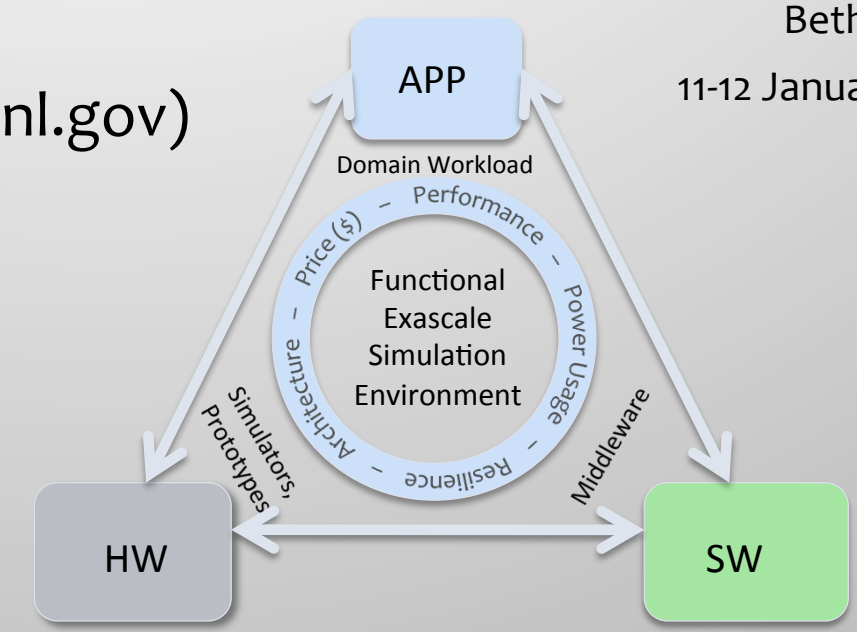
MGI (Nature / Nurture) and Computational Materials Science: Lessons from the DOE/ASCR Exascale Materials Co-design Center

James Belak (belak@llnl.gov)
www.exmatex.org

co-PI: Tim Germann (tcg@lanl.gov)

2016 MGI PI Meeting
Bethesda, MD
11-12 January, 02016



'There is not pure science and applied science but only science and the applications of science'
- Louis Pasteur

Ref's
Pasteur's Quadrant, Stokes, 1997
The Idea Factory, Gertner, 2013

LLNL-PRES-681050

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and supported by Office of Science Advanced Scientific Computing Research Program.

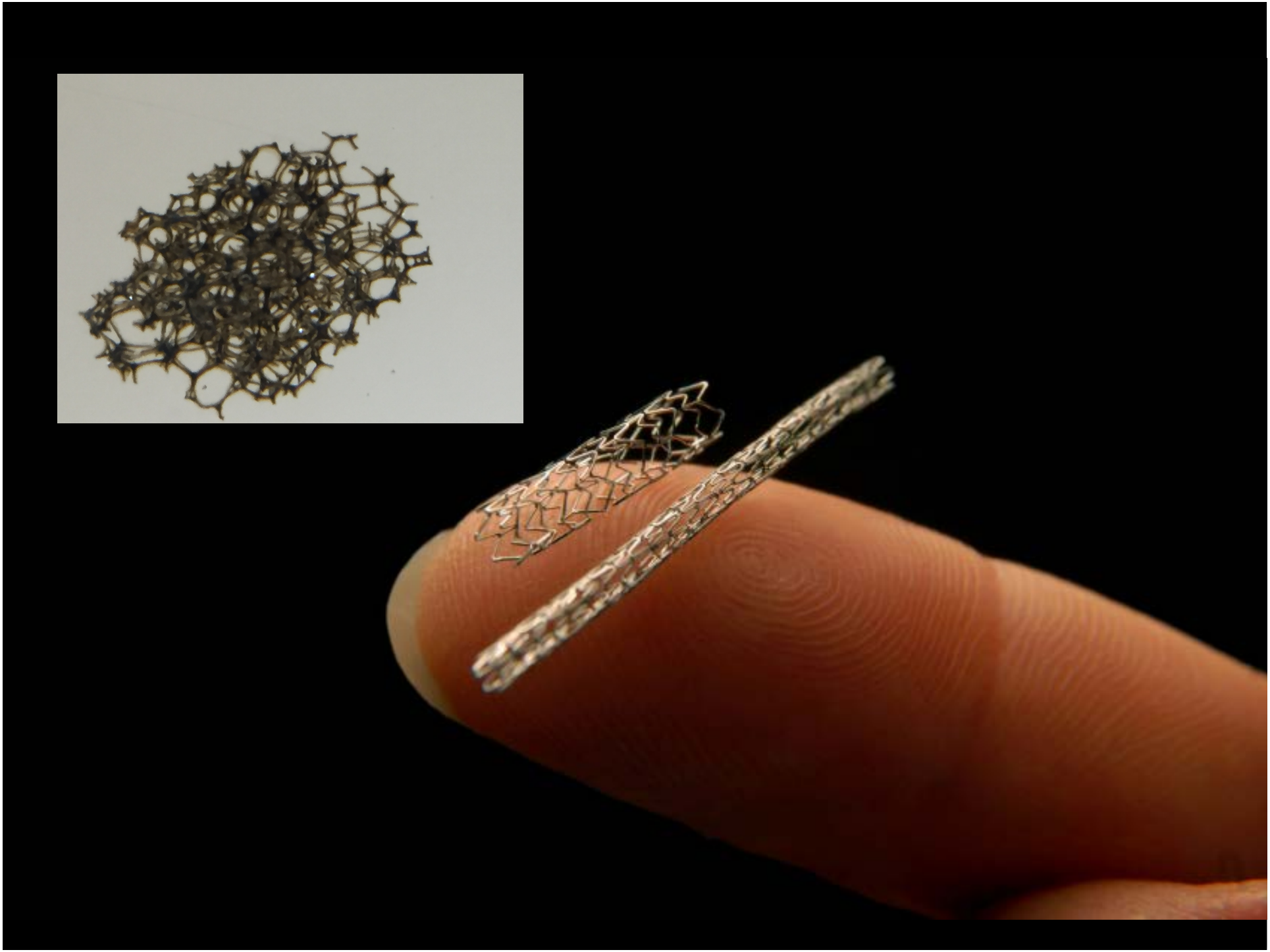
The Fourth Paradigm

Data-Intensive Scientific Discovery

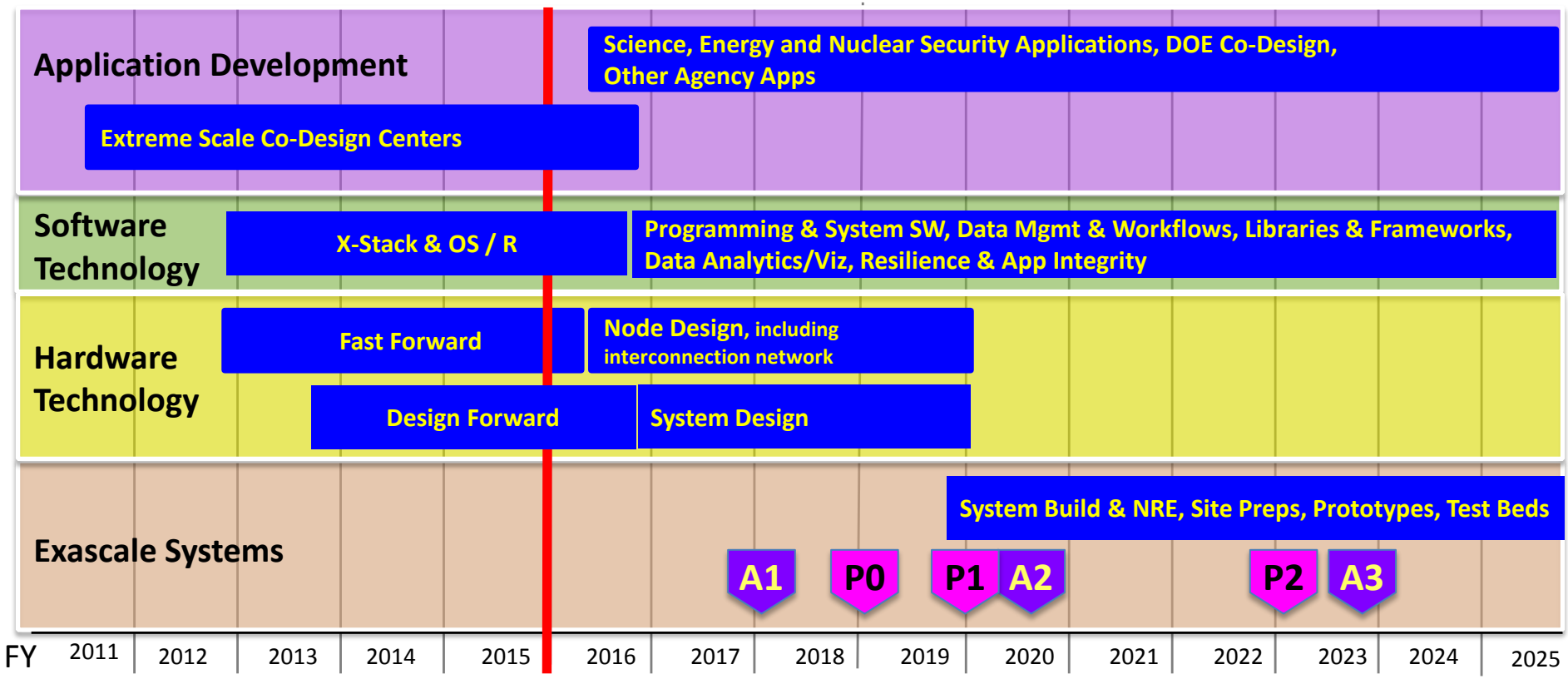
Dedicated to Jim Gray

EDITED BY TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE

http://research.microsoft.com/en-us/collaboration/fourthparadigm/



Steve Binkley, Deputy Director, DOE Office of Advanced Scientific Computing Research



P0 Node Prototype

P1 Petascale Prototype

P2 Exascale Prototype

A1 CORAL Systems

A2 APEX Systems

A3 Exascale Systems

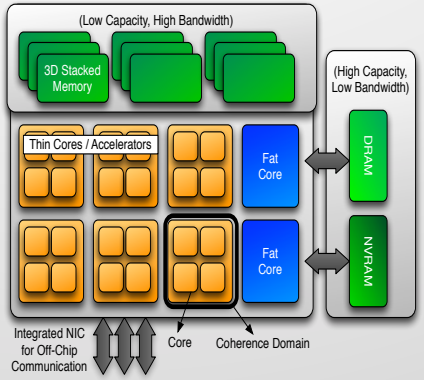
Procurement of A1, A2 and A3 are not executed, but are highly leverage, by the ECI Project

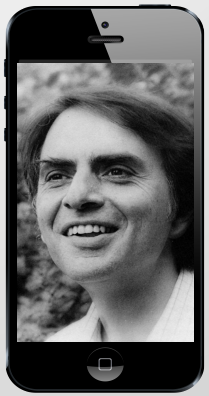
Common asked questions in exascale: What is it?

Attack of the Killer Micros

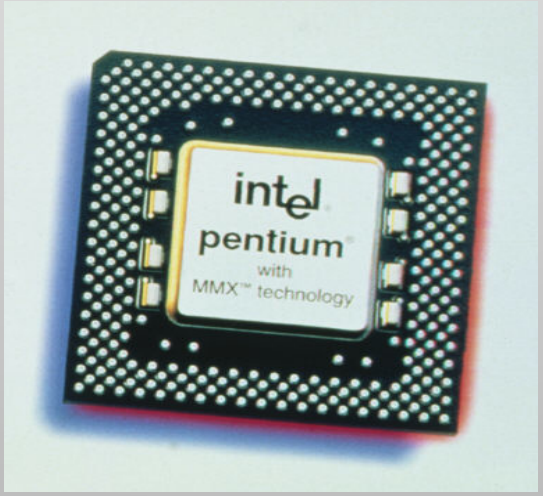
1970-1990

Attack of the Killer Cell Phones / GPUs





"Billions and Billions of Cell Phone Processors"

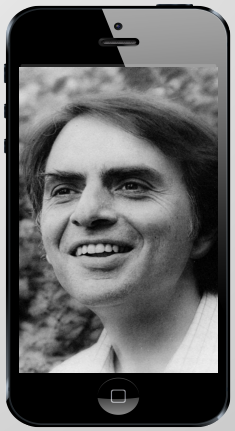


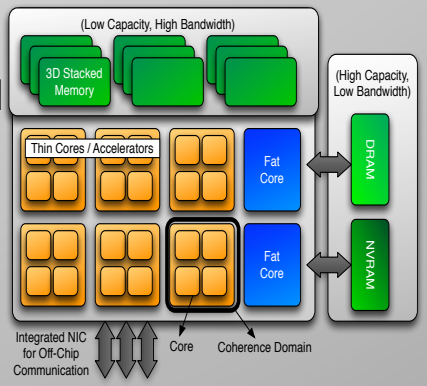
1990-2015

2015-????

Commonly asked questions in exascale: What is the problem?

