CCMT

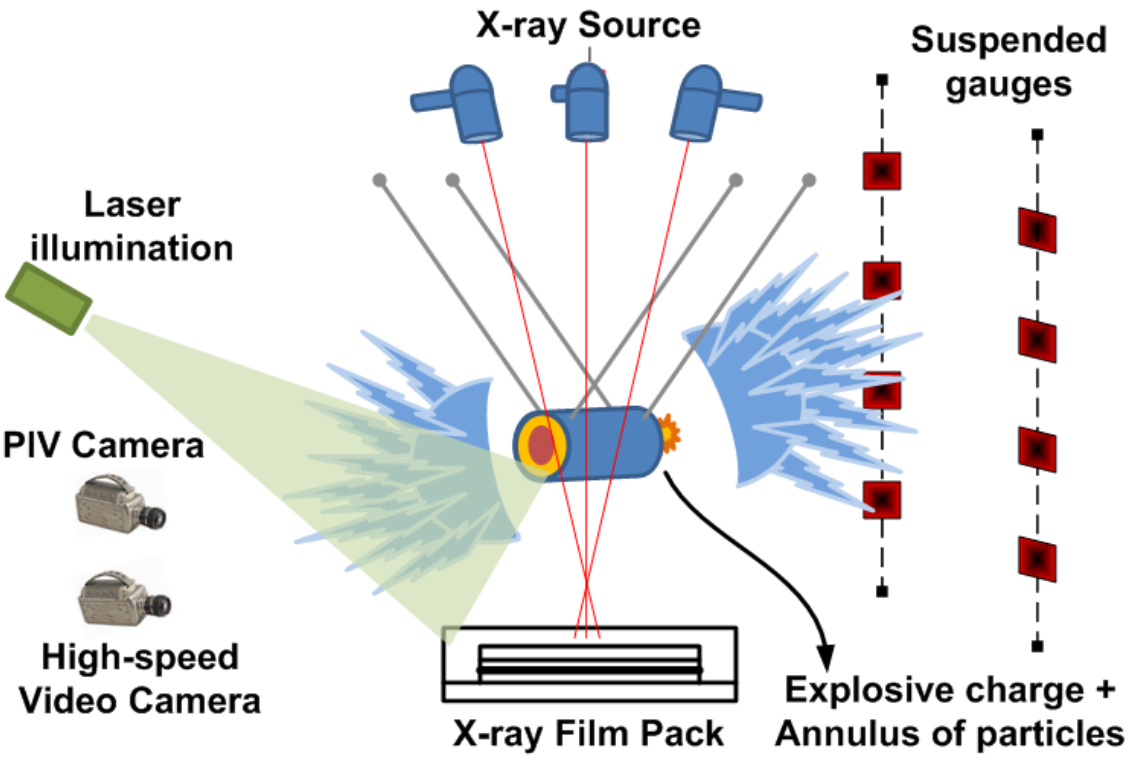
Center For Compressible Multiphase Turbulence

