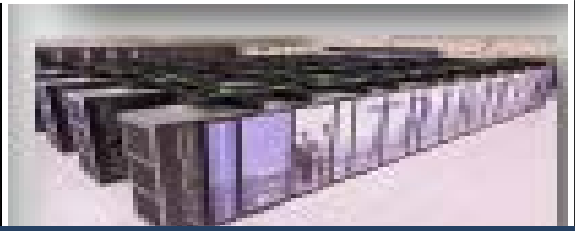
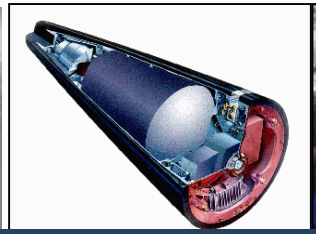
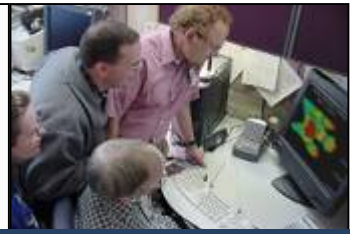
Computing Challenges for the Third Decade of Stockpile Stewardship

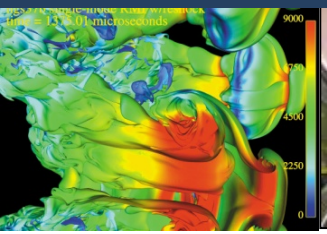
Mark Anderson, Director
Office of Advanced Simulation and Computing and Institutional Research and Development

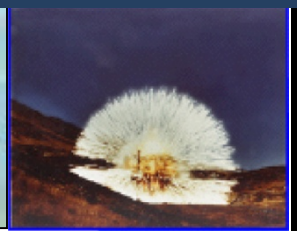
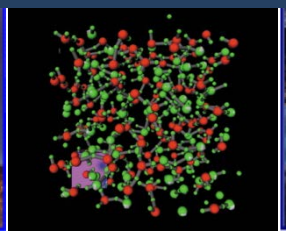
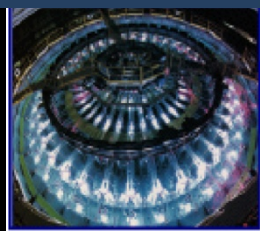
ASC is a key element of the nuclear security enterprise



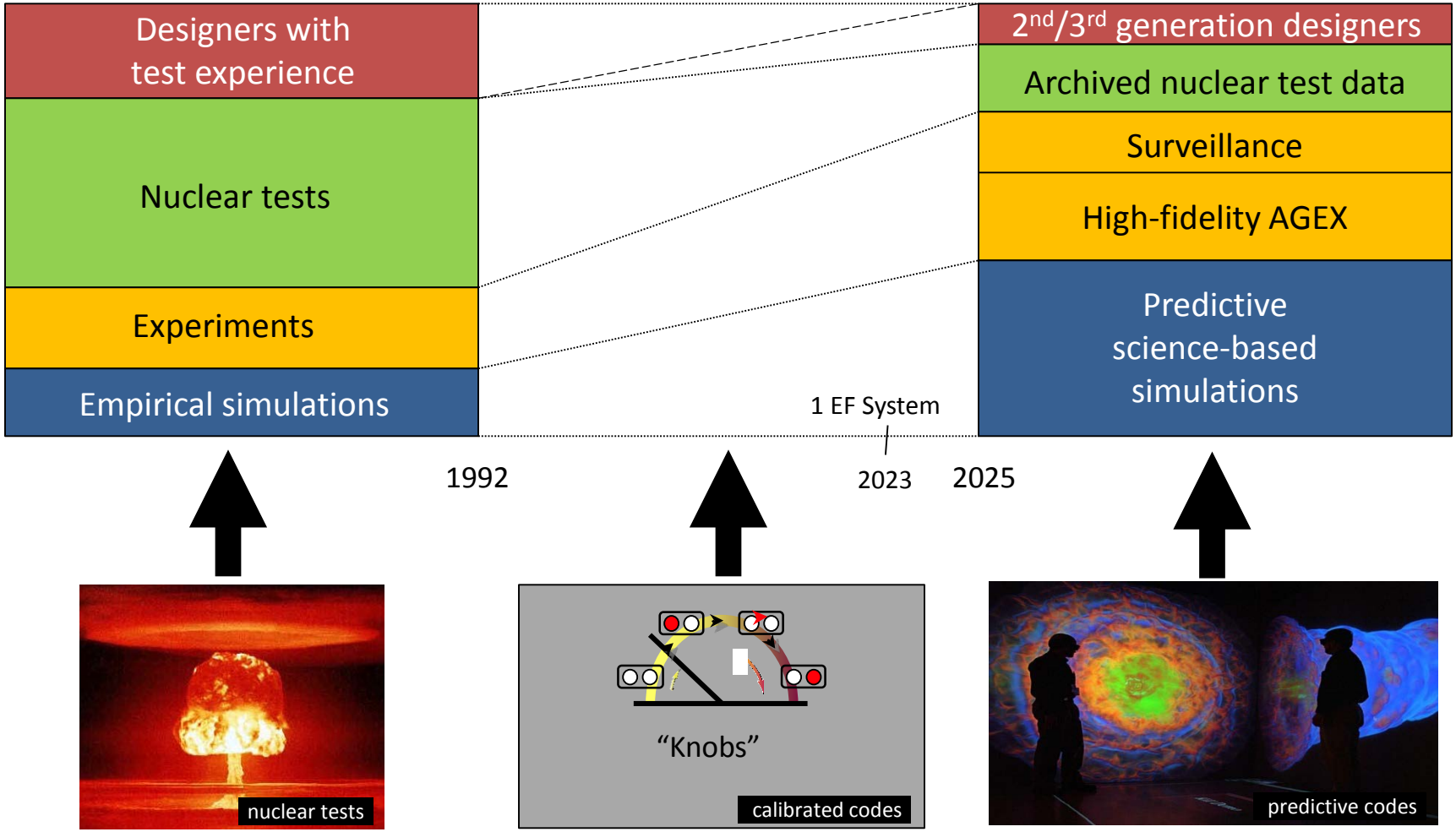




- Defense Program's Mission: maintain a safe, secure, and effective nuclear deterrent
- ASC is the lens through which we apply our science and nuclear test history to assess, qualify, and certify the stockpile without testing
- We also support a variety of threat reduction activities



Confidence in stockpile assessment rests on our people and tools



Our confidence, once based on nuclear tests, now must be founded on improved physics understanding incorporated into predictive simulations