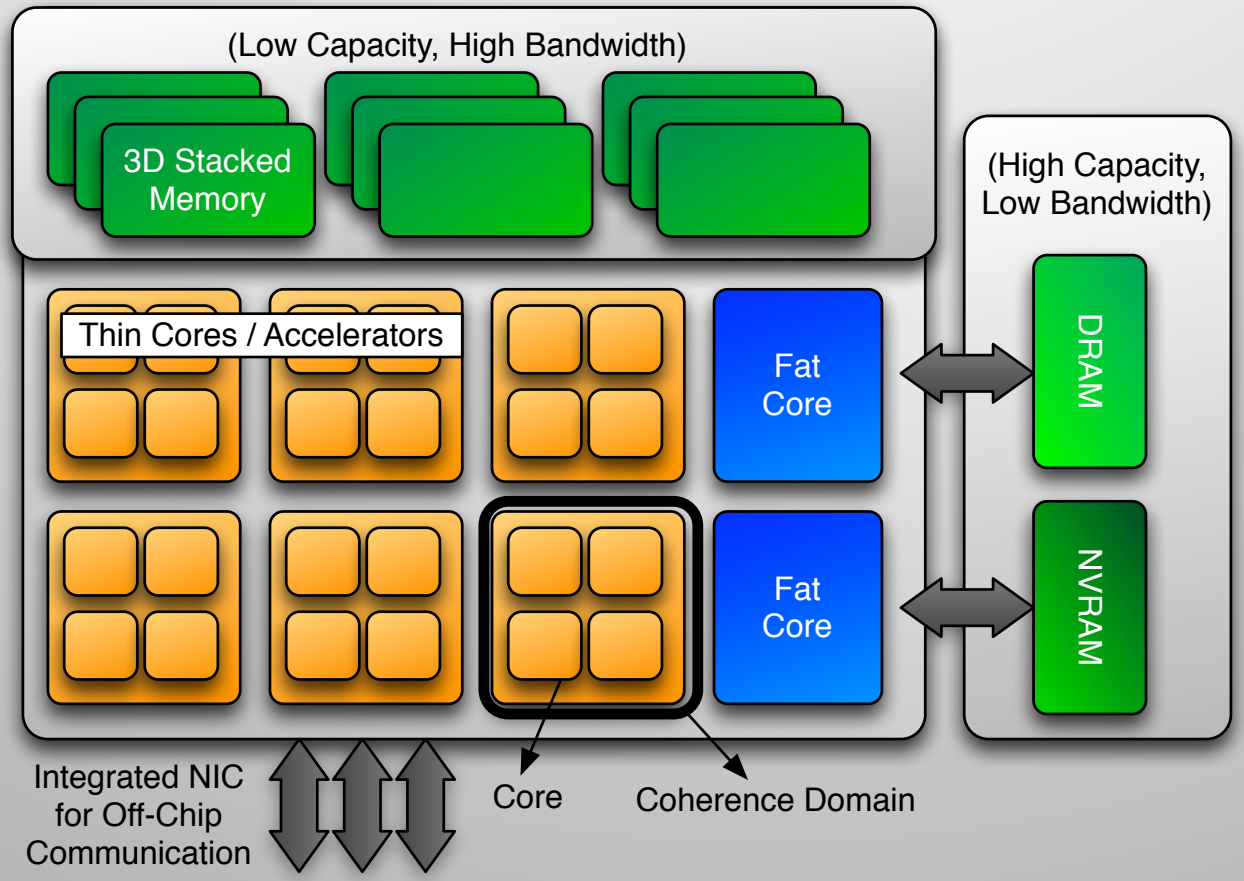
- **Power, energy, and heat dissipation** are the central issues
- Imagine a computer with **billions and billions** of cell phone processors (14MW) or **millions and millions** of throughput optimized cores, GPGPUs (20MW)
  - How do you program it to work on one science problem?
  - The architecture will be **heterogeneous** and **hierarchical**, with very high **flop/byte ratios**.
  - Single program multiple data bulk synchronous parallelism will no longer be viable.
- **Data Movement** will be expensive and computation will be cheap
  - Need to present the physics so the computation occurs where the data is!
  - Traditional global checkpoint/restart will be impractical: need local micro checkpoint (flash memory?)
- Simulation codes will need to become **fault tolerant and resilient**
  - Recover from soft and hard errors, and anticipating faults
  - Ability to drop or replace nodes and keep on running
  - The curse of silent errors

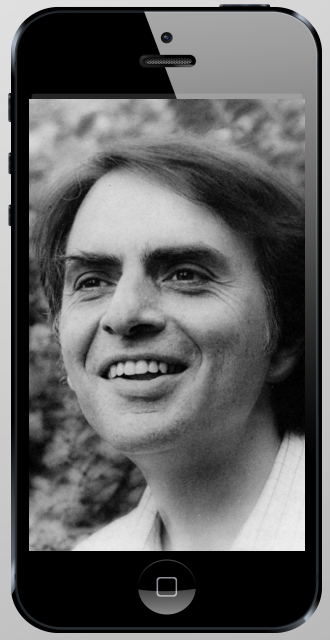




Common asked questions in exascale: What is it?

Billions of tasks all performing some calculation every nanosecond.
Conceptual Design: The Revolution is about the Node



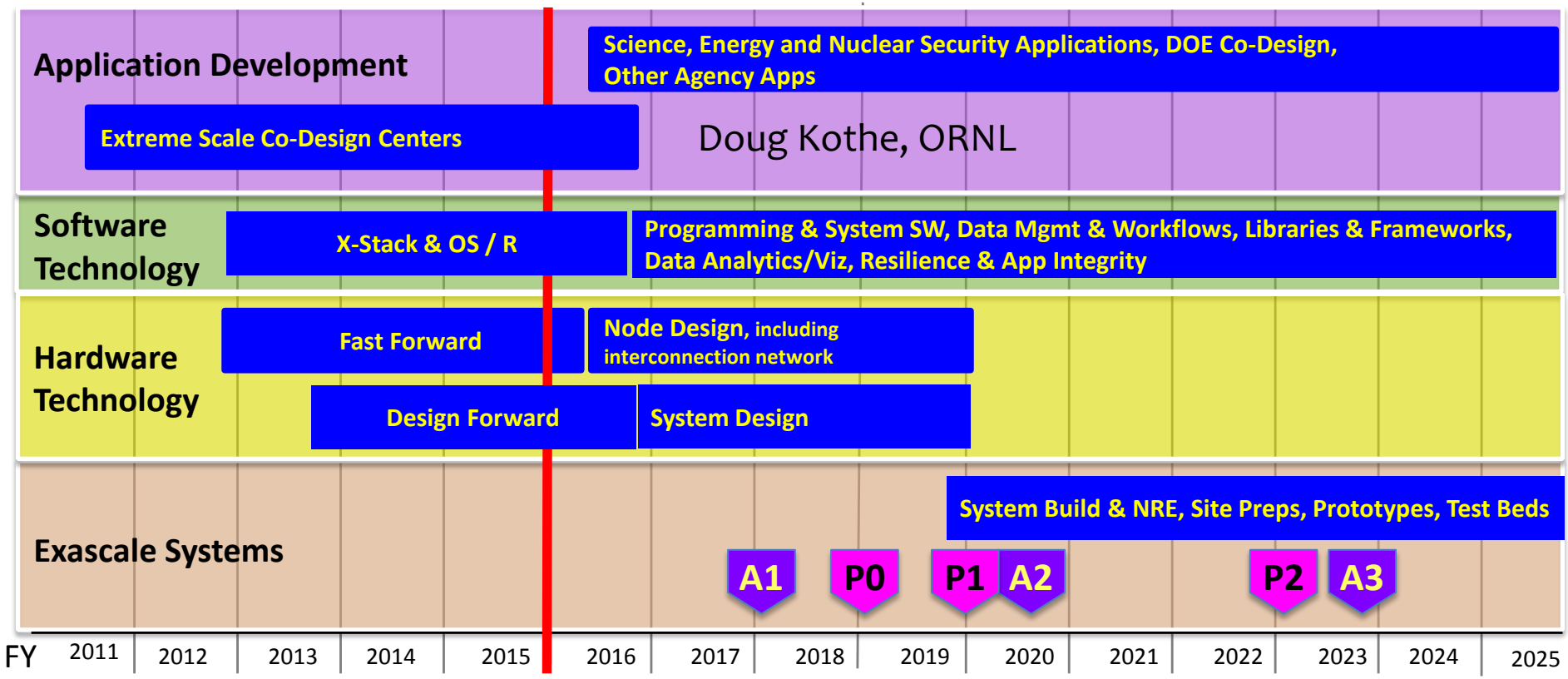


http://science.energy.gov/~/media/ascr/pdf/research/cs/cal_amm_v1_1.pdf

Petascale (exascale) application developers must optimize for a complex parallel machine

- (Inter-Node) System-level. No shared memory. Traditionally uses MPI to communicate data between disjoint address spaces.
- (Intra-Node) Cores. Modern nodes have multiple CPU cores. The work needs to be distributed across them. Old fashioned MPI is one option, but this increases surface to volume. Solutions such as OpenMP that acknowledge the shared address space (possibly NUMA) among the cores are probably preferred, especially in new code.
- (Intra-Node) Threads. Modern cores are supporting multiple hardware threads per node. Among other things, multiple threads per core cover latencies since some threads can typically proceed while others are stalled. Multiple threads may allow for better register usage, reduced pipeline stalls, etc.
- (Intra-Node) SIMD. We are now seeing quad-double SIMD units on intel and AMD hardware as well as BG/Q. Memory access need to be aligned to allow vector registers to be filled efficiently. Throwing away SIMD instructions is instantly giving away a factor of 4 in performance. On GPUs, warps are rather like SIMD instructions since all threads in a warp execute the same instruction.
- (Intra-Node) Functional Units. BG/Q has both an integer and a floating point unit for each core. It is the integer unit that loads data so in order to do useful processing you need to keep both active (to both load and process data). Note that any thread can only issue an instruction to one of the units per cycle so at least two threads are needed to fully exploit the units. It is also important to structure algorithms so that the use of the functional units is balanced.

Steve Binkley, Deputy Director, DOE Office of Advanced Scientific Computing Research



P0 Node Prototype

P1 Petascale Prototype

P2 Exascale Prototype

A1 CORAL Systems

A2 APEX Systems

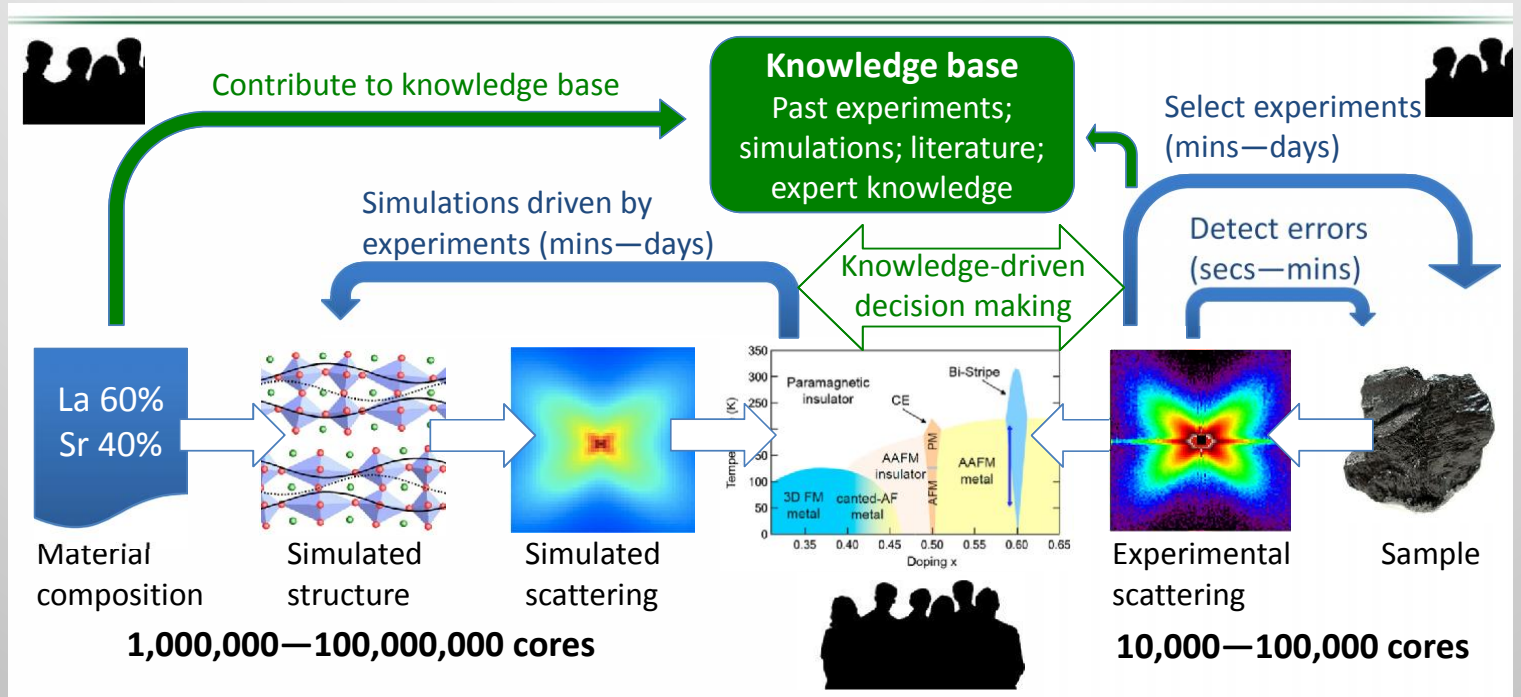
A3 Exascale Systems

Procurement of A1, A2 and A3 are not executed, but are highly leverage, by the ECI Project

Why do we care? As we speak, these applications are being prioritized, down-selected and multi-institute teams formed.