S. Balachandar and H. Lam

Department of Mechanical and Aerospace Engineering

Department of Electrical and Computer Engineering
University of Florida

Demonstration Problem

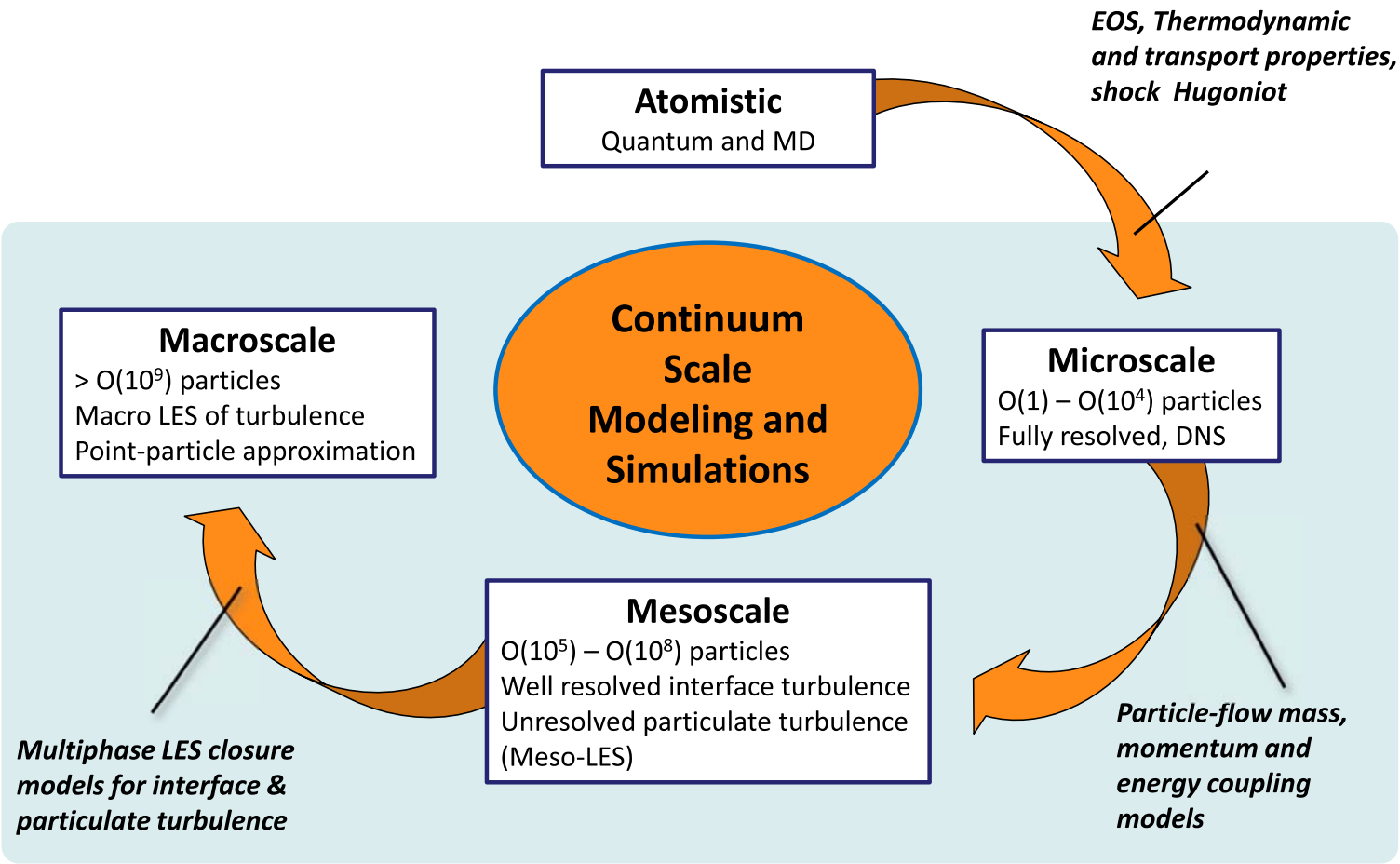


- Integrated simulations
- Experimental measurements for validation

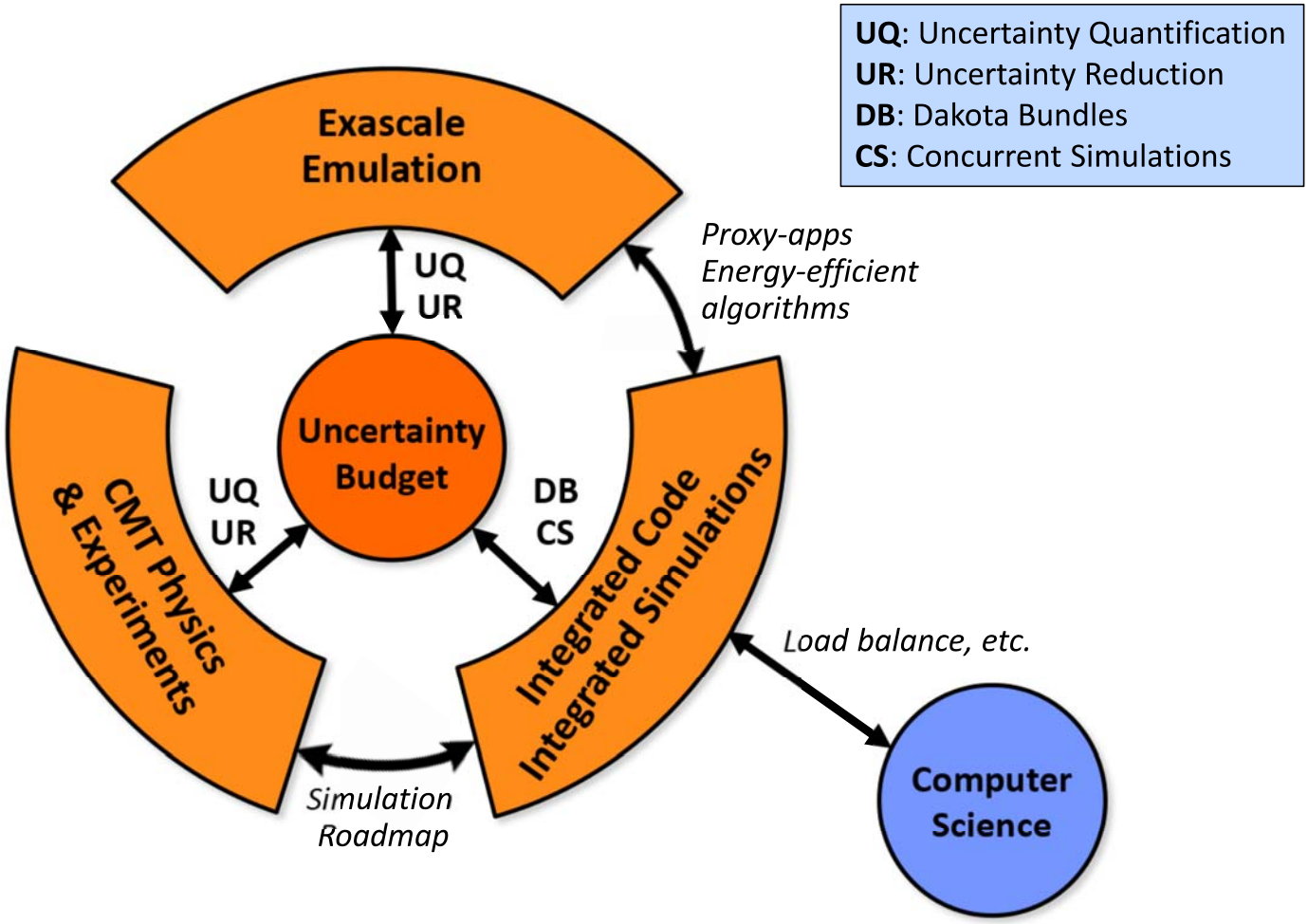