Nuclear security threats are changing

Global: Russia, China

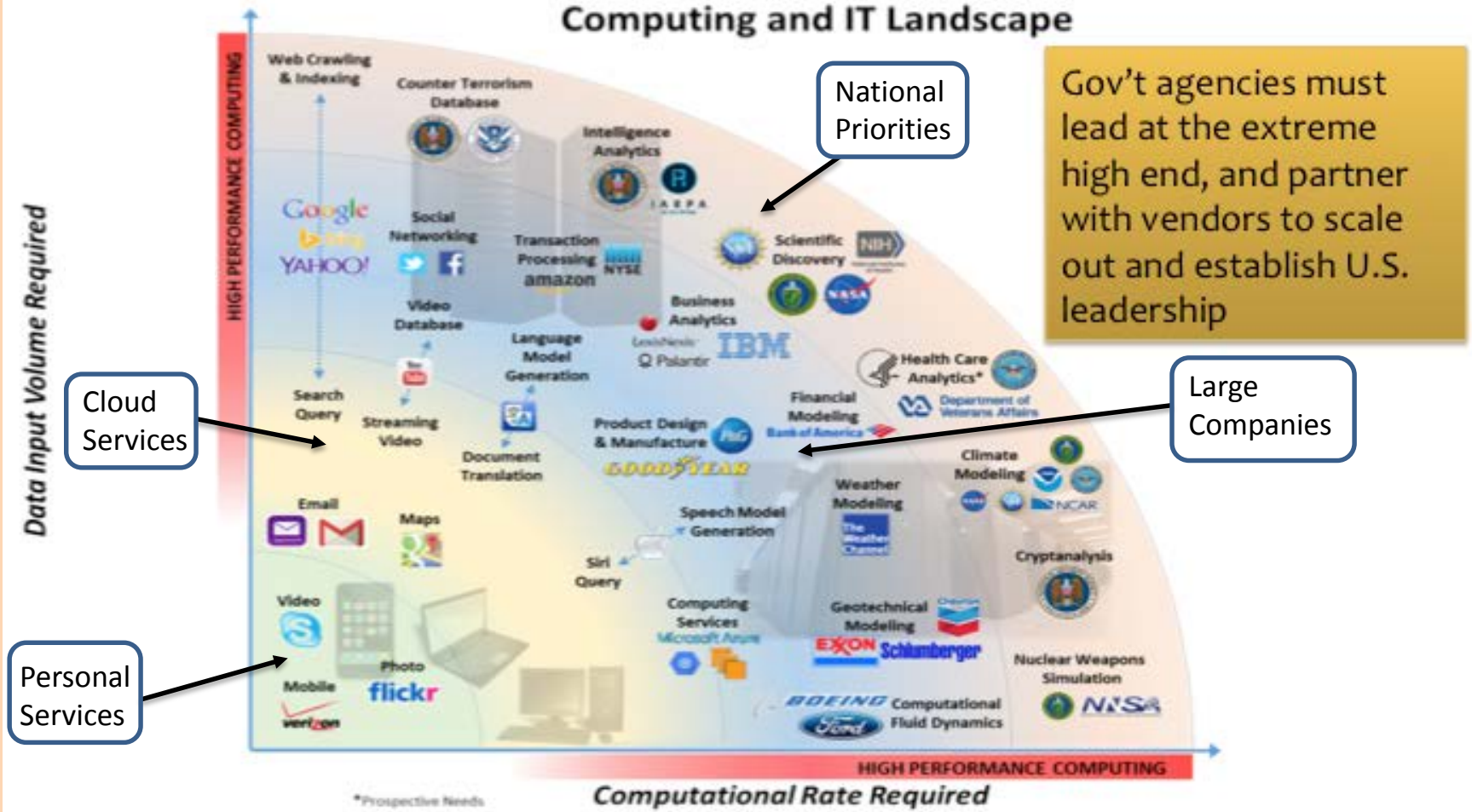
2018 Nuclear Posture Review directs NNSA to deliver

- Low-yield sub-launched ballistic missile
- Sea-launched cruise missile

ASC provides tools to design and certify

Regional: North Korea, Iran

Computing technology is also evolving rapidly, driven by market forces broader than just HPC



Our idealized technology maturation model is modified by market forces

Testbeds	Prototypes	Advanced Technology Systems	Commodity Technology Systems
			
Small scale (node-level) R&D. Understand the potential of new technology	Built from emergent technologies. Prove rack-level production-capable systems useful for mission applications	Leading-edge technology. Includes non-recurring engineering and early access systems	Scalable production technology. Run mission simulation workloads

These are usually proposed by the vendors at the largest scale

Sierra, our most recent ATS system, is being deployed at Livermore

Compute Node

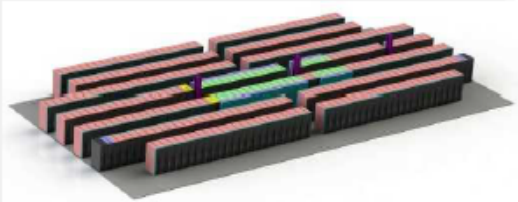
2 IBM POWER9 CPUs
4 NVIDIA Volta GPUs
NVMe-compatible PCIe 1.6 TB SSD
256 GiB DDR4
16 GiB Globally addressable HBM2 associated with each GPU
Coherent Shared Memory

Compute Rack

Standard 19”
Warm water cooling

Compute System

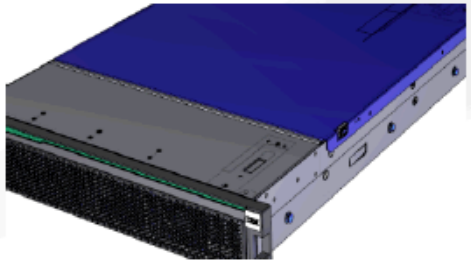
4320 nodes
1.29 PB Memory
240 Compute Racks
125 PFLOPS
~12 MW



Components

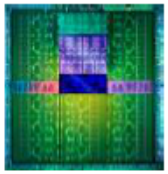
IBM POWER9

- Gen2 NVLink



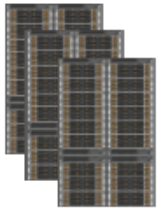
NVIDIA Volta

- 7 TFlop/s
- HBM2
- Gen2 NVLink



Mellanox Interconnect

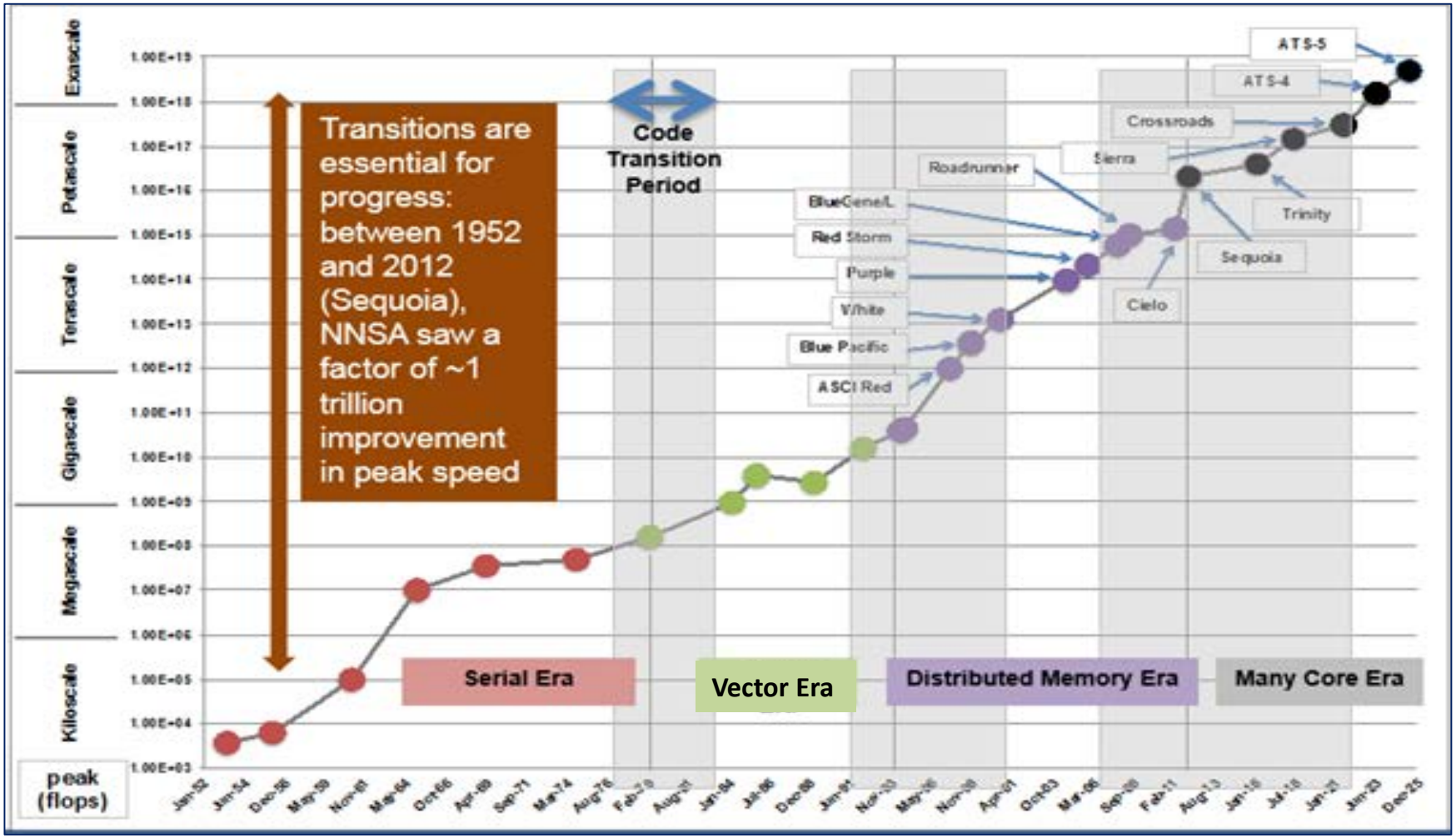
Single Plane EDR InfiniBand
2 to 1 Tapered Fat Tree