- **Call for input from all 17 DOE Laboratories released on May 31, 2015 to identify potential applications that could deliver new science capabilities on exascale systems. Input will be used by ASCR/ASC**
  - to identify additional key scientific areas for exascale discovery, and specific opportunities for new and existing scientific applications. They will also
  - Provide broad input on the kinds of partnerships and investments required to address technical challenges of exascale applications.
  - Short time frame – lab responses due June 15th --- **133 responses received**
- **NIH-NSF-DOE Request for Information to identify scientific research topics that need High Performance Computing (HPC) capabilities that extend 100 times beyond today's performance on scientific applications.**
  - Information will be used to assist agencies to construct a roadmap, build an exascale ecosystem required to support scientific research, and inform the research, engineering and development process. It is likely that a range of advanced capabilities will need to be developed to respond to the varied computing needs across science disciplines.
  - Released September 15, 2015 and due November 13, 2015 --- **115 responses received**

Common asked questions in exascale: Why do we care? Accelerating the Scientific and Design Processes



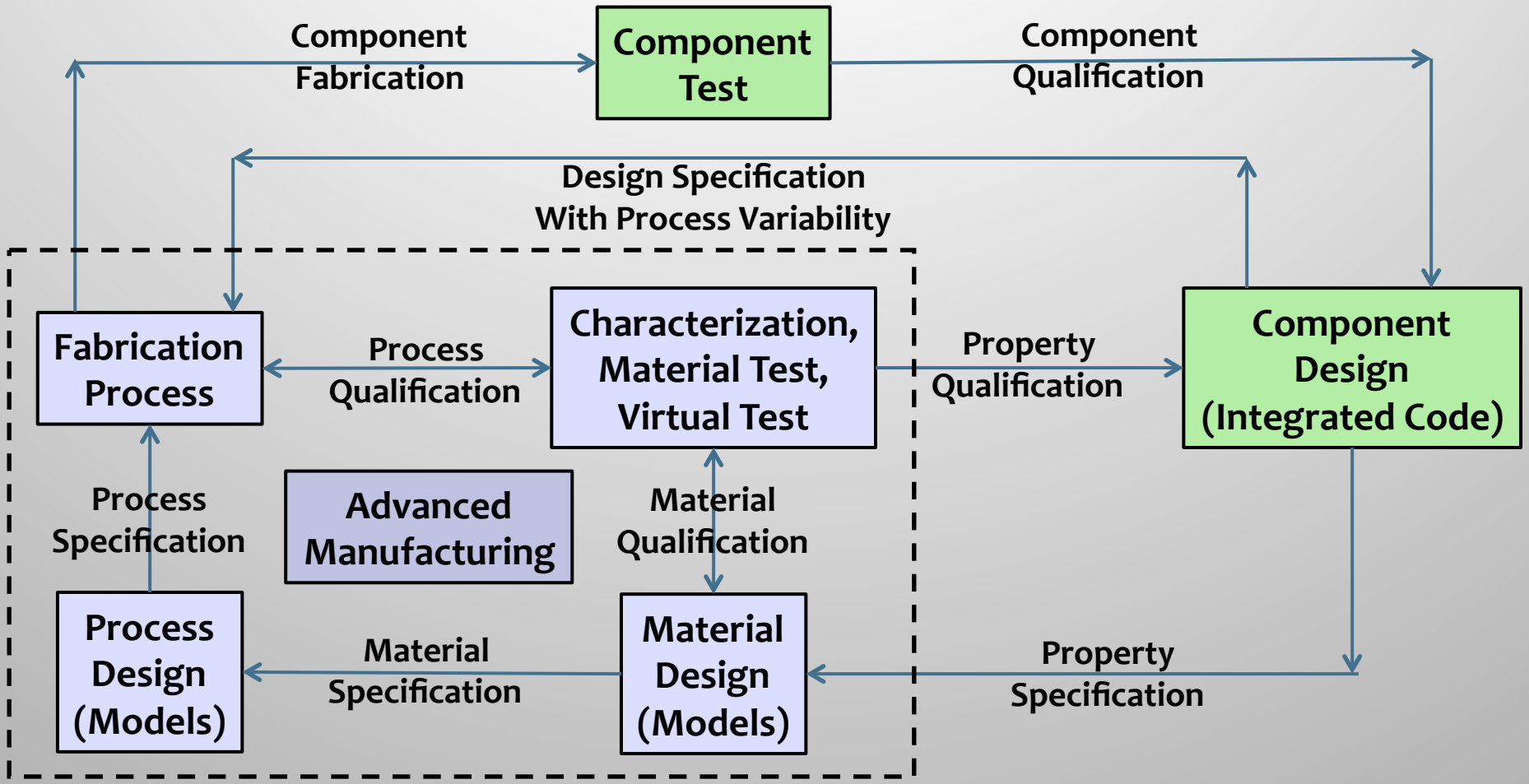
Many similar opportunities across a broad spectrum of DOE science:

Next-generation instruments will increase data rates by x1000 or more

Diffuse scattering images from Ray Osborn et al. (MSD, APS, MCS)

Ian Foster, ANL

Common asked questions in exascale: Why do we care?
S&E Use-Case: Accelerating Materials Development

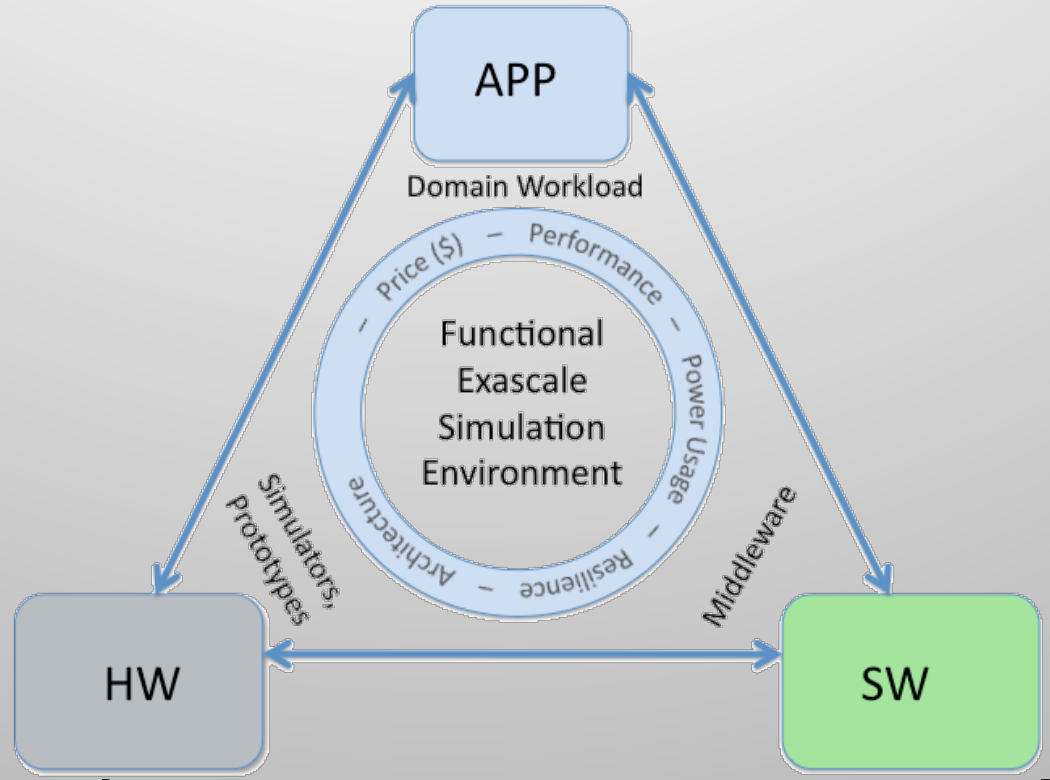


Objective: Agile Design and Qualification with Minimal Component-scale Test

Co-design requires an ongoing feedback between application, architecture, and middleware developers

- Coordinated efforts of Domain Scientists, Computer Scientists and Hardware Developers

Application Co-design Centers



Fast Forward
Design Forward

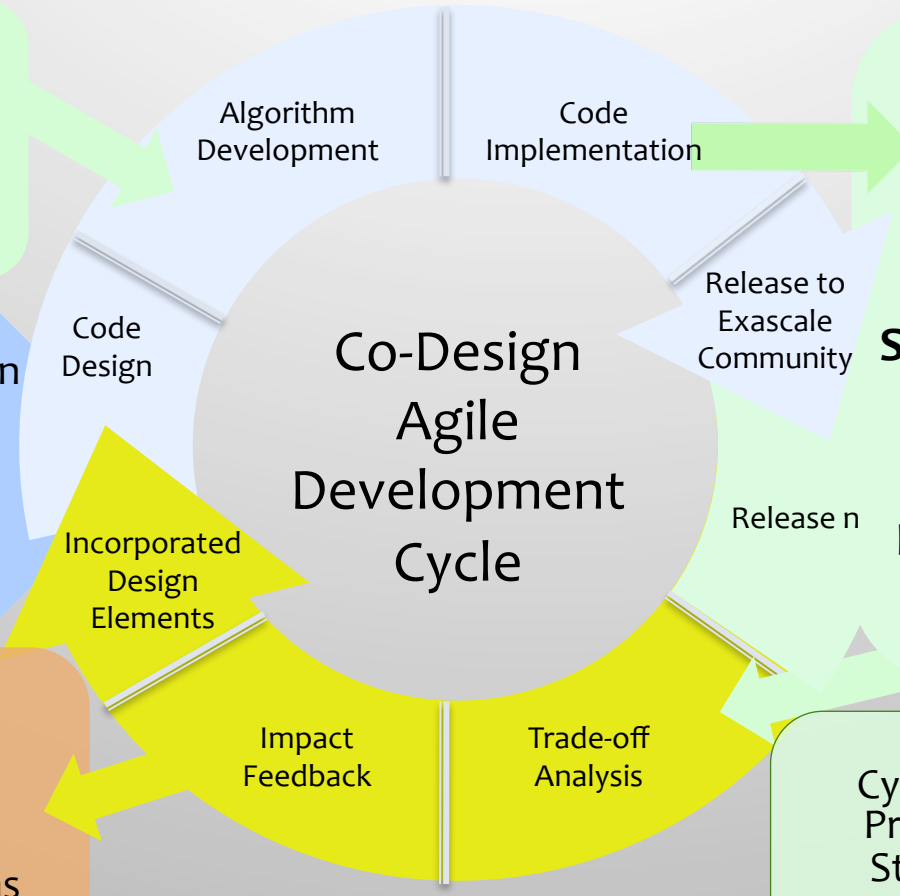
X-Stack
Exascale OS/R

The Solution: Application-Driven Co-design

Domain Science:
Domain Workload
Physical Models
Algorithms
Simulations

Preparation:
Science and Mission
Stakeholder Buy-in
Assemble Team
Implementation Plan
Development Plan

Cycle Artifacts:
R&D Backlog
Algorithm and Model Implementation
Proxy Applications
Architecture Evaluation

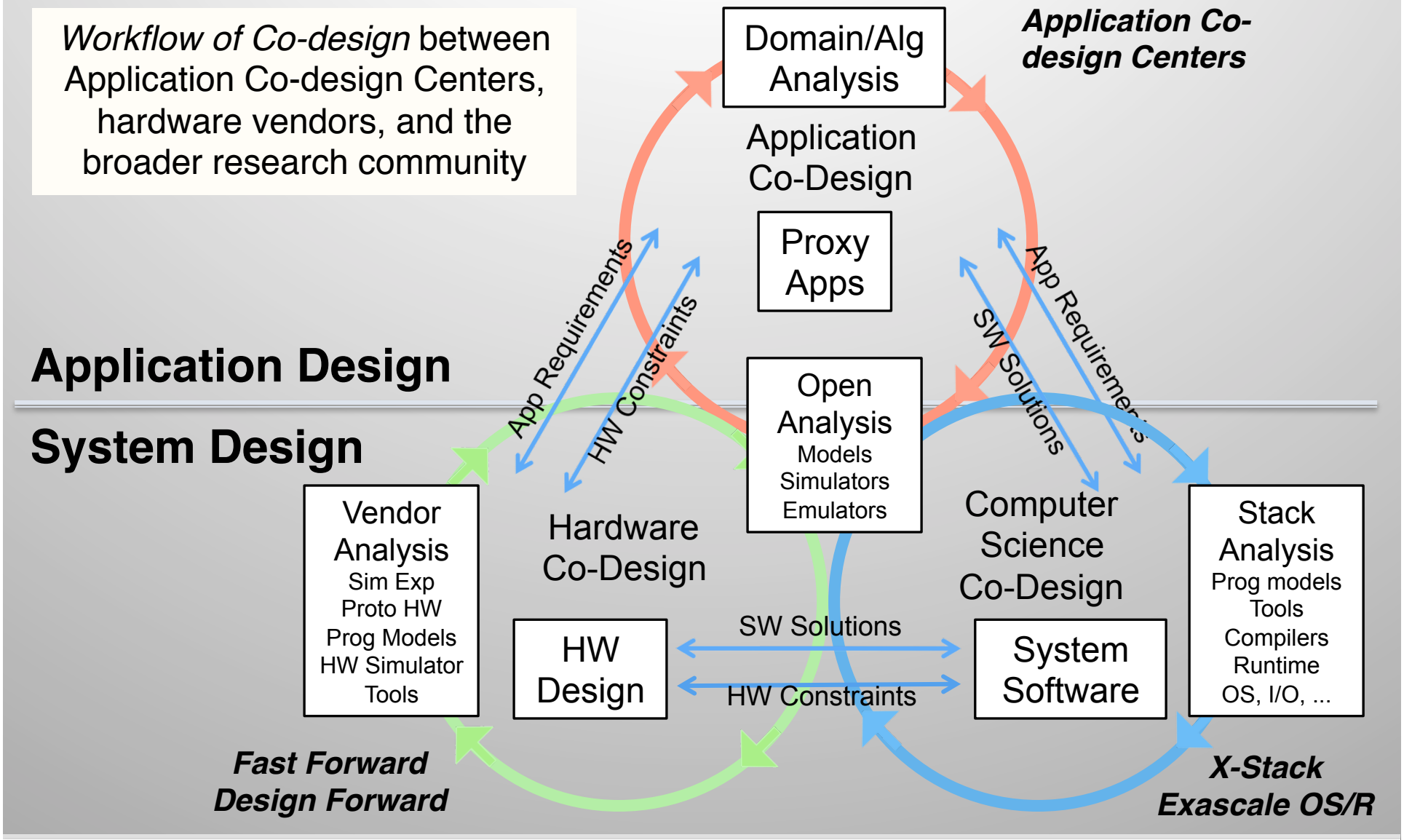


Exascale Community: Release Artifacts:
HW Requirements
SW Constraints
Proxy Applications
Documentation
Software Development:
ASCR X-stack, ASC CSSE
Data/Analysis
Hardware Development: Vendors, Fastforward, ASCR Advanced Architecture