Extreme Multiphase Flow

We desire to perform predictive simulation of these flows with as much multi-scale physics as possible

Multiscale Coupling Strategy



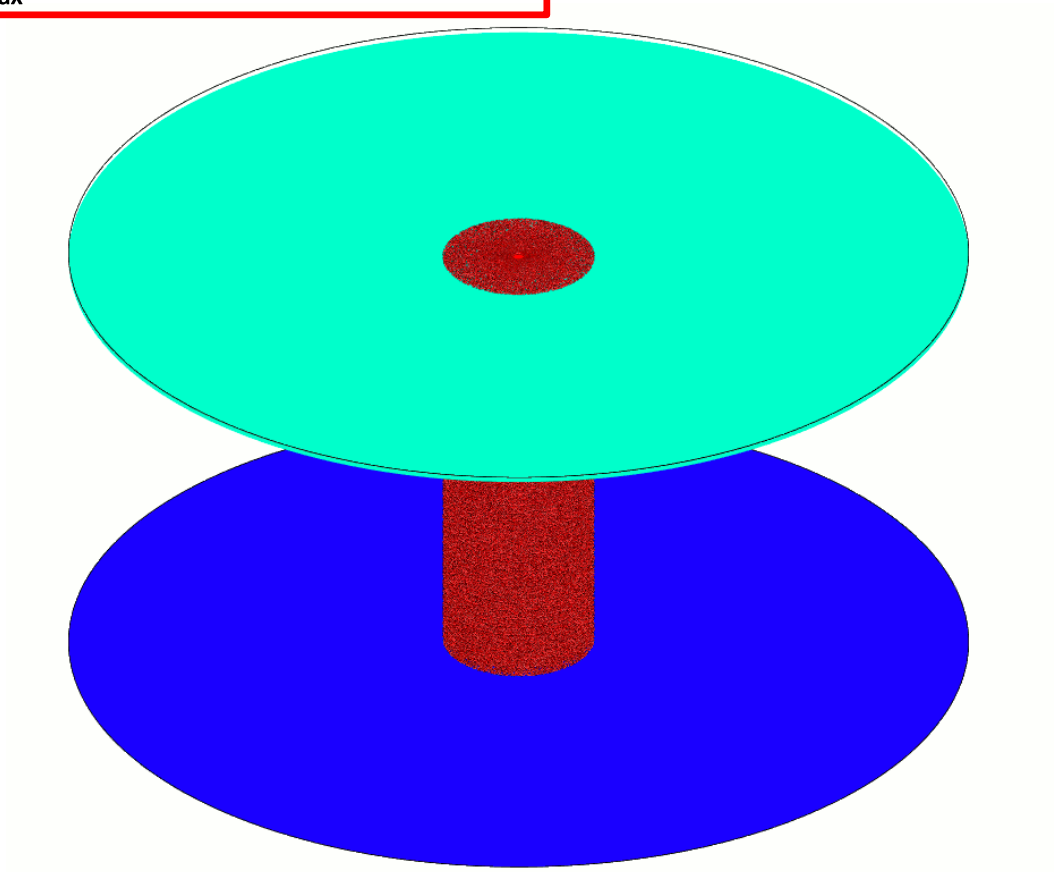
How Different Pieces Fit



Cylindrical Explosive Dispersal of Particles up to 200µs

Features:

