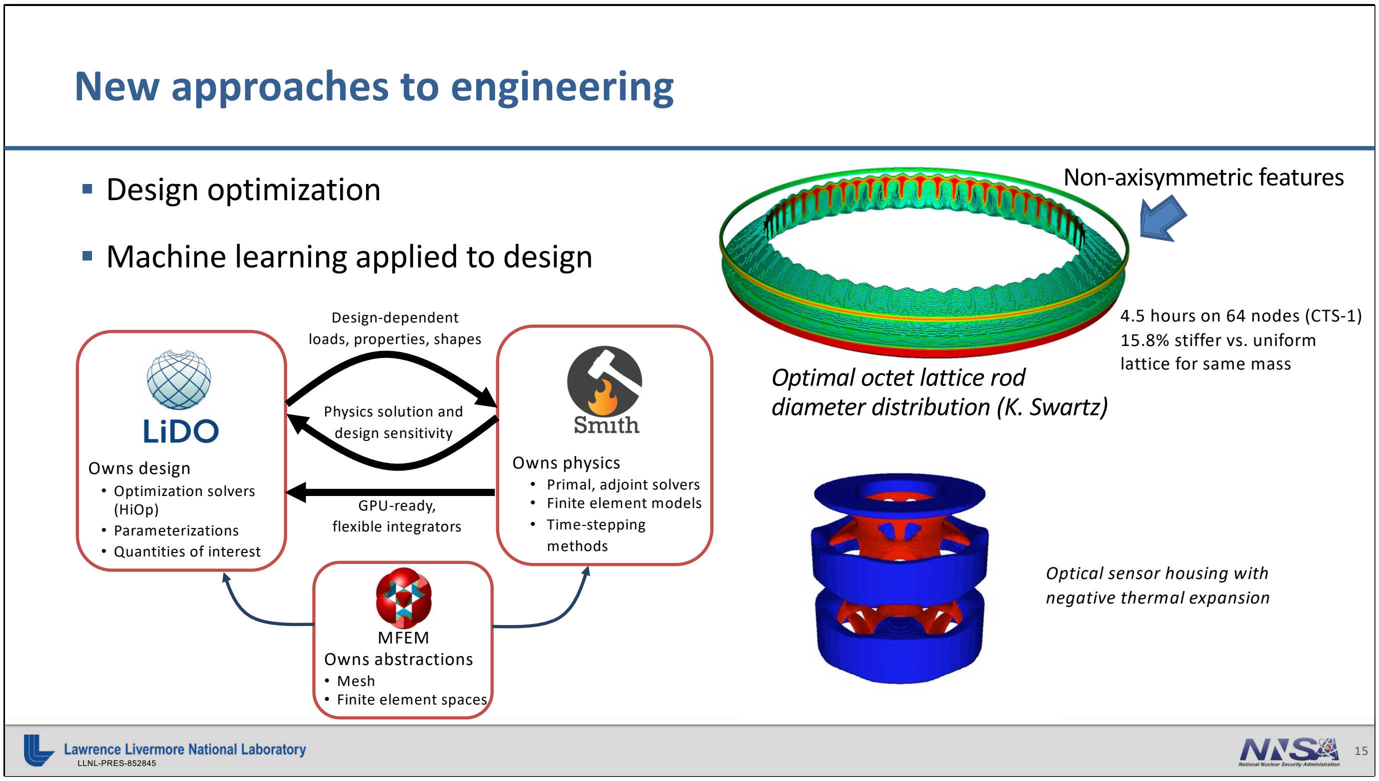
There is a strong overlap between productivity and performance portability and programing models and runtime systems so we've already talked about some of the main interests in this area.

All of our critical mission codes need productivity and performance portability on two axes. Our codes need to run on multiple current systems from laptops to supercomputers, and on chips from multiple vendors. Codes also need to be "future-proof" to run well on future architectures with minimal changes.