GPFS File System

154 PB usable storage
1.54 TB/s R/W bandwidth

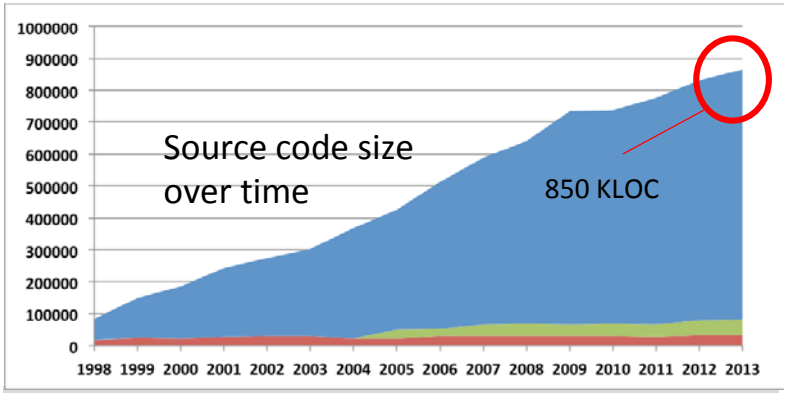
Coping with new hardware technologies requires new programming approaches



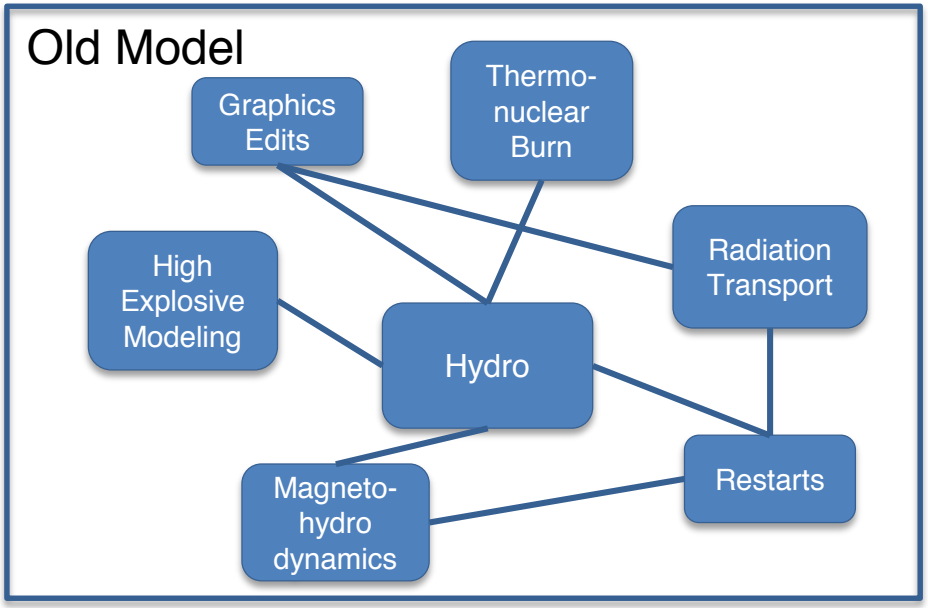
Each of these eras defines a new common programming model and transitions are taking longer; NNSA is entering a fourth era.

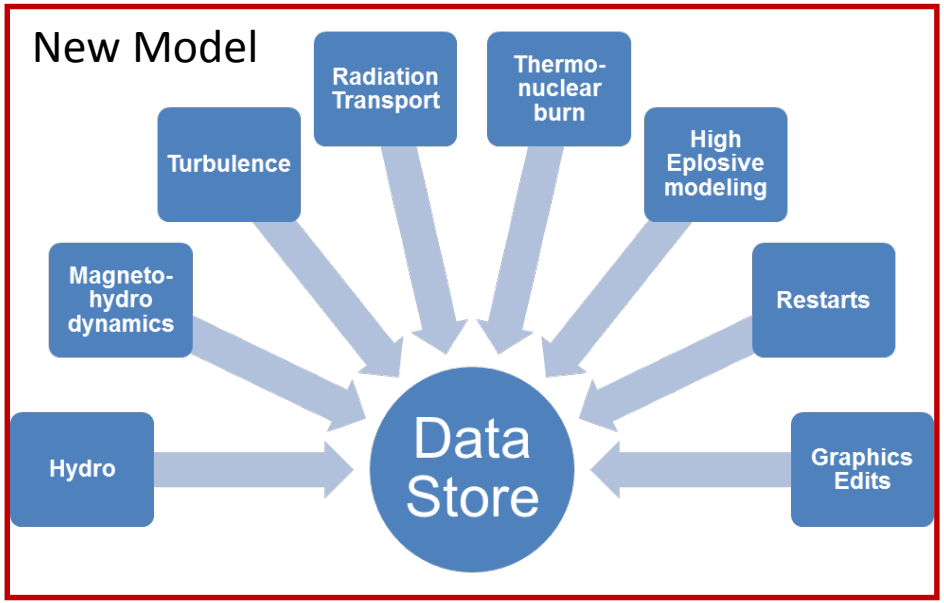
Increasing code complexity, coupled with architecture uncertainty, is driving our labs to a new, modular approach

- Codes today are >5x larger than pre-ASCI codes
- Modular infrastructure can be reused by other codes
- Central Data Store allows individual components independently



Average of 60 kilo lines of code (KLOC) per year of additional code to support a representative integrated code





We are transforming our existing codes and developing new ones to capitalize on the hardware evolution

Fundamental limitations of CMOS (chip) technology are driving changes in how high-performance software is written

Recent past applications/software designed around:

- Floating point (*numerical*) bottlenecks
- Coarse-grained parallelism
- Stable/predictable architecture designs

Code Transformation Process

Theoretical prediction of performance limiters (*proactive*)

We are here

Empirically based studies on emerging hardware (*responsive*)

Multiple years of effort
Millions of lines of code

Future applications must design for:

- Memory and network (*data motion*) bottlenecks
- Fine-grained concurrency
- Multiple diverse architecture designs

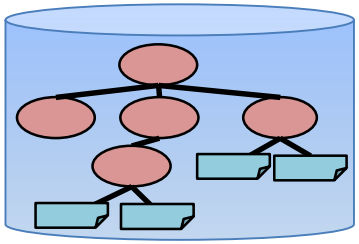
Optimized software performance lags the potential offered by new hardware while architectural impacts are understood and code is transformed

NNSA aims to minimizes that lag through:

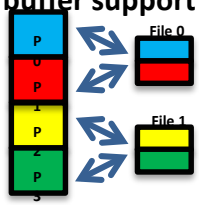
a) Co-design and Centers of Excellence (partnering with vendors)
b) Research (partnering with ASCR)
c) Early access hardware (testbeds)
d) Market incentives (large procurements)
e) Mission incentives (pressure from the users to transform)

AXOM & RAJA – Applications sharing CS software infrastructure (Livermore)

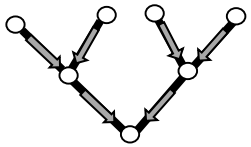
Hierarchical key-value data management



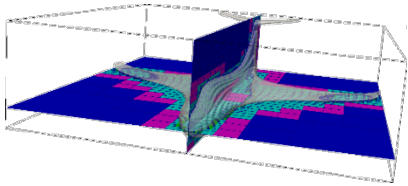
Parallel file I/O & burst buffer support



Unified cross-package message logging & parallel processing

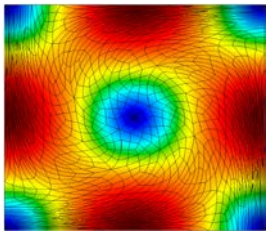


Surface queries & spatial acceleration data structures



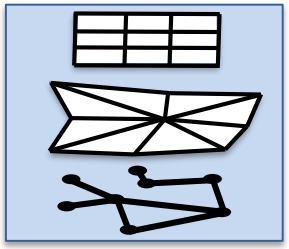
AXOM: Common CS Infrastructure

Mesh-aware data schema



```
"coordsets": {
  "coords": {
    "type": "explicit"
    "values": {
      "x": [double],
      "y": [double]
    }
  }
},
"topologies": {
  ...
...
```

Mesh data model



Application Code

RAJA

Serial | OMP3 | CUDA

OMP4.5 | ROCm | ??

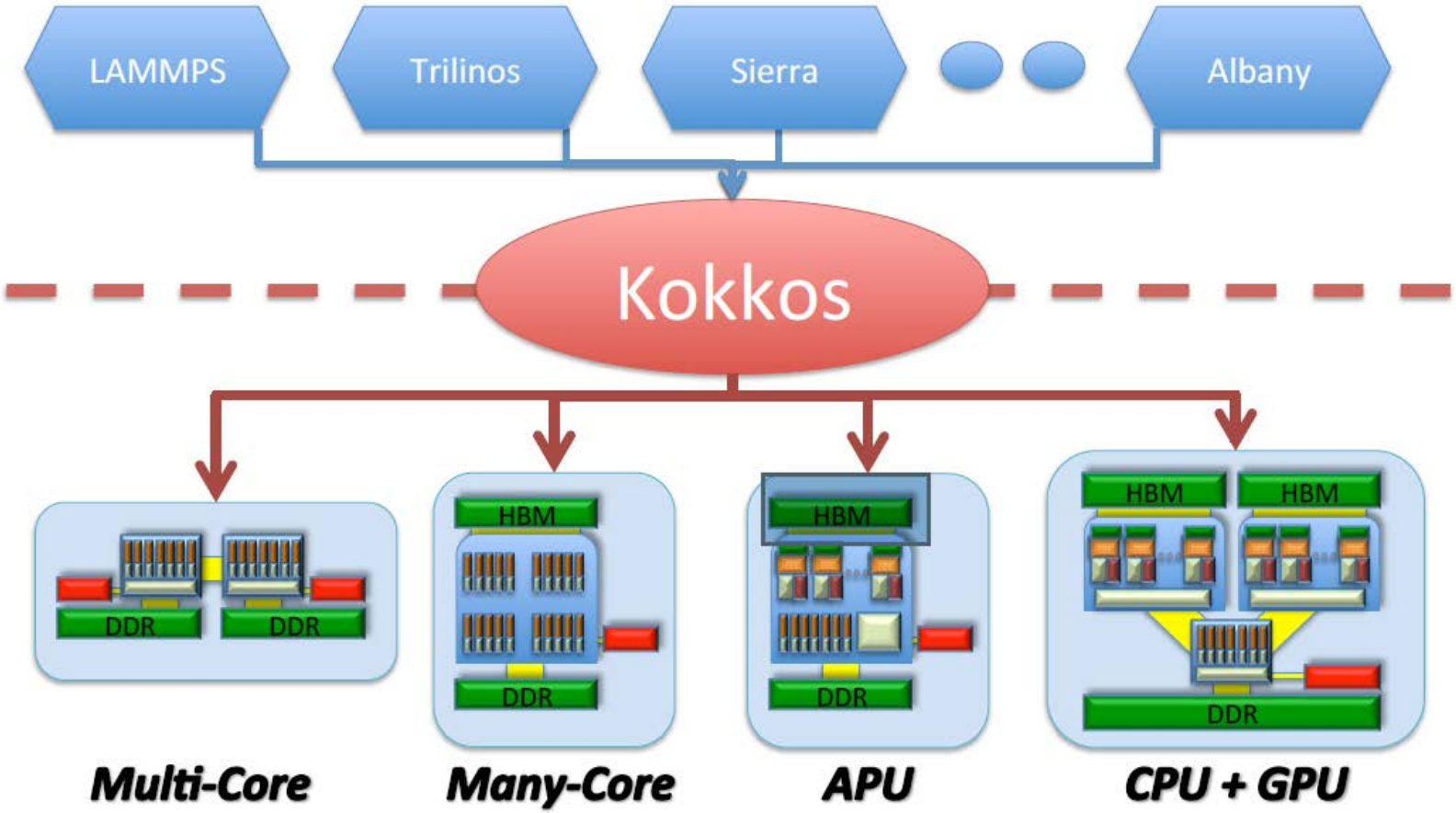
Mature Programming Models

Emerging Programming Models

RAJA allows ease of switching between backends enabling the best tools for our codes and also helping steer vendor interactions.