Team Roles:
Cycle Master: Co-design PI
Project Team: Labs, Univ's
Stakeholders: ASCR, ASC, Vendors
Customers: Scientists, HW+SW Developers

Productive exascale development requires an ongoing feedback between application, architecture, and middleware developers

Workflow of Co-design between Application Co-design Centers, hardware vendors, and the broader research community



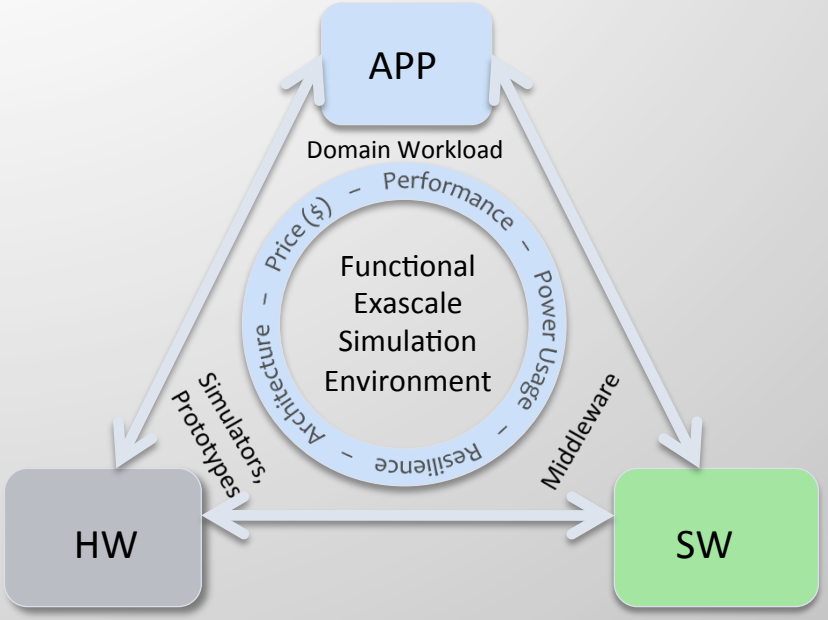
DOE currently funds three ASCR Co-design projects and a tri-Lab ASC Co-design project. More are expected in the future.

DOE Office of Science ASCR Co-design Projects:
Center for Exascale Simulation of Advanced Reactors (CESAR)
http://cesar.mcs.anl.gov
Director: Andrew Siegel (ANL/Uchicago), Deputy Director: Paul Fischer (ANL)
Center for Exascale Simulation of Combustion in Turbulance (ExaCT)
http://exactcodesign.org
Director: Jacqueline Chen (SNL), Deputy Director: John Bell (LBNL)
Exascale Co-Design Center for Materials in Extreme Environments (ExMatEX)
http://exmatex.lanl.gov
Director: Tim Germann (LANL), Deputy Director: Jim Belak (LLNL)

DOE NNSA ASC Co-design Project:
National Security Applications Co-Design Project (NSApp CDP)
Contacts: *Sriram Swaminarayan(LANL), Rob Neely(LLNL), Hoekstra/Heroux(SNL-A)*
http://proxyapps.lanl.gov

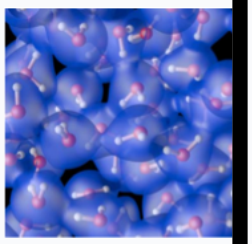
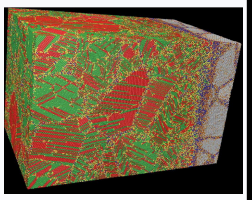
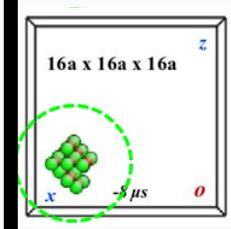
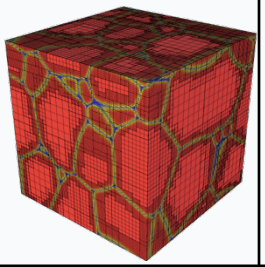
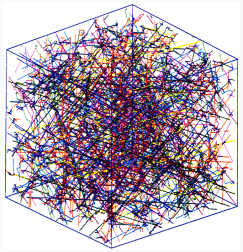
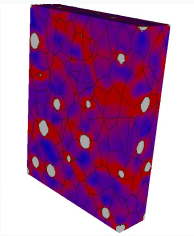
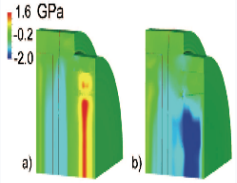
The Exascale Materials Co-design Center (ExMatEx) will Enable Vitual Test

- Our **goal** is to establish the interrelationship between hardware, middleware (software stack), programming models, and algorithms required to enable ***a productive exascale environment*** for multiphysics simulations of materials in extreme mechanical and radiation environments.
- We will exploit, rather than avoid, the greatly increased levels of concurrency, heterogeneity, and flop/byte ratios on the upcoming exascale platforms.



- Our **vision** is an uncertainty quantification (UQ)-driven *adaptive physics refinement* in which meso- and macro-scale materials simulations spawn micro-scale simulations as needed.
  - This *task-based* approach leverages the extensive concurrency and heterogeneity expected at exascale while enabling fault tolerance within applications.
  - The programming models and approaches developed to achieve this will be broadly applicable to a variety of multiscale, multiphysics applications, including astrophysics, climate and weather prediction, structural engineering, plasma physics, and radiation hydrodynamics.

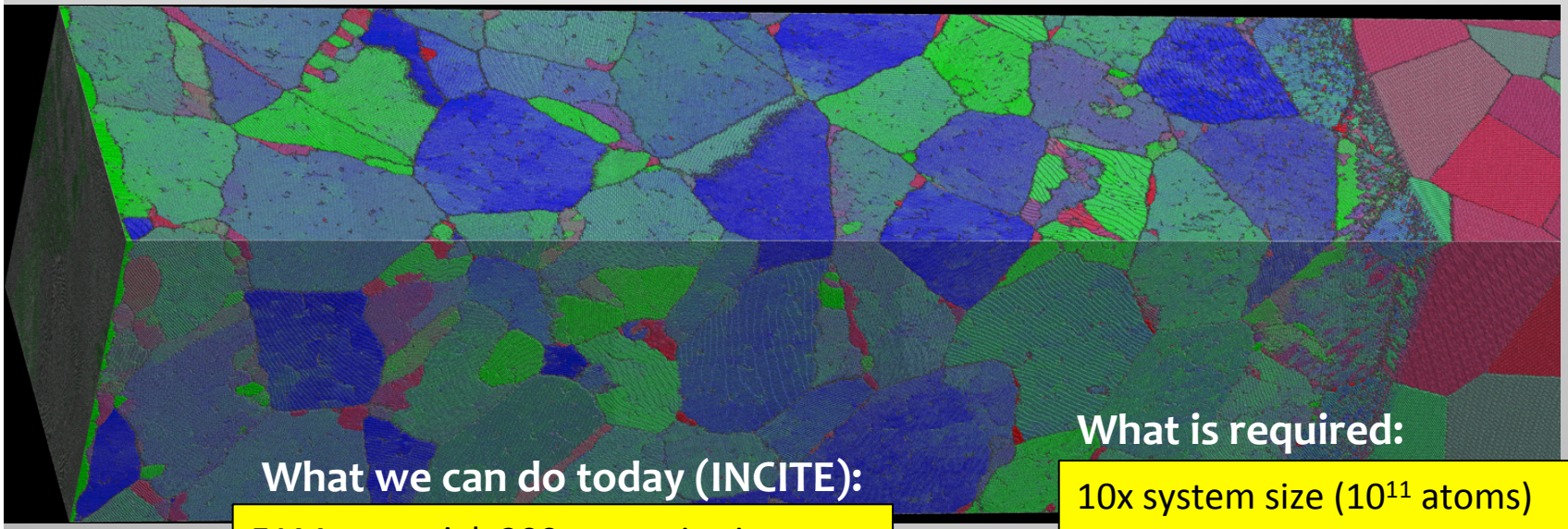
Seven Pillars of Computational Materials Science

Ab-initio	Atoms	Long-time	Microstructure	Dislocation	Crystal	Continuum
Inter-atomic forces, EOS, excited states	**Defects and interfaces, nucleation**	**Defects and defect structures**	**Meso-scale multi-phase, multi-grain evolution**	**Meso-scale strength**	**Meso-scale material response**	**Macro-scale material response**
						
Code: Qbox/ LATTE Motif: Particles and wavefunctions, plane wave DFT, ScaLAPACK, BLACS, and custom parallel 3D FFTs Prog. Model: MPI + CUBLAS/CUDA	Code:SPaSM/ ddcMD/CoMD Motif: Particles, explicit time integration, neighbor and linked lists, dynamic load balancing, parity error recovery, and *in situ* visualization Prog. Model: MPI + Threads	Code: SEAKMC Motif: Particles and defects, explicit time integration, neighbor and linked lists, and *in situ* visualization Prog. Model: MPI + Threads	Code: AMPE/GL Motif: Regular and adaptive grids, implicit time integration, real-space and spectral methods, complex order parameter Prog. Model: MPI	Code: ParaDiS Motif: "segments" Regular mesh, implicit time integration, fast multipole method Prog. Model: MPI	Code: VP-FFT Motif: Regular grids, tensor arithmetic, meshless image processing, implicit time integration, 3D FFTs. Prog. Model: MPI + Threads	Code: ALE3D/ LULESH Motif: Regular and irregular grids, explicit and implicit time integration. Prog. Model: MPI + Threads

Jim Belak, ExMatEx

Use case: competing dislocation, twinning, and/or phase transitions under shock loading

- Direct non-equilibrium molecular dynamics simulation matching time and length scales of planned LCLS experiments
  - ~1-2 μm thick nanocrystalline samples (Cu, Ti, Fe, Ta), ~400 nm grain size
  - Laser drive: 10-20 ps rise time, 150 ps duration
  - 50 fs duration X-ray "snapshot" interrogation pulses at 10 ps intervals



NEMD simulation of shocked nc-Ta on Cielito (R. Ravelo, LANL/ UTEP)

What we can do today (INCITE):

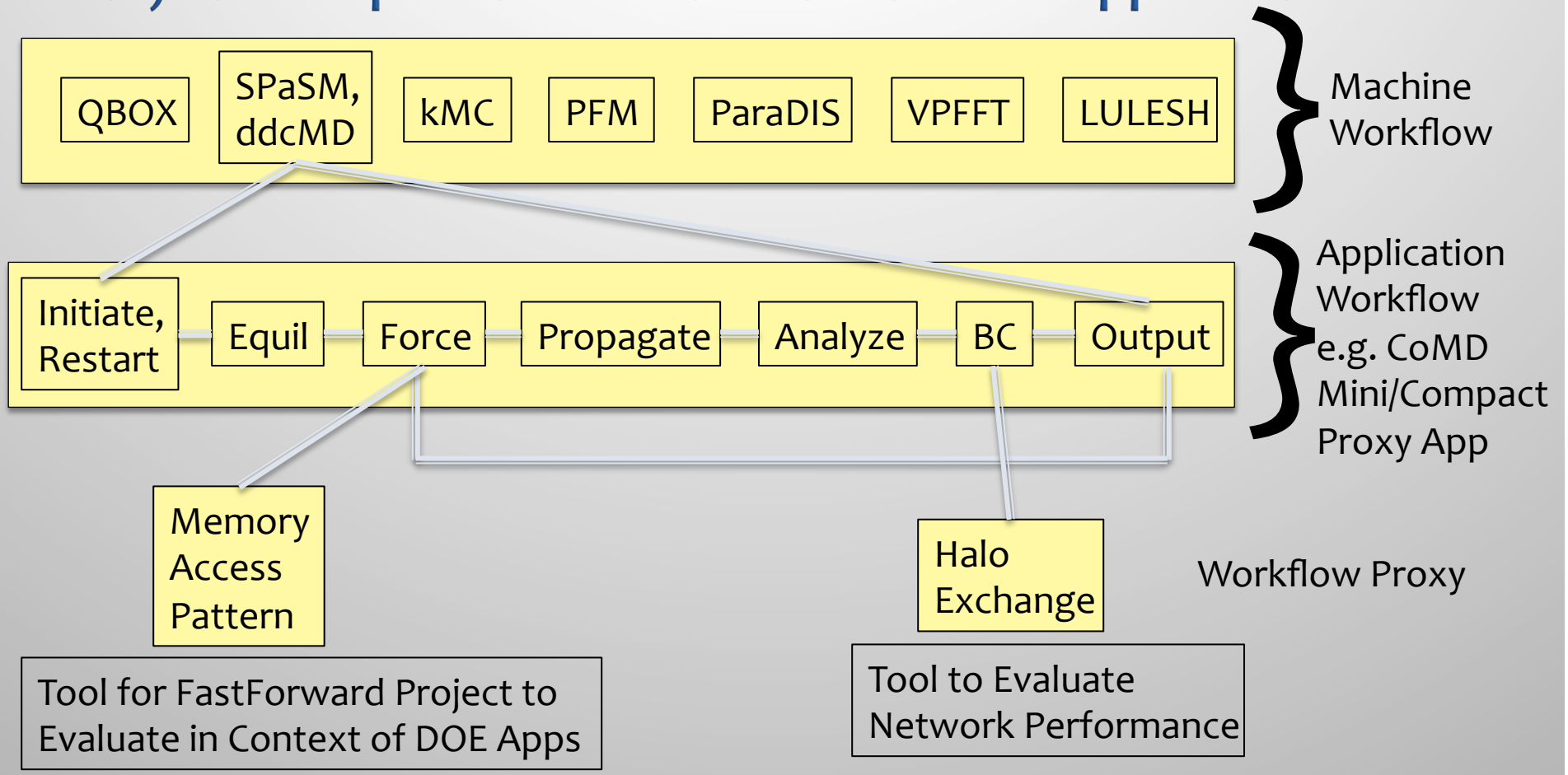
EAM potential, 200 nm grain size
10^{10} atoms (0.5 μm x 0.5 μm x 1.5 μm)
Simulation time: 4 nsec (10^6 steps)
Wall clock: 2 days on Mira (½ Sequoia)

What is required:

10x system size (10^{11} atoms)
1 μm x 1 μm x 2 μm, 400 nm grain size

More accurate MGPT potential: 100x
3 weeks on exascale system

Proxy App: a small application code that proxifies (stands for) some aspect of the workflow of a full application



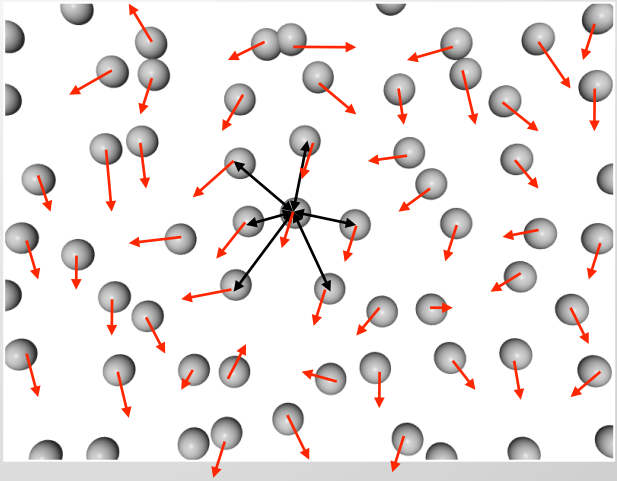
Proxy Apps are fundamentally different from Benchmark Apps. They enable the lessons learned from Co-design to be incorporated back into the full app.