As we prepare for El Capitan, we're seeing the need for memory abstractions that can handle both traditional separate GPU/CPU memory spaces as well as the single memory space of the MI300 APU.

We have already discussed abstractions such as Kokkos and RAJA, which hide the complexity of heterogeneous computing systems.

And of course anything that can help design, build, test, and deliver applications into production is of interest.



New approaches to engineering

- Design optimization
- Machine learning applied to design

LiDO

Owns design
- Optimization solvers (HiOp)
- Parameterizations
- Quantities of interest

Design-dependent loads, properties, shapes

Physics solution and design sensitivity

GPU-ready, flexible integrators

Smith

Owns physics
- Primal, adjoint solvers
- Finite element models
- Time-stepping methods

MFEM

Owns abstractions
- Mesh
- Finite element spaces

Non-axisymmetric features

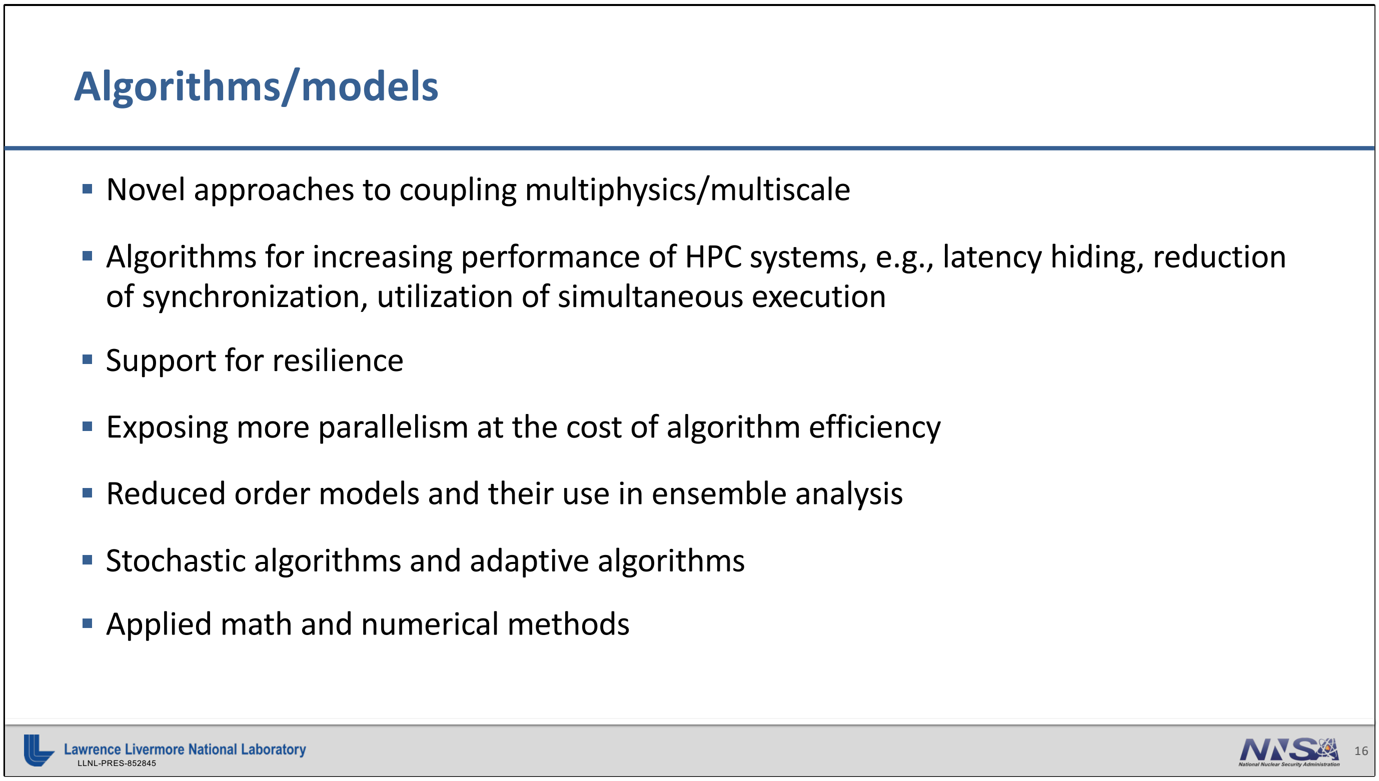
4.5 hours on 64 nodes (CTS-1)
15.8% stiffer vs. uniform lattice for same mass

Optimal octet lattice rod diameter distribution (K. Swartz)

Optical sensor housing with negative thermal expansion

New approaches to engineering is another topic of interest and includes using machine learning for topological design optimization. Some examples are shown here.

We are currently working on building modular applications for design optimization. Taking advantage of existing numerical methods and physics models accelerates development. Compared to monolithic approaches, it is also much easier to swap out or exchange physics, constraints, quantities of interest.



Algorithms/models

- Novel approaches to coupling multiphysics/multiscale
- Algorithms for increasing performance of HPC systems, e.g., latency hiding, reduction of synchronization, utilization of simultaneous execution
- Support for resilience
- Exposing more parallelism at the cost of algorithm efficiency
- Reduced order models and their use in ensemble analysis
- Stochastic algorithms and adaptive algorithms
- Applied math and numerical methods

Finally, this slide is a catch all for algorithms and models that are of interest.

Novel approaches to coupling of multi-physics at multiple scales is desired.

Algorithms for increasing performance of HPC systems is desired.

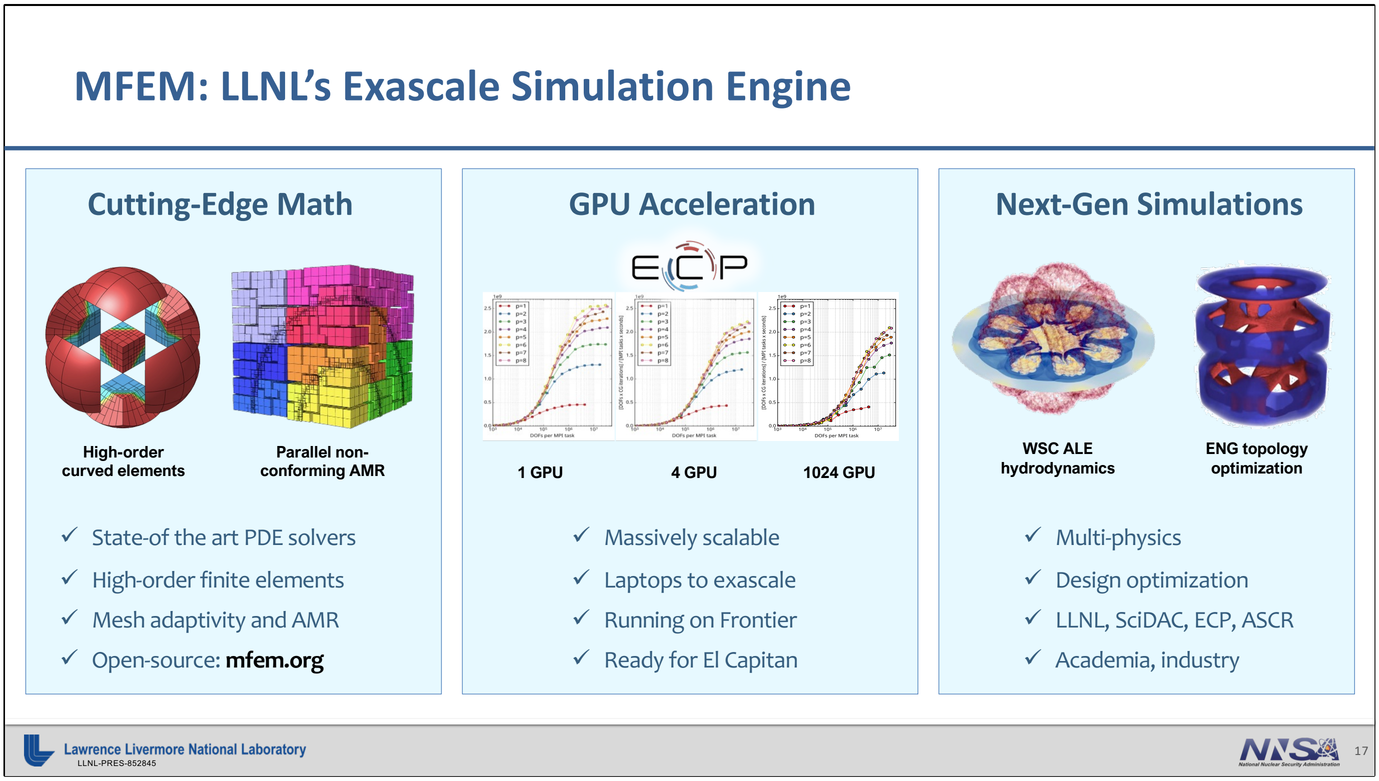
In the cases where Exascale HPC becomes less reliable, resilience and fail-over become important.