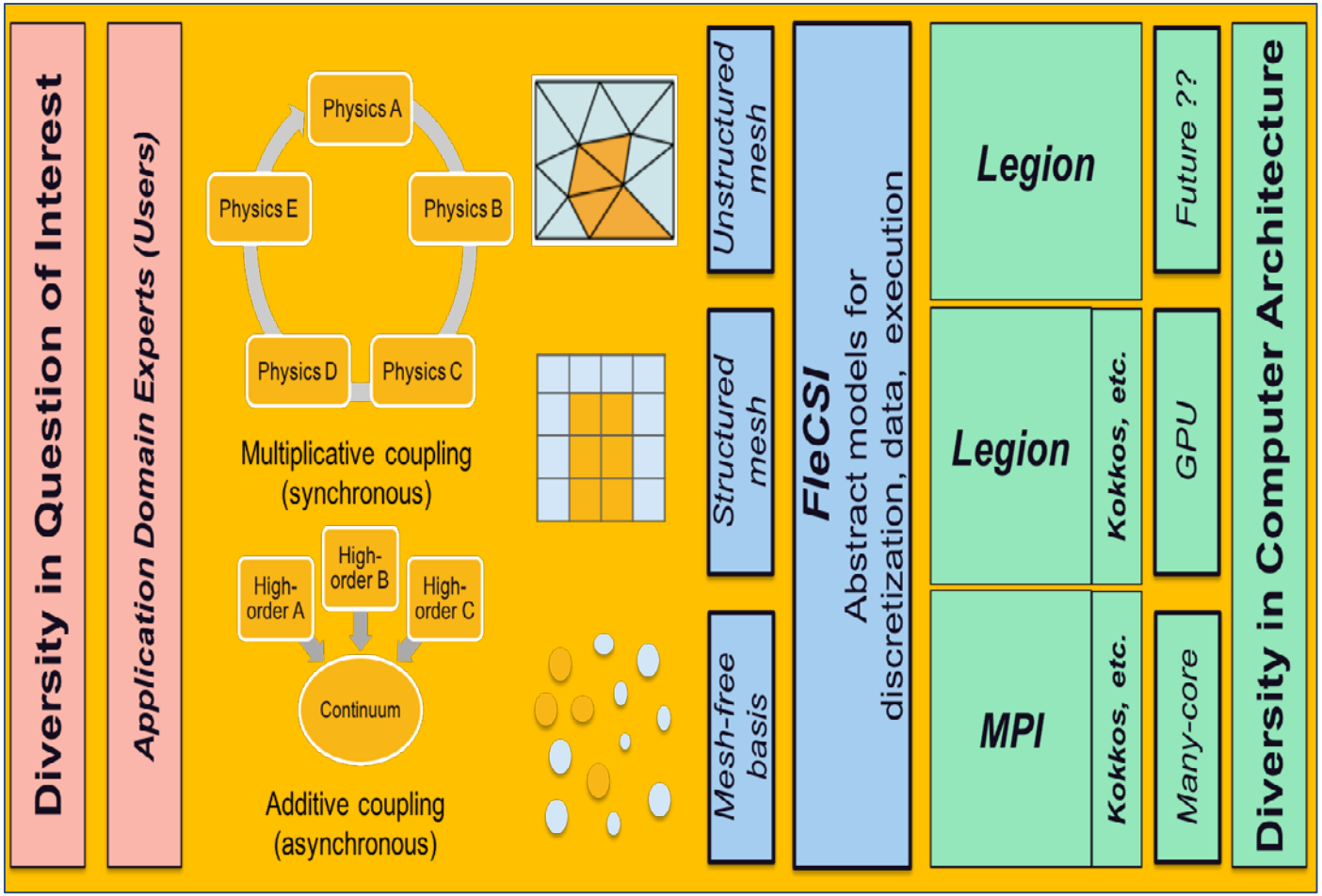
RAJA Performance Portability Library

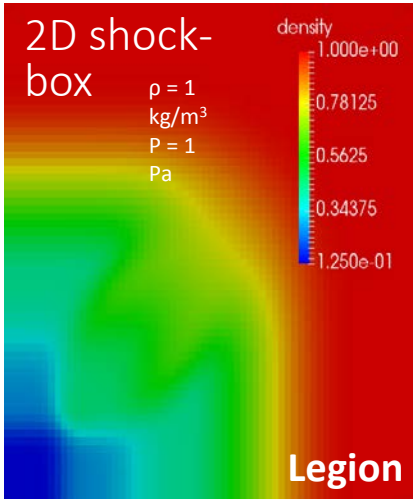
Kokkos – Performance, portability and productivity (Sandia)

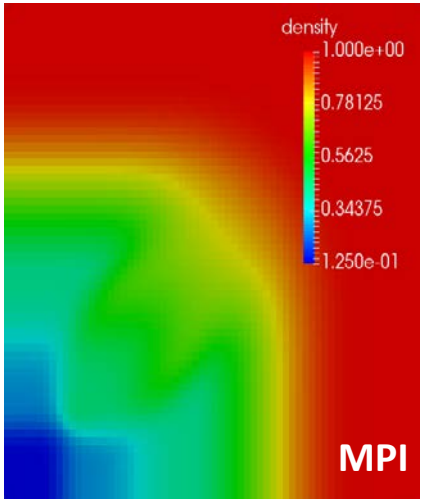


https://github.com/kokkos

Ristra – A software architecture for multi-physics (Los Alamos)

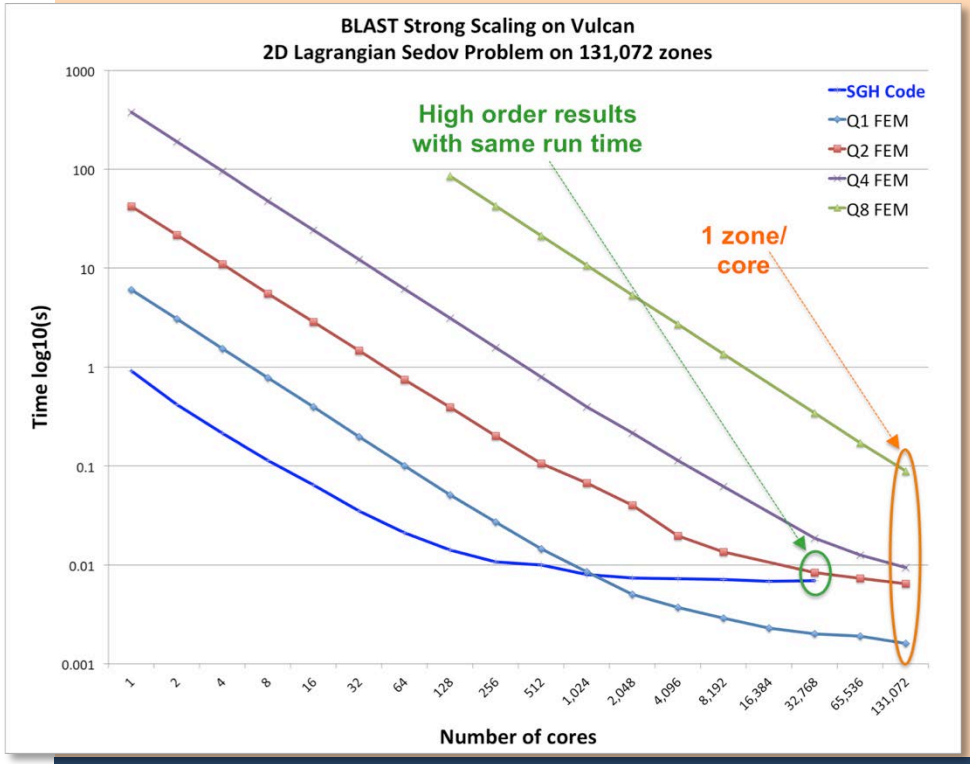


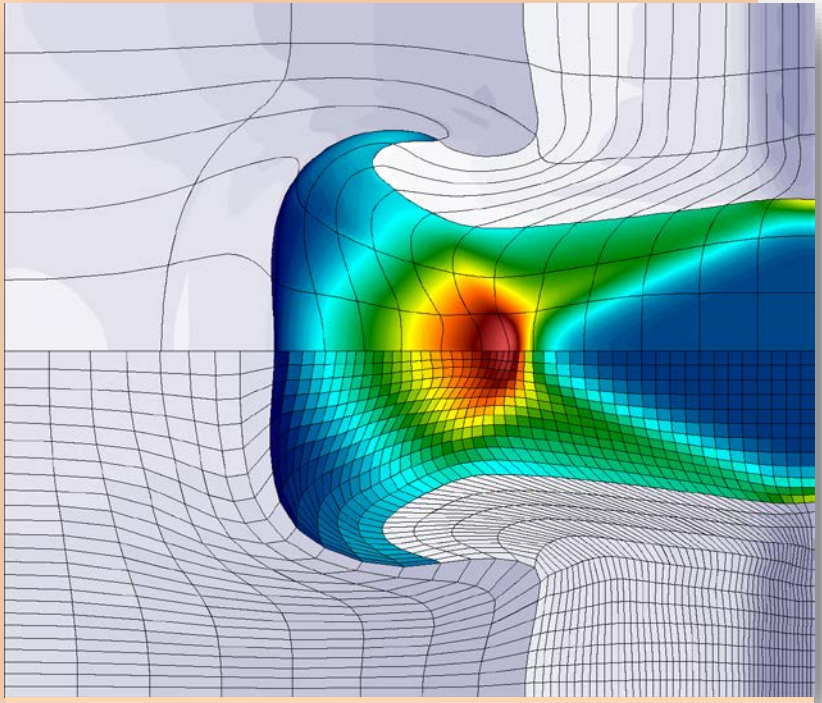




In addition to programming model changes, the labs now need to rethink basic algorithms

Algorithms that reduce the number of variables by increasing the computational work per variable will scale better on advanced architectures



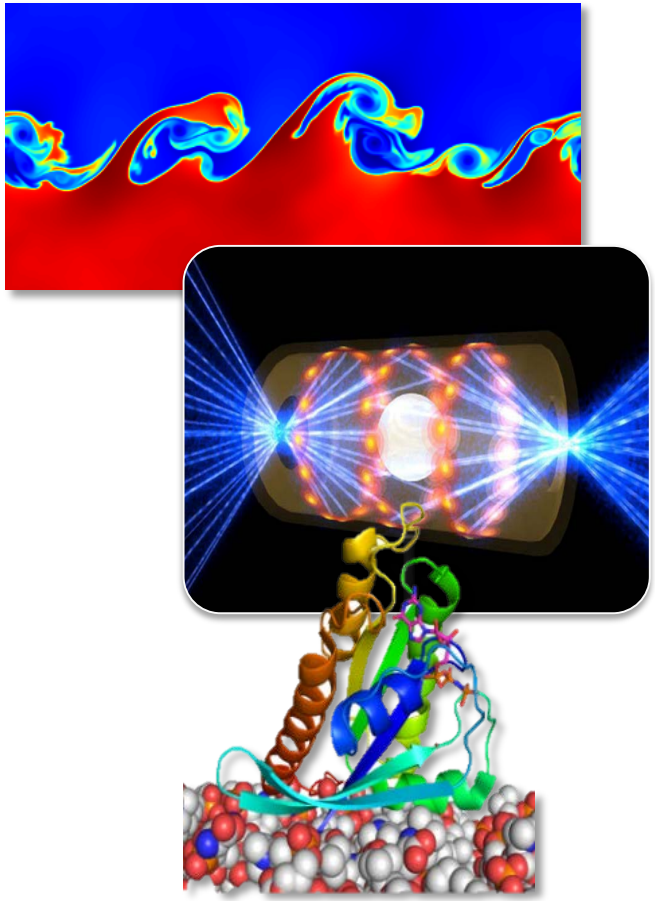


Excellent scalability is possible because of the high compute to memory fetch ratio

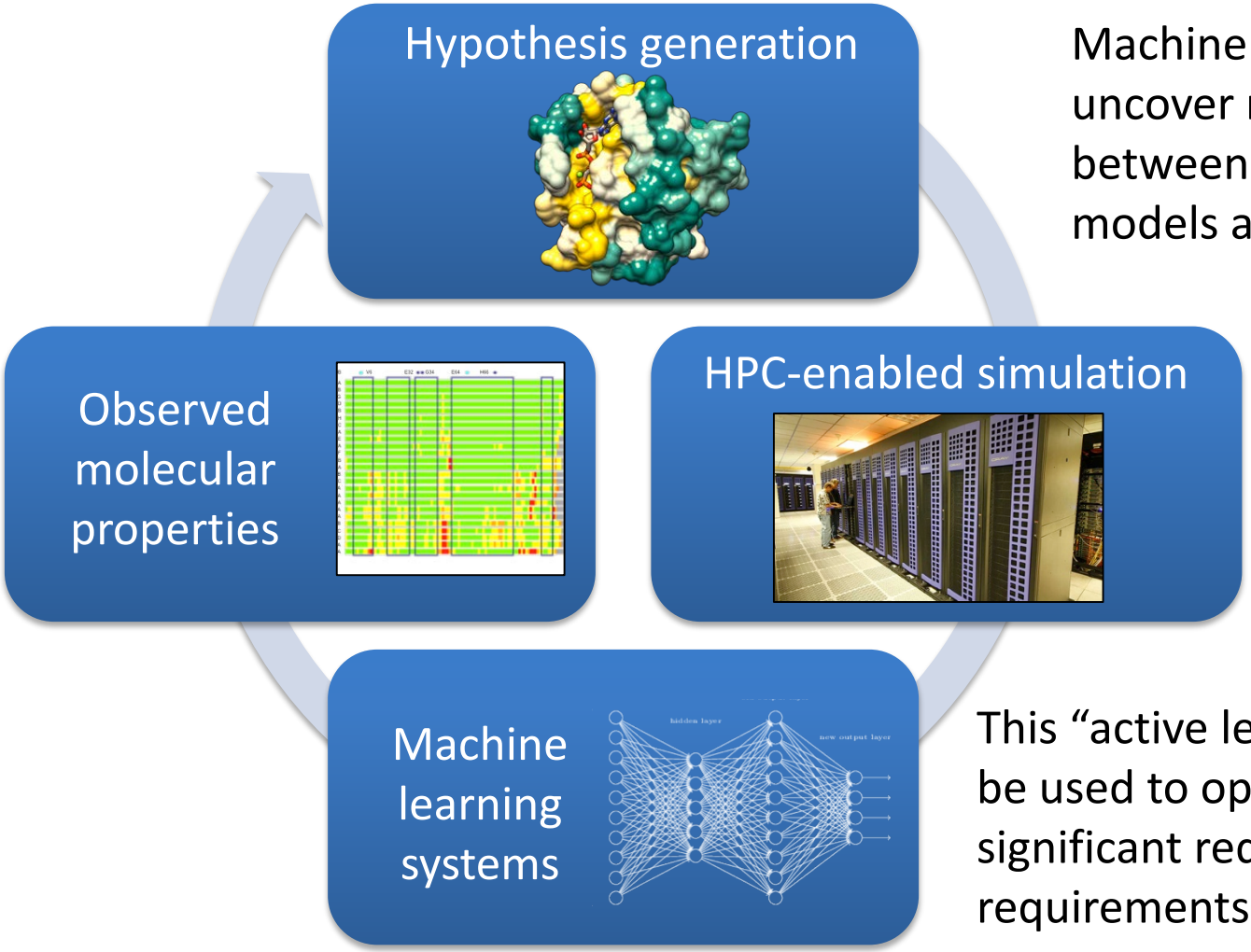
Just as in the past, these complexities point to why the NNSA labs need to engage with the vendors (co-design) to assure NNSA codes are efficient.