Molecular Dynamics (MD) – CoMD

Molecular dynamics: particles interact via explicit interatomic potentials and evolve in time according to Newton's equations of motion:

$$\dot{\mathbf{r}}_i = \mathbf{p}_i / m_i \qquad \dot{\mathbf{p}}_i = \mathbf{f}_i$$

$$\mathbf{f}_i = m_i \ddot{\mathbf{r}}_i = -\sum_j \nabla V_{ij}$$

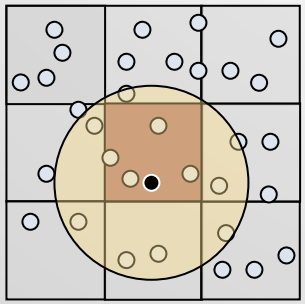


Interaction potentials determine both the physics and computer science

- Complex potentials are more accurate, but can require many more floating point operations.
- Locality of potential informs parallelization strategy, e.g. short-ranged potentials require only point to point communication.

Challenge Problem: Can you use an exascale computer with billion-way parallel parallelism to simulate longer in time? (not just more atoms)

How are forces calculated in a parallel MD code?



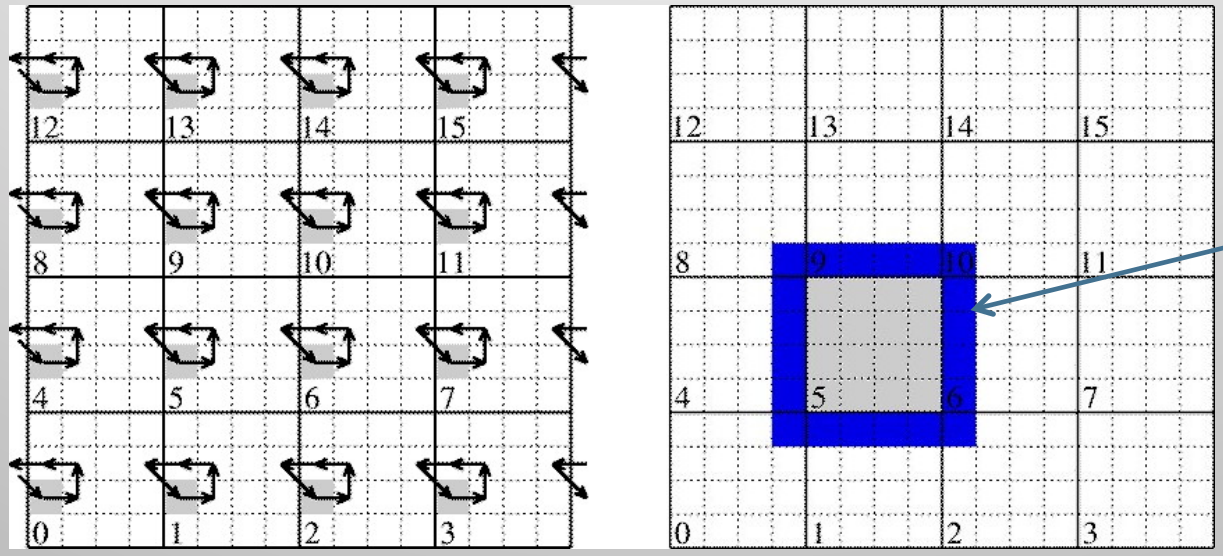
~ 20 atoms in each box

⇒ each atom interacts with 540 other atoms

⇒ However, only ~70 atoms lie within cutoff

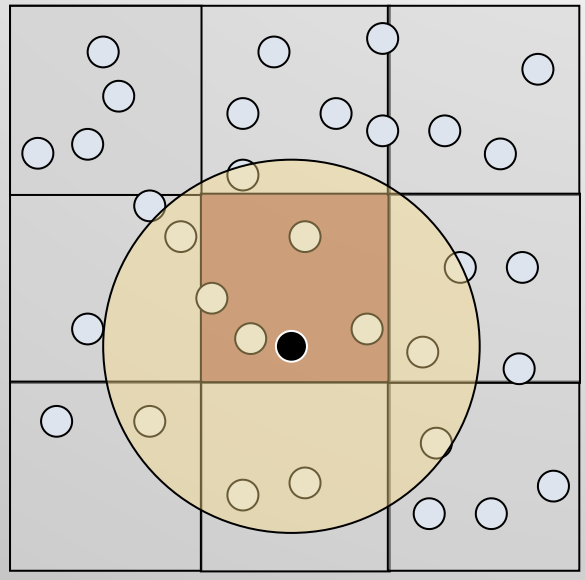
⇒ Lots of wasted work

⇒ We need a means of rejecting atoms efficiently even within this reduced set

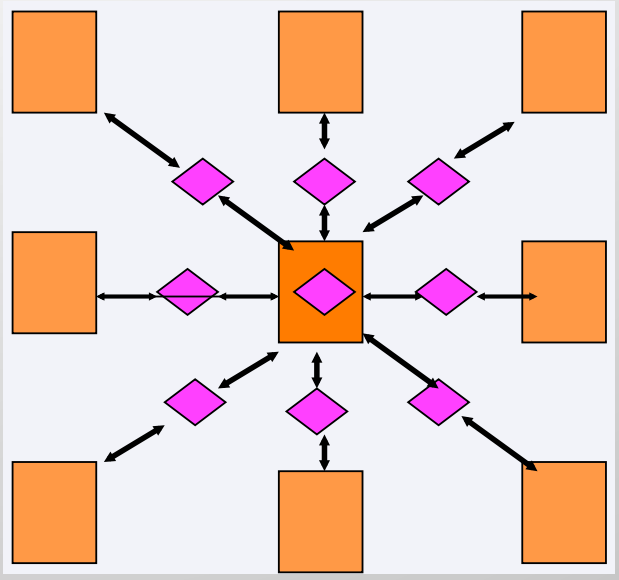


Fixed geometric domain decomposition limits scalability for any heterogeneous problem. Furthermore, statistical fluctuations in the force calculation between processors leads to an effective scalar term that also limits scaling (Amdahl's law).

Task-based programming: Change in perspective from "worker" to "work." Do the work where the data is!



MD "MPI"
Worker Decomposition



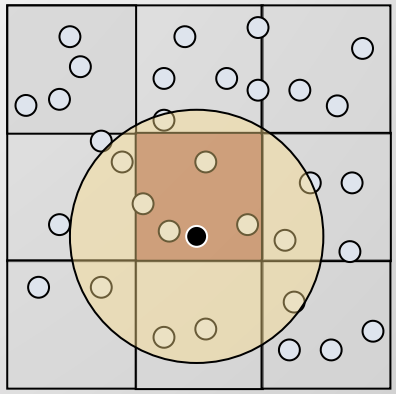
Charm++
Work Decomposition

c.f. http://charm.cs.uiuc.edu/

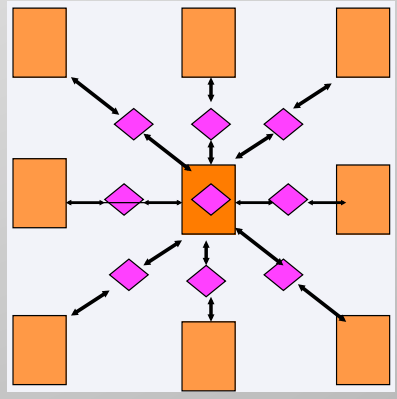
2014 LLNL Co-Design Summer School focus on Task-based Programming (Whiteboard - Priceless!)

Students and Programming Models

- Sam Reeve (Purdue) LAMMPS, CoMD, leanMD
- Riyaz Haque (UCLA), CNC (Habenero-C)
- Luc Jualmes (Barcelona) OmpSs, Chapel
- Sameer Abu Asal (LSU) C++11 futures, HPX
- Aaron Landmehr (Delaware) OCR
- Sanian Gaffer (NM-State) UPC++
- Gheorghe-Teodor Bercea (Imperial) PyOP2 DSL
- Zach Rubinstein (Chicago), Troels Henriksen (Copenhagen) Embedded DSL

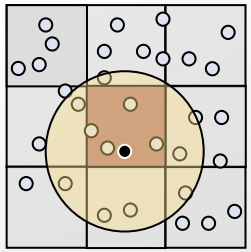


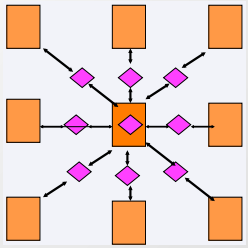
Worker

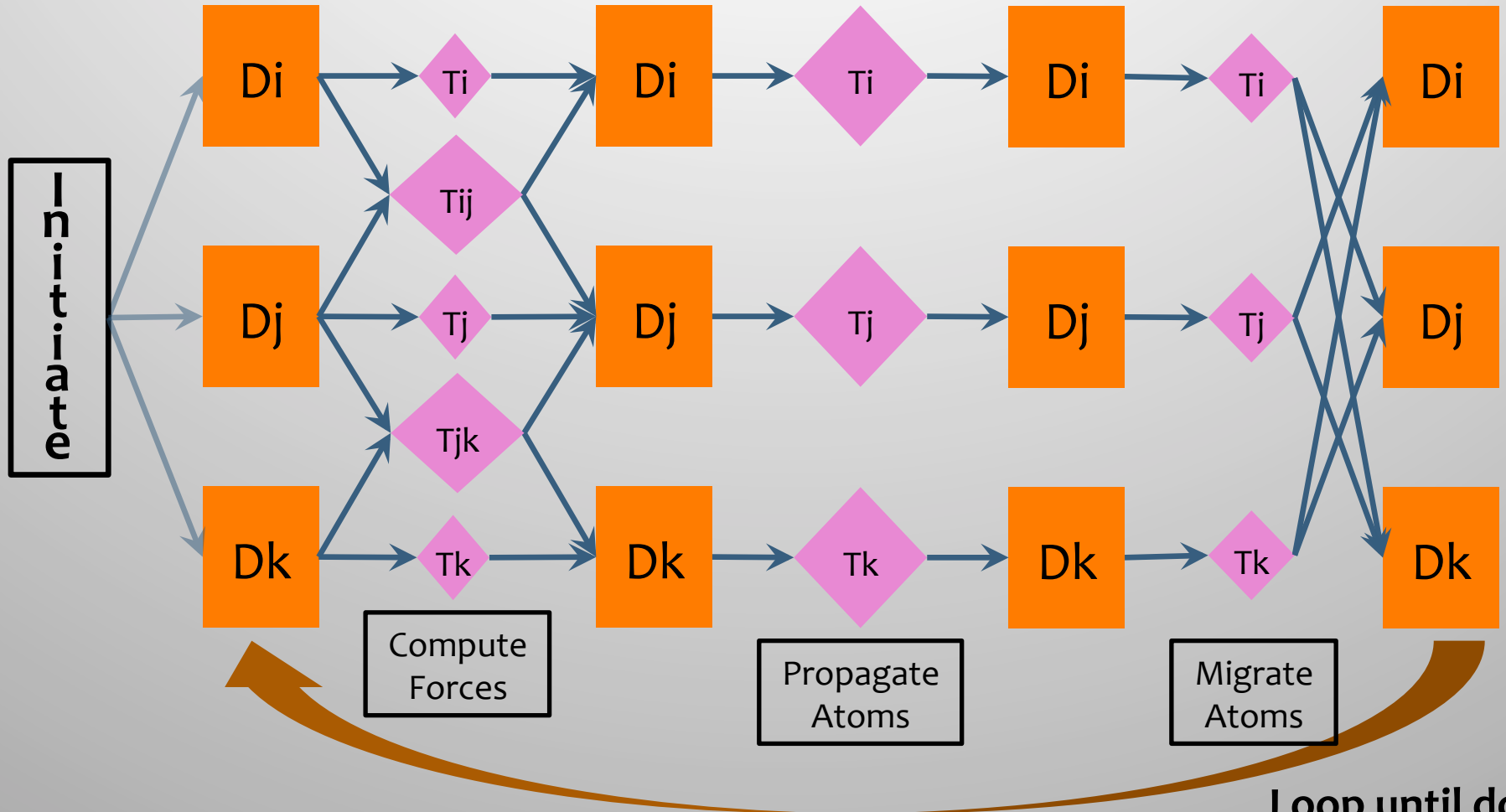


Work

Task and data dependency graph







Loop until done

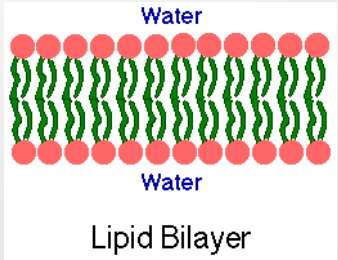
All Models have Merit

- What did we learn:
  - No Free Lunch: Strong "Runtime" required to coordinate Work Decomposition
  - Would like to be able to give runtime hints:
    - Schedule tasks to minimize race conditions
    - Schedule tasks to minimize data movement
    - Schedule tasks for data reuse (cache)
  - Data locality matters:
    - static specification of data layout is norm (there are exceptions)
    - need dynamic for load balancing and data migration
    - Q: Can the runtime discover data decomposition patterns? E.g. surface to volume from heterogeneous communication