As processors and accelerators continue to advance there is a need to expose even more parallelism, and efficiency is more appropriately measured in wall time as compared to minimizing the number of operations

Reduced order models and their use in ensemble analysis is needed.

Stochastic algorithms such as stochastic optimization is possible because of Exascale.

And finally, we should not forget about applied mathematics research and numerical methods that are specifically adapted to GPUs or other exascale enabling architectures.



MFEM: LLNL's Exascale Simulation Engine

Cutting-Edge Math

High-order curved elements

Parallel non-conforming AMR

- ✓ State-of the art PDE solvers
- ✓ High-order finite elements
- ✓ Mesh adaptivity and AMR
- ✓ Open-source: **mfem.org**

GPU Acceleration

ECP

1 GPU

4 GPU

1024 GPU

- ✓ Massively scalable
- ✓ Laptops to exascale
- ✓ Running on Frontier
- ✓ Ready for El Capitan

Next-Gen Simulations

WSC ALE hydrodynamics

ENG topology optimization

- ✓ Multi-physics
- ✓ Design optimization
- ✓ LLNL, SciDAC, ECP, ASCR
- ✓ Academia, industry

Lawrence Livermore National Laboratory
LLNL-PRES-852845

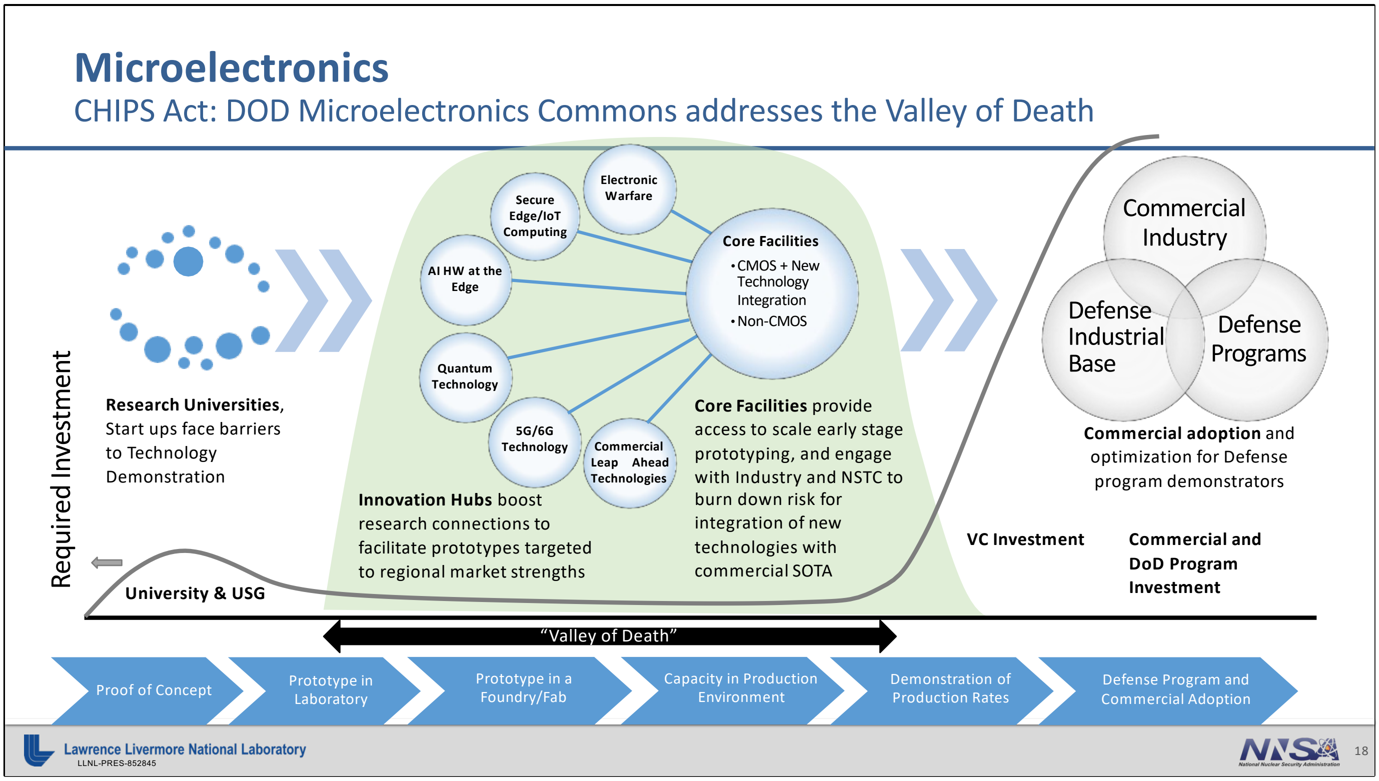
NNSA
National Nuclear Security Administration

17

The MFEM library is just one example of NNSA efforts to develop numerical methods that are specifically tailored to GPUs. High-order finite elements and matrix-free methods help optimize time to solution on GPU architectures.

Microelectronics

CHIPS Act: DOD Microelectronics Commons addresses the Valley of Death



Research Universities, Start ups face barriers to Technology Demonstration

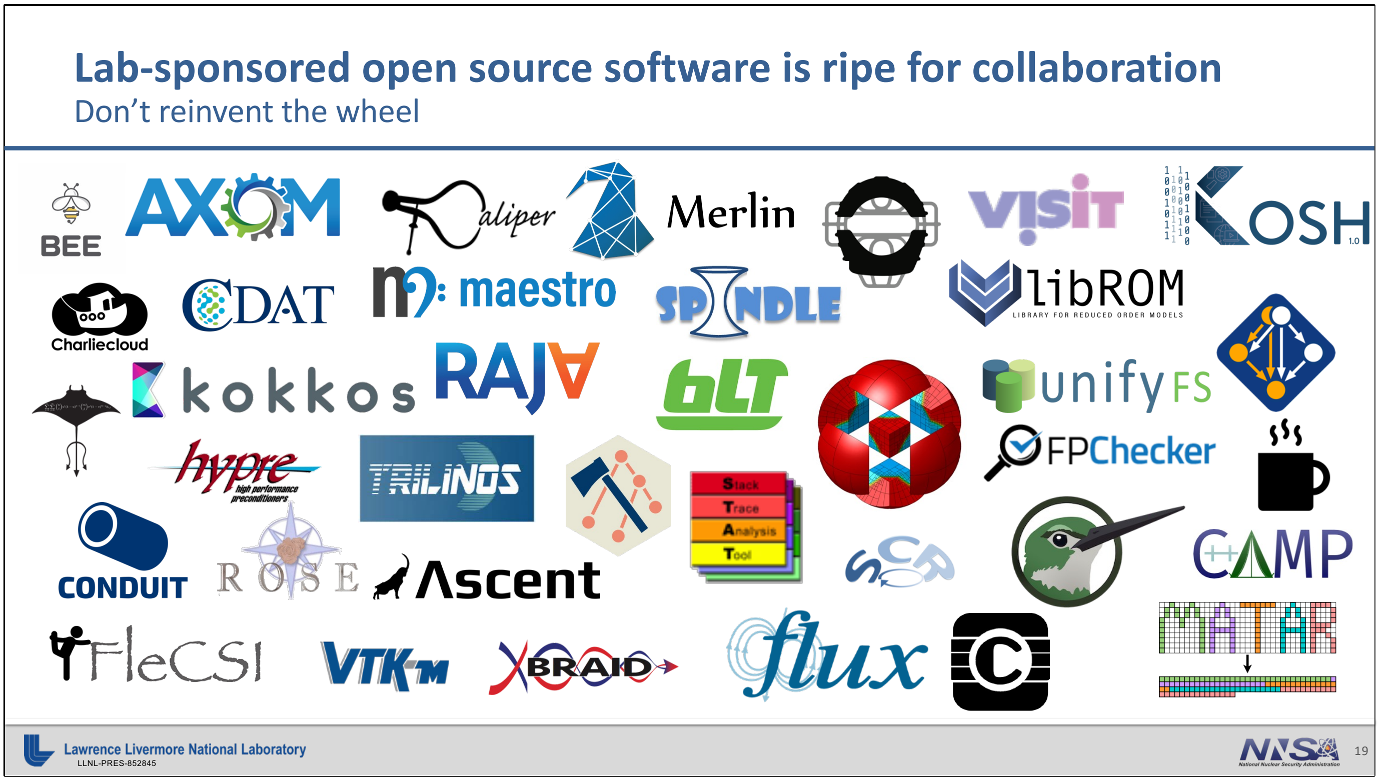
Innovation Hubs boost research connections to facilitate prototypes targeted to regional market strengths

Core Facilities provide access to scale early stage prototyping, and engage with Industry and NSTC to burn down risk for integration of new technologies with commercial SOTA

Commercial adoption and optimization for Defense program demonstrators

Commercial and DoD Program Investment

We anticipate that the CHIPS act will support university research in microelectronics. Be aware of potential synergies or opportunities to leverage such research in PSAAP activities.



Lab-sponsored open source software is ripe for collaboration

Don’t reinvent the wheel

BEE
AXOM
Caliper
Merlin
VisIt
KOSH 1.0
Charliecloud
CDAT
maestro
SPINDLE
libROM
LIBRARY FOR REDUCED ORDER MODELS
kokkos
RAJA
bLT
unifyFS
FPChecker
hypre
high performance preconditioners
TRILINOS
Stack
Trace
Analysis
Tool
CONDUIT
ROSE
Ascent
SCR
CAMP
MATAR
FleCSI
VTK-m
XBRAID
flux

Please do not overlook opportunities to use and extend the many open-source software resources developed at the NNSA labs. There are many opportunities to collaborate with developers at the labs.

Thanks to those who helped provide material for these slides

- David Beckingsale
- Jamie Bramwell
- Bronis de Supinski
- Erik Draeger
- Charles Doutriaux
- John Feddema
- Cyrus Harrison
- Rob Hoekstra
- Judy Hill
- Dan Laney
- Katie Lewis
- Tzanio Kolev
- Anna Pietarila Graham
- Tom Scogland
- Galen Shipman
- Tom Stitt

Lawrence Livermore
National Laboratory