Brain-inspired computing requires new approaches drawing on both simulation and data science

- Conventional machine learning (train models, optimize simulation process, etc.)
- Improving simulation workflow and optimization
- Recasting problems and algorithms to exploit new architectures designed for ML



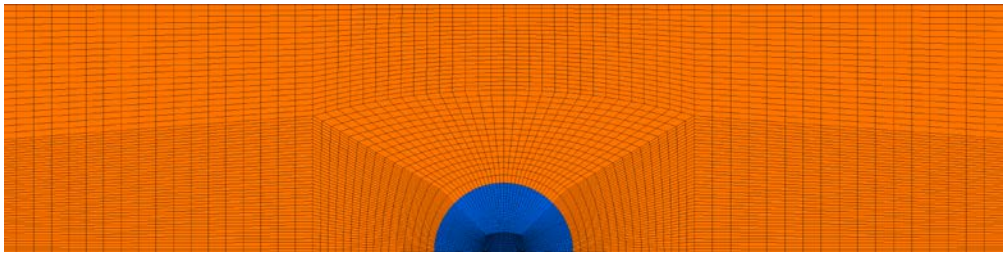
Machine learning through the simulation cycle optimizes the simulation process



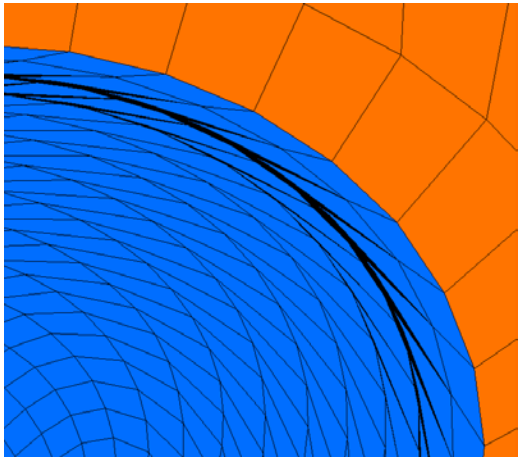
Machine learning systems uncover relationships between computational models and experimental data

This “active learning” approach can be used to optimize solutions with significant reduction in compute requirements

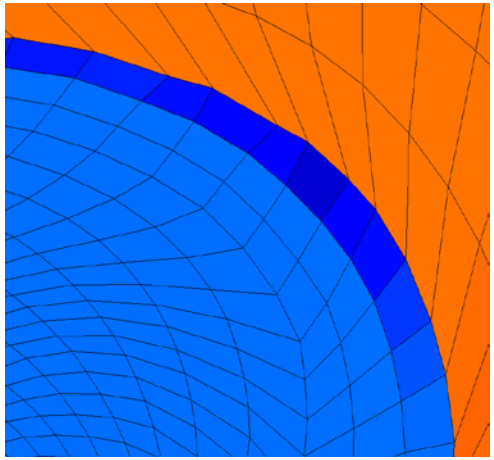
Future large-scale simulations will be analyzed and optimized by integrated machine learning



Bubble shock simulation: initial cycle



Tangled mesh



Machine learning relaxed mesh

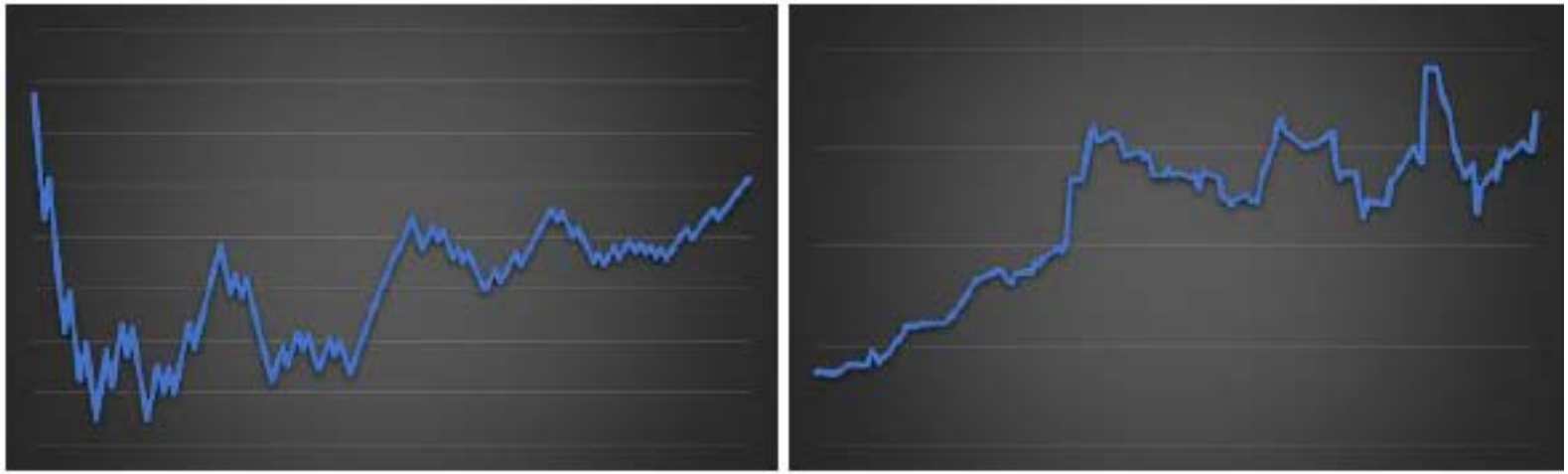
Large state-of-the-art simulations are becoming too complex for humans to manage and control

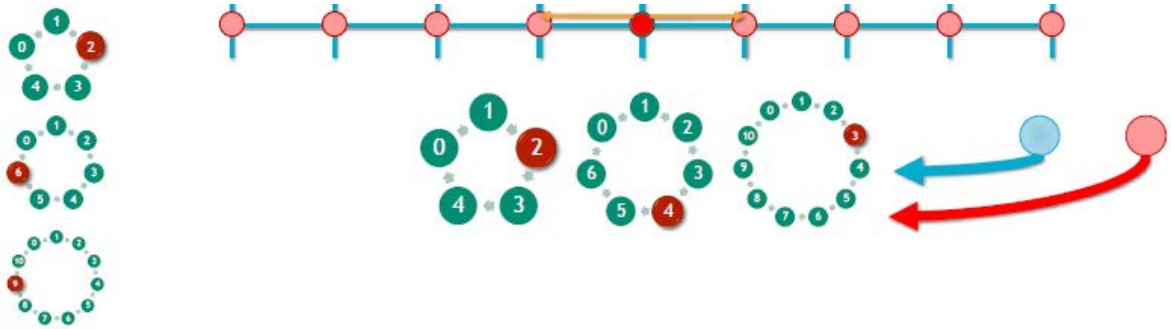
- Failures in workflow e.g. mesh tangling
- Outputs too large for current analysis and visualization methods
- Complex parameter spaces requiring large ensembles of runs

Early results show the promise of learning systems for simulation workflow management

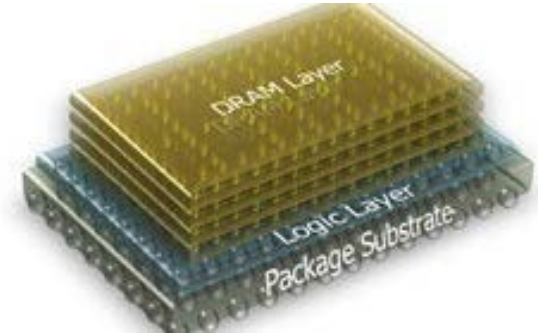
Recasting algorithms will let us use ML-centric architectures

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2}$$

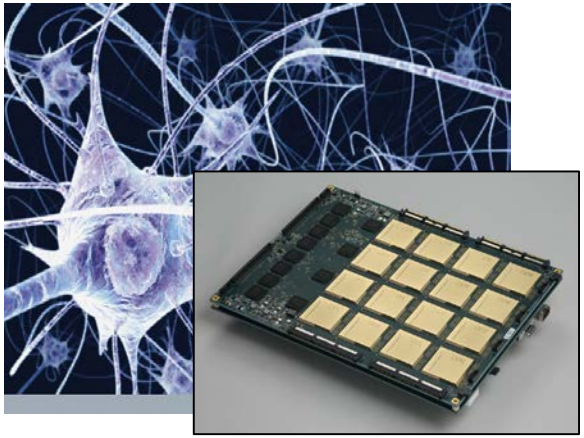




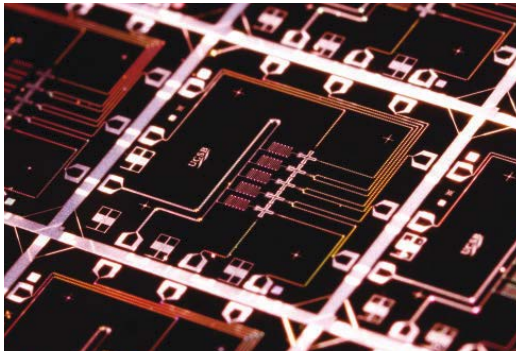
Beyond Von Neumann architectures and Moore's Law to the next-gen computing technologies



Newer memory architectures



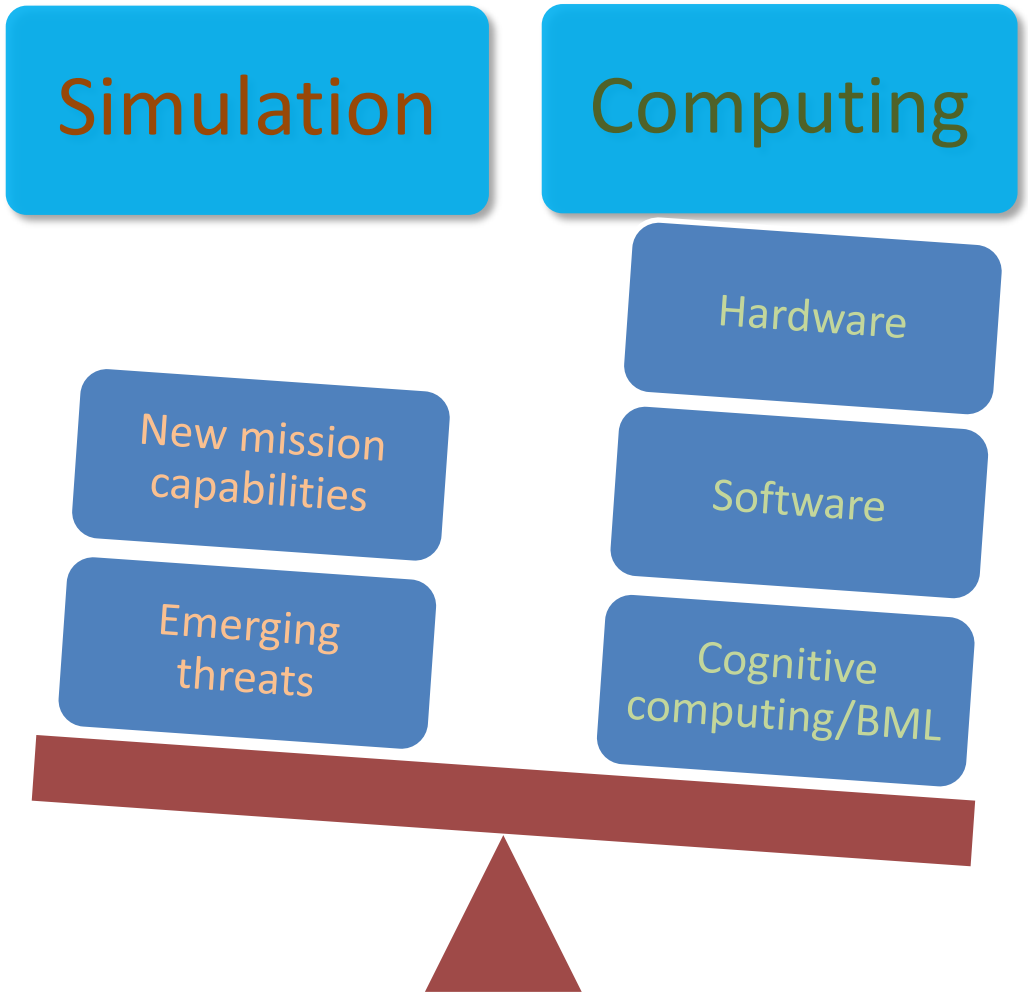
Neuromorphic systems



Quantum information systems

The computing landscape will continue to evolve; we must capitalize on future technologies

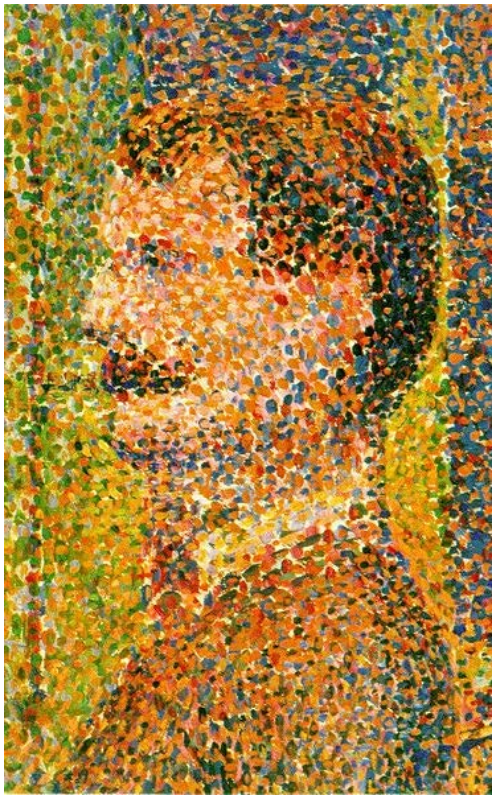
Balancing the simulation & computing requirements is a challenge



We also must realize that simulation and computing are tools to achieve solutions, not solutions themselves

$$\frac{BEM}{BAC} \notin \mathbb{R}$$

$$\frac{BEM}{BAC} \sim \mathbb{R}$$



Detail from *The Parade*, Georges Seurat, 1889

Assessing the costs and benefits of NNSA's investments in HPC must be mission-focused

We must remain a viable alternative to nuclear testing

How do we measure ROI for investments in HPC?

HPC systems are only a small investment in the total ASC program's portfolio

Will the new technologies stemming from Exascale research, Cognitive Computing and BML give us our ROI?

How do we address these questions?