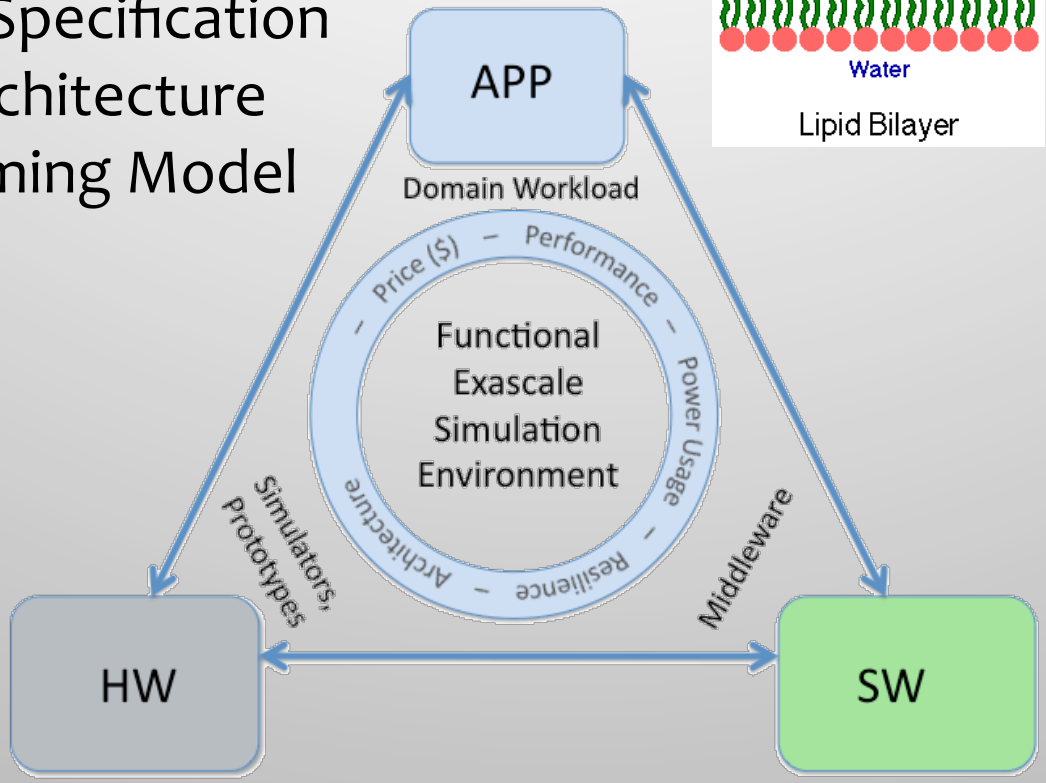
Creating Community Codes (1994) LAMMPS CRADA: Cray (Carpenter), Bristol-Myers (Stouch), Dupont (Lustig), LLNL (Belak), Sandia (Plimpton)

Three Essential Elements:

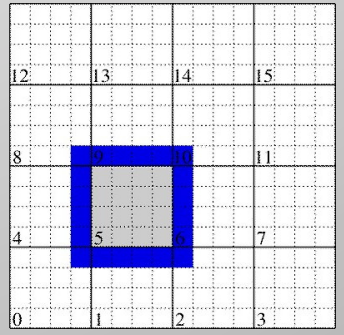
- Problem Specification
- Target Architecture
- Programming Model



Diffusion of a small drug molecule through a bio-membrane



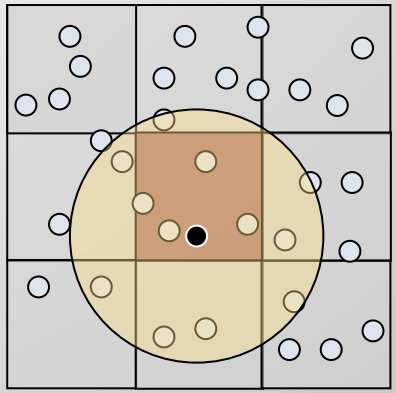
Cray T3D, Clock Speed: 0.15GHz, Date: 1994
Microprocessor Peak Teraflops: 0.02, Memory: 0.01TB
Processors: 128. Power Consumption: 41.40 kW



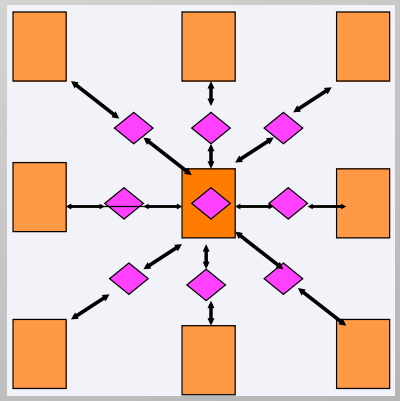
SPMD, MPI, Geometric domain decomposition

1997: LAMMPS became a successful community code because of Steve's idea to release it as open source, the ubiquitous Beowolf cluster and the stability of MPI.

- 20 Years Ago
  - Hardware: Linux Beowolf Cluster
  - Software Programming Model: MPI/SPMD
  - Application Code: LAMMPS
- 5+ Years in the Future
  - Hardware: Exascale (Petascale in a Rack)
  - Software Programming Model: MPI+X, Q? Task-based with "Control" of Data Locality (User and Runtime)
  - Application Codes???



Worker



Work

The fundamental challenges facing the development of application codes for extreme-scale computing hardware are:

- **sustainability** (the code lives for multiple generations of hardware);
- **portability** (the code compiles and runs on hardware from multiple vendors ranging from desktop to high performance computing);
- **performance** (the code makes optimal use of available hardware resources); and
- **productivity** (the code is maintainable, refactorable, usable, and easily evolved).

Application codes that are successful at these challenges separate concerns:

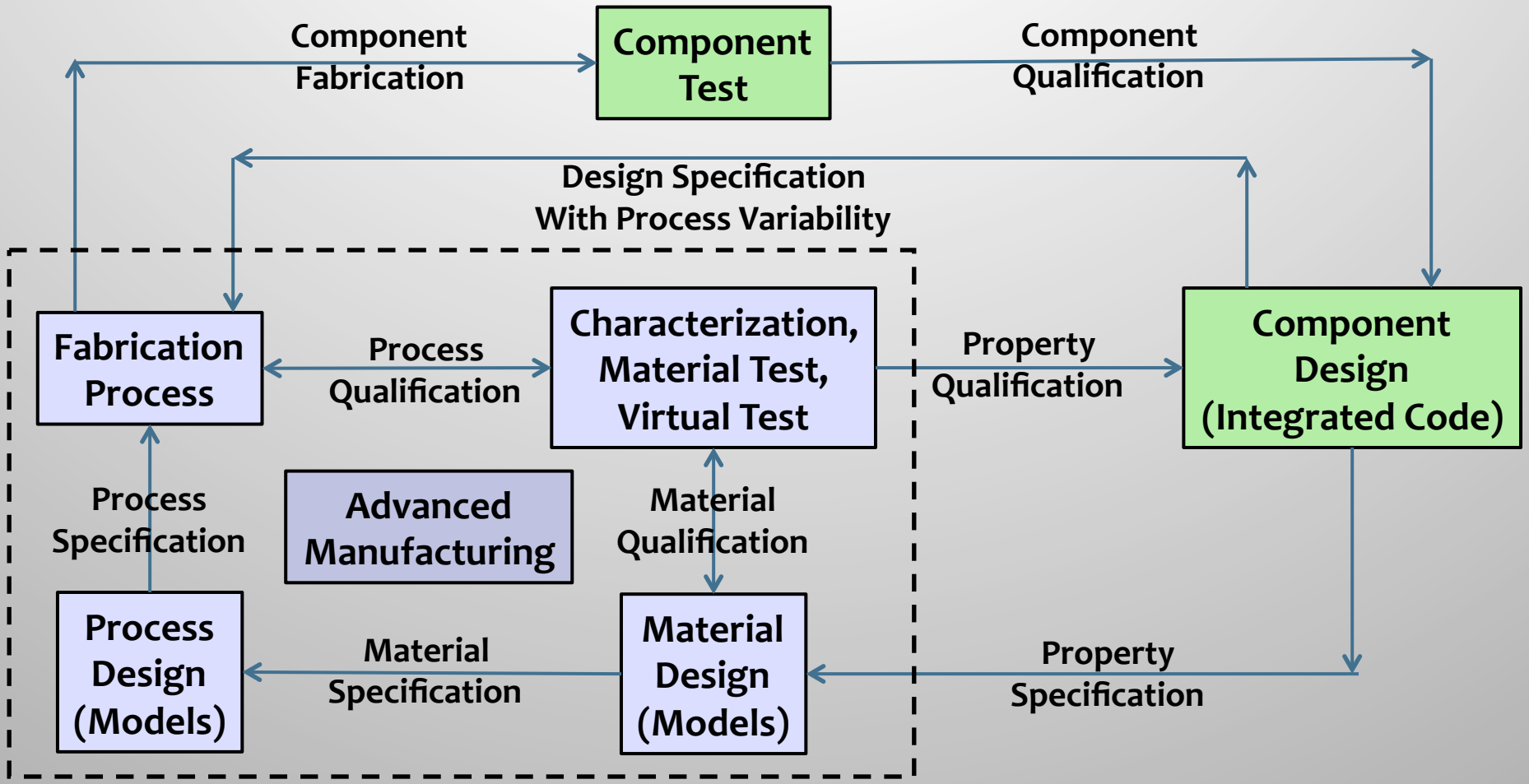
- The **encapsulation of computation** so that no function has side effects that inadvertently effect other functions;
- The **regularization of irregular objects** by pre-analysis to present application work to best utilize available hardware;
- **The separation of work from worker:** focus on the application work and present the work so that the OS/RT can best schedule the work on available hardware; and
- **The separation of application data layout from machine memory layout:** present the data layout as required by the application work and use domain maps to best utilize machine physical memory layout

Zoology: MPI+X, OpenMP, RAJA/Kokkos, Task (Charm++, Legion, …)

MPI + X

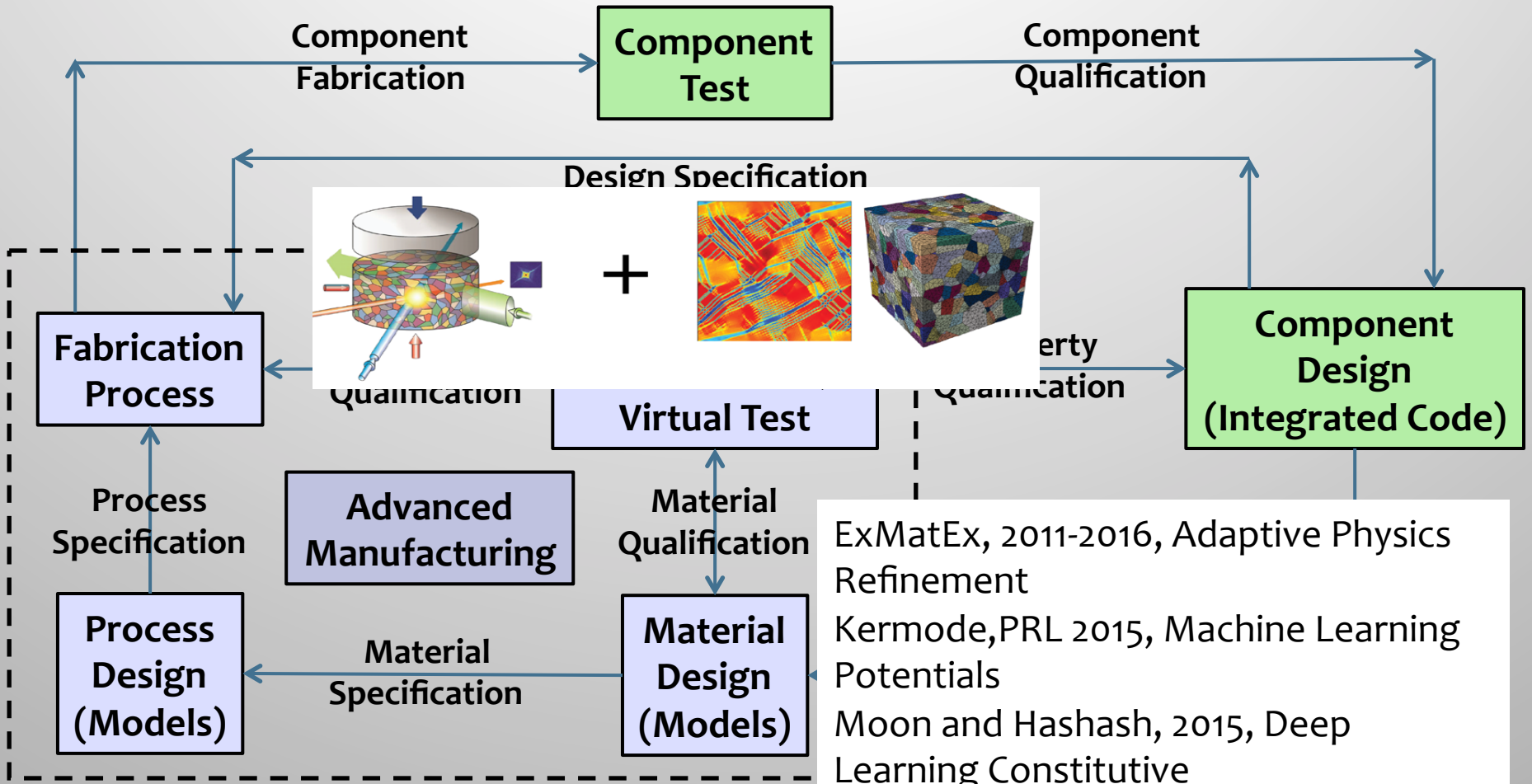
- OpenMP
- C++ Frameworks (RAJA, Kokkos)
- OpenMP-Tasks, Legion, Charm++, Uintah, …

Common asked questions in exascale: Why do we care?
S&E Use-Case: Accelerating Materials Development



Objective: Agile Design and Qualification with Minimal Component-scale Test

Common asked questions in exascale: Why do I care?
S&E Use-Case: Accelerating Materials Development



ExMatEx, 2011-2016, Adaptive Physics Refinement
Kermode,PRL 2015, Machine Learning Potentials
Moon and Hashash, 2015, Deep Learning Constitutive
Han, 2015, Process First-class Citizen

Objective: Agile Design and Qualification w

Use Case: Shaped-charge jets, breakup and 3D effects (e.g. spinning) require crystal plasticity and anisotropy

What is required:

Resolution: 10^{12} zones (10 cm cube)

Simulation time: 100 μsec (10^5 steps)

Strain rate: 10^6 /sec

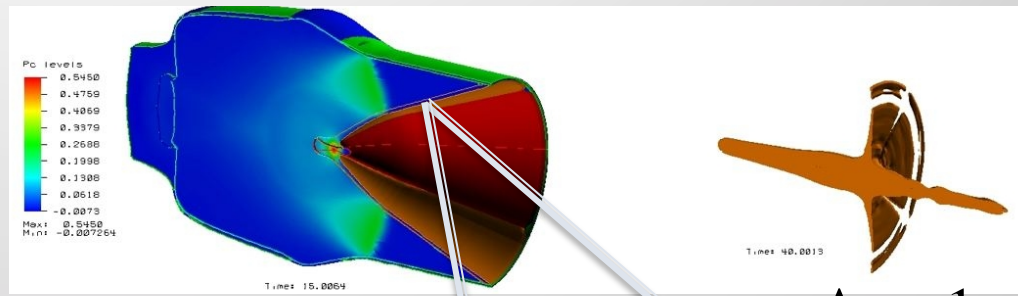
Strain: 1-3

Using Small Strain Crystal Plasticity Model: ~10^4 sec (~3 h) wall clock on 10^9 cores

Large Strain Crystal Plasticity Model: 10x

Twinning / Scale Bridging Model: 100x

ALE3D simulation of shaped-charge jet (Rose McCallen, LLNL)



$\Delta\varepsilon \geq 1$

What we can do today:

Crystal plasticity simulation of high rate deformation (Nathan Barton, LLNL)

Model: Small Strain Crystal Plasticity

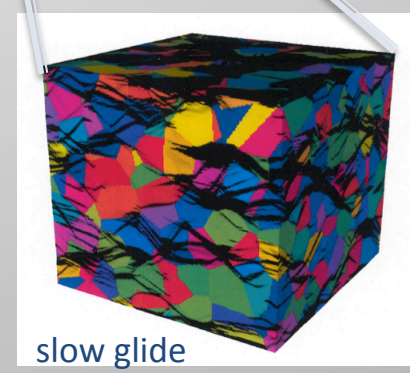
Number Zones: 10^7 (100 micron cube)

Simulation time: 10 μsec (10^4 steps)

Strain rate: 10^6 /sec

Strain: 0.15

Wall Clock: 1 day on 1/10 Cielo

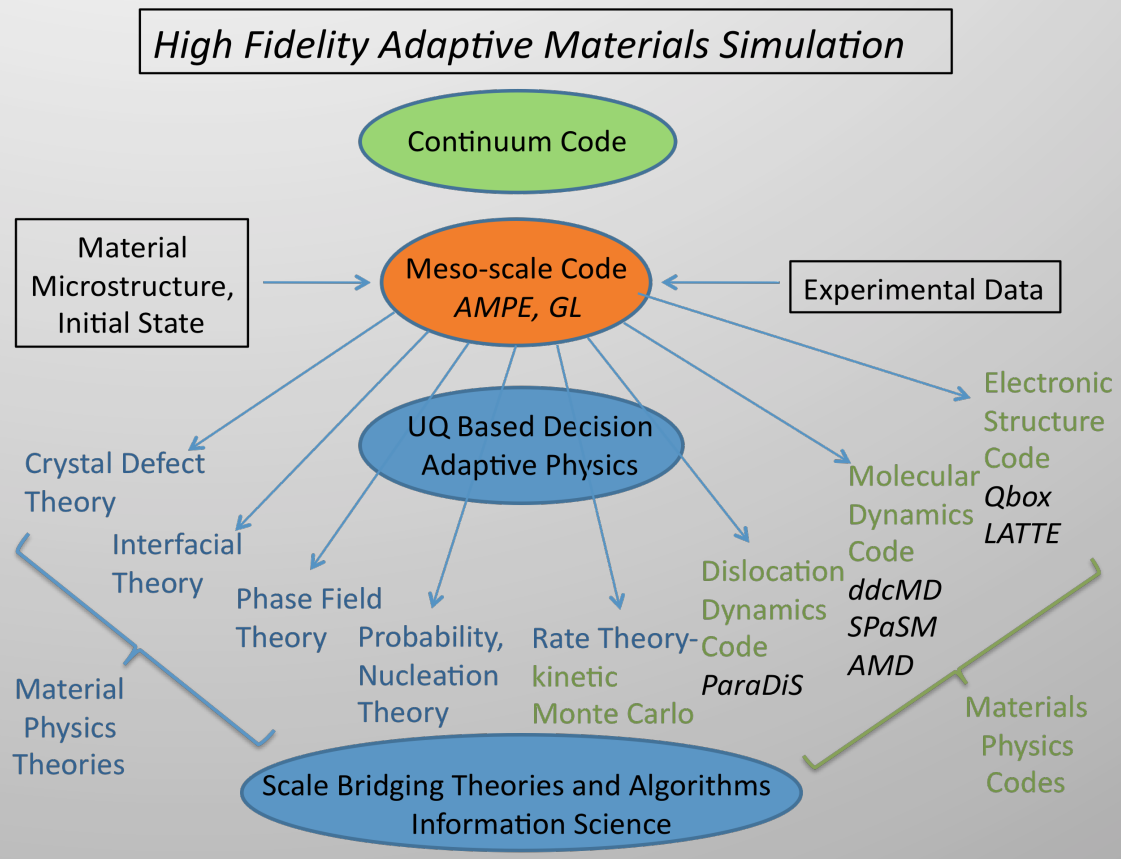


$\Delta\varepsilon = 0.15$

Objective: Full utilization of exascale concurrency and locality

- Task-based embedded Scale-Bridging escapes the traditional synchronous SPMD paradigm and exploits the heterogeneity expected in exascale hardware.

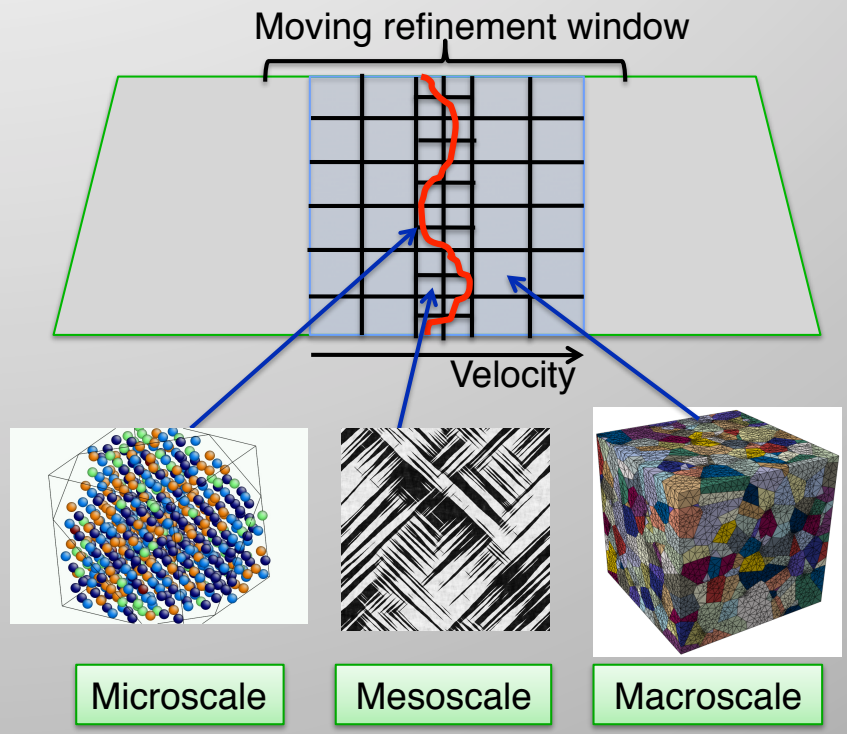
- To achieve this, we are developing a UQ-driven *adaptive physics refinement* approach.
- Coarse-scale simulations dynamically spawn tightly coupled and self-consistent fine-scale simulations as needed.
- This *task-based* approach naturally maps to exascale heterogeneity, concurrency, and resiliency issues.



Embedded Scale-Bridging Algorithms

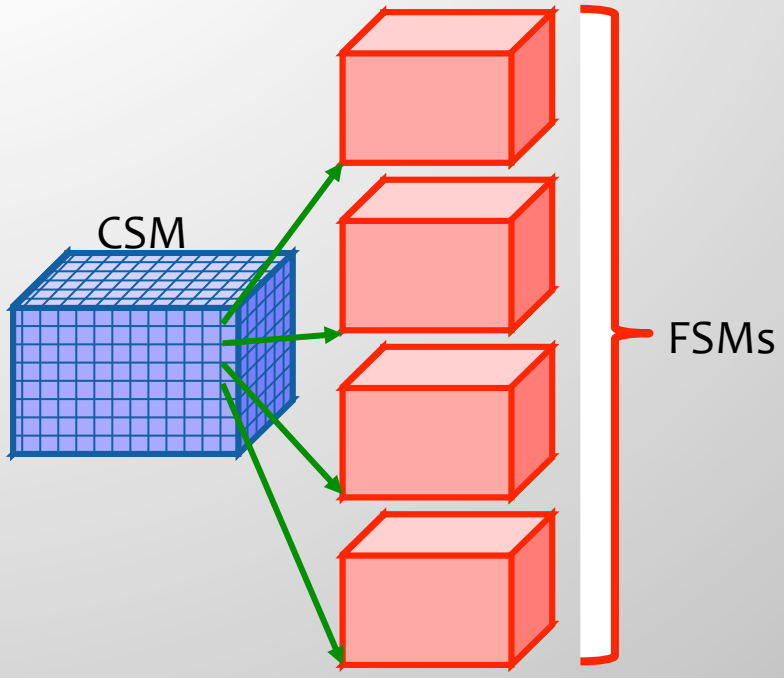
- Our goal is to introduce more detailed physics into computational materials science applications in a way which escapes the traditional synchronous SPMD paradigm and exploits the heterogeneity expected in exascale hardware.

- To achieve this, we are developing a UQ-driven *adaptive physics refinement* approach.
- Coarse-scale simulations dynamically spawn tightly coupled and self-consistent fine-scale simulations as needed.
- This *task-based* approach naturally maps to exascale heterogeneity, concurrency, and resiliency issues.



Direct multi-scale embedding requires full utilization of exascale concurrency and locality

- *Brute force multi-scale coupling:* Full fine scale model (FSM, e.g. a crystal plasticity model) run for every zone & time step of coarse scale mode (CSM, e.g. an ALE code)
- *Adaptive Sampling:*
  - Save FSM results in database
  - Before running another FSM, check database for FSM results similar enough to those needed that interpolation or extrapolation suffices
  - Only run full FSM when results in database not close enough

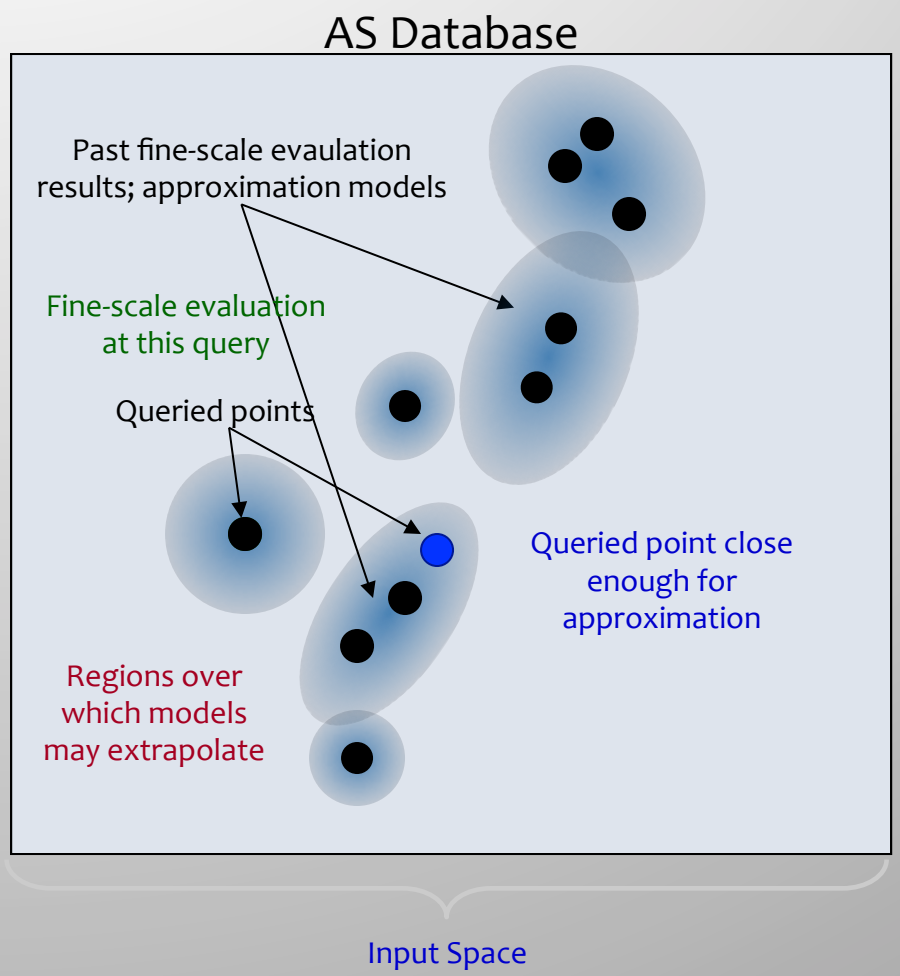


- Heterogeneous, hierarchical MPMD algorithms map naturally to anticipated heterogeneous, hierarchical architectures
- Escape the traditional bulk synchronous SPMD paradigm, improve scalability and reduce scheduling
- Task-based MPMD approach leverages concurrency and heterogeneity at exascale while enabling novel data models, power management, and fault tolerance strategies

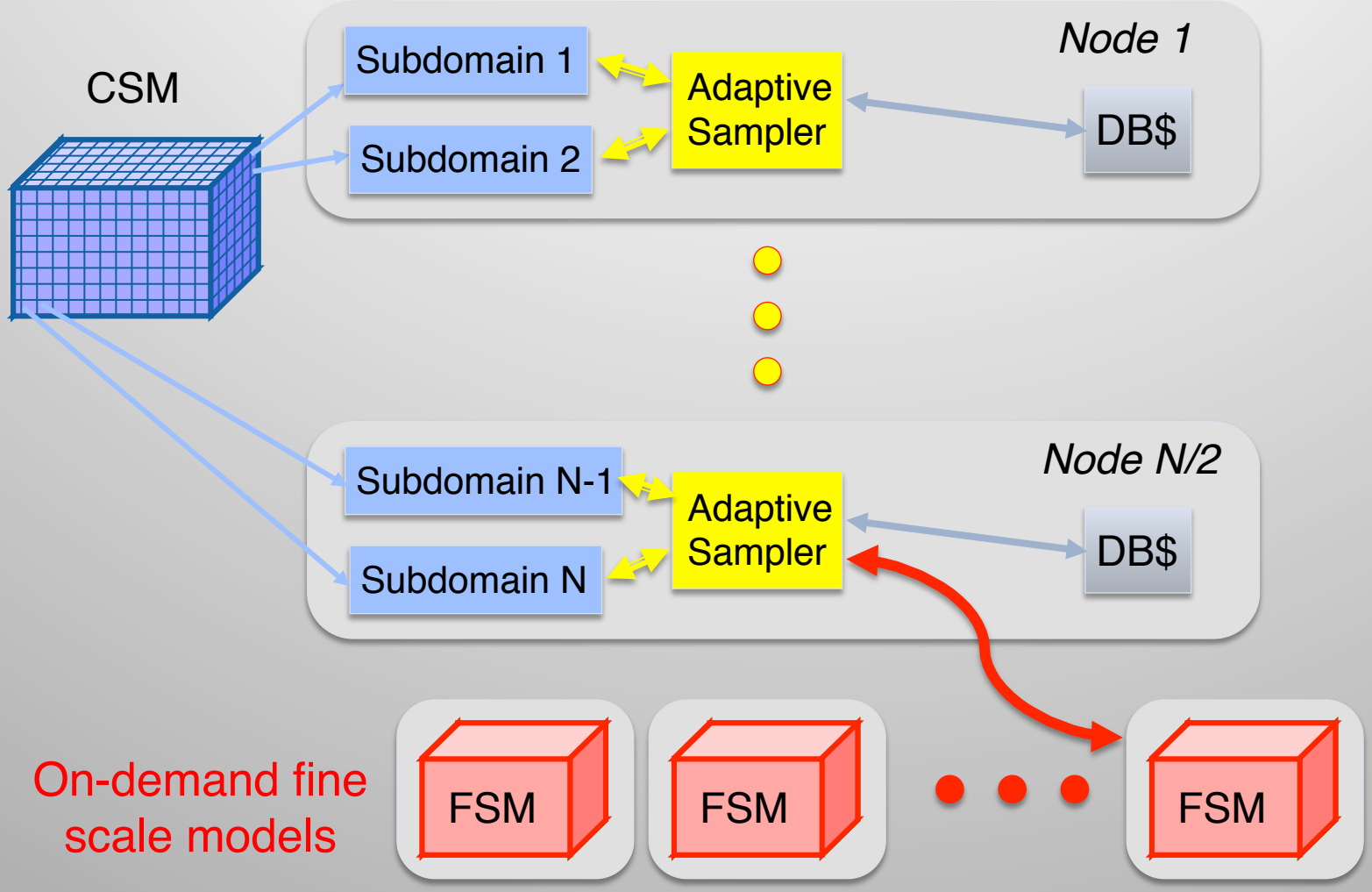
Ref: Barton *et.al*, 'A call to arms for task parallelism in multi-scale materials modeling,' *Int. J. Numer. Meth. Engng* 2011; 86:744–764

Adaptive Sampling builds response on the fly

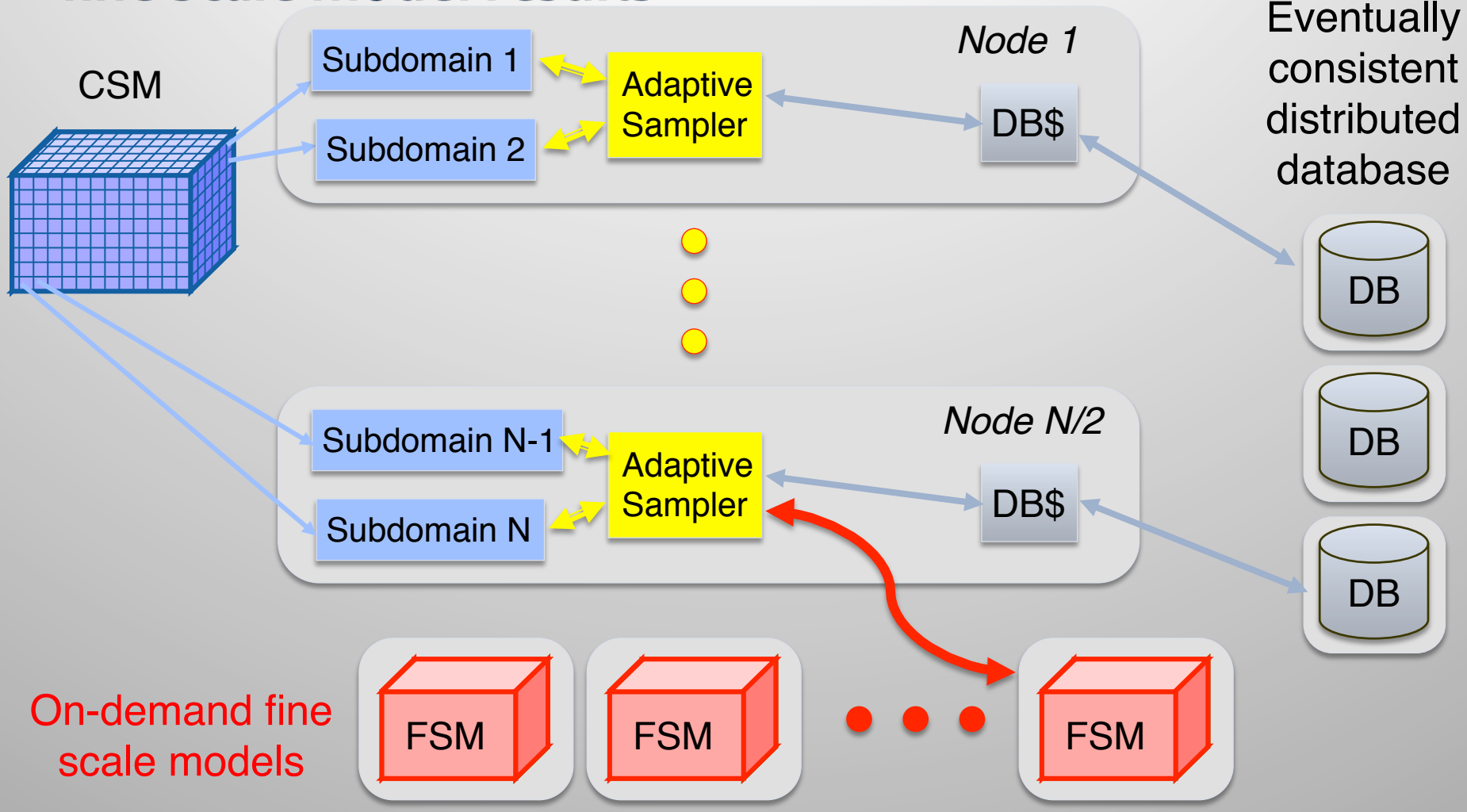
- Coarse scale model queries database for fine-scale material response
- If possible, approximate response from past evaluations
- Otherwise perform fine scale evaluation
- Fine-scale evaluations grow database



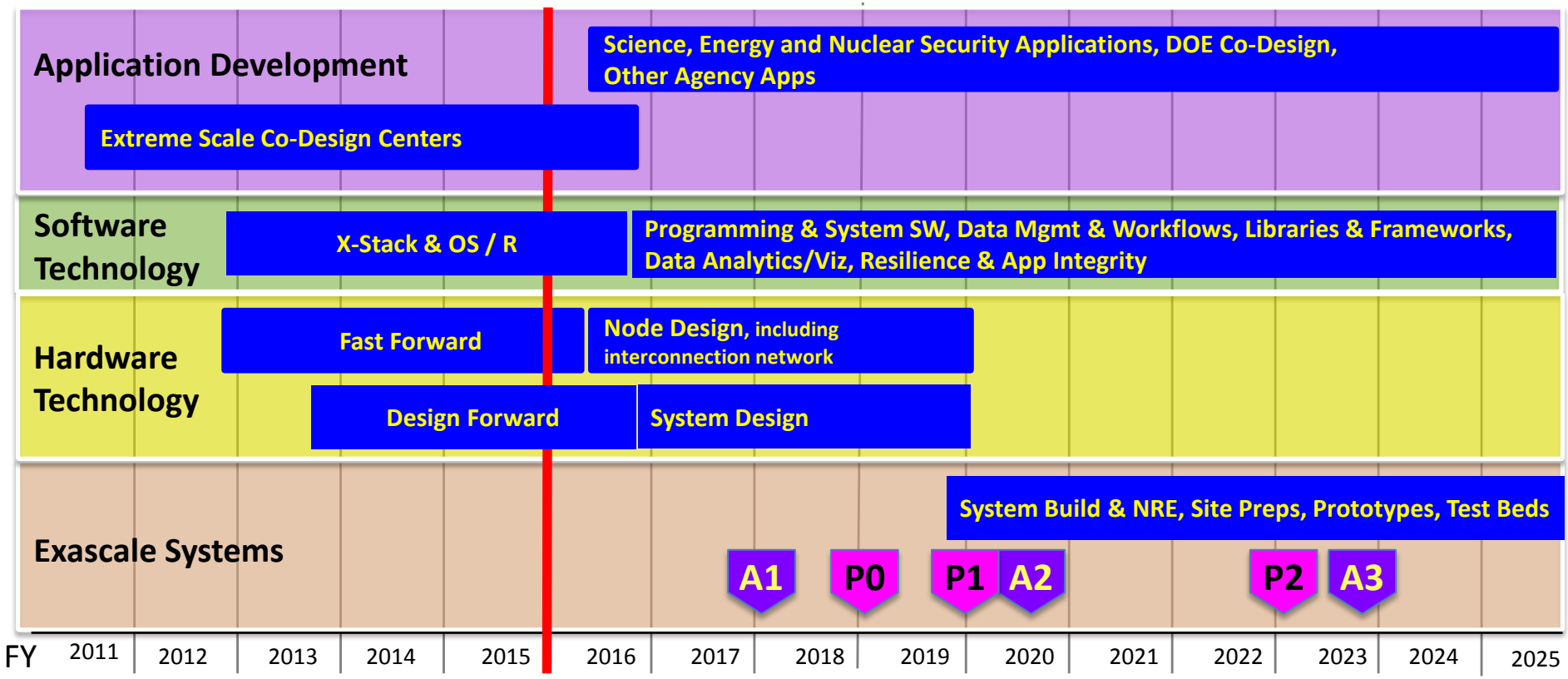
Tradeoff: re-use vs. re-computation of expensive fine-scale model results



Tradeoff: re-use vs. re-computation of expensive fine-scale model results



Steve Binkley, Deputy Director, DOE Office of Advanced Scientific Computing Research



P0 Node Prototype

P1 Petascale Prototype

P2 Exascale Prototype

A1 CORAL Systems

A2 APEX Systems

A3 Exascale Systems

Procurement of A1, A2 and A3 are not executed, but are highly leverage, by the ECI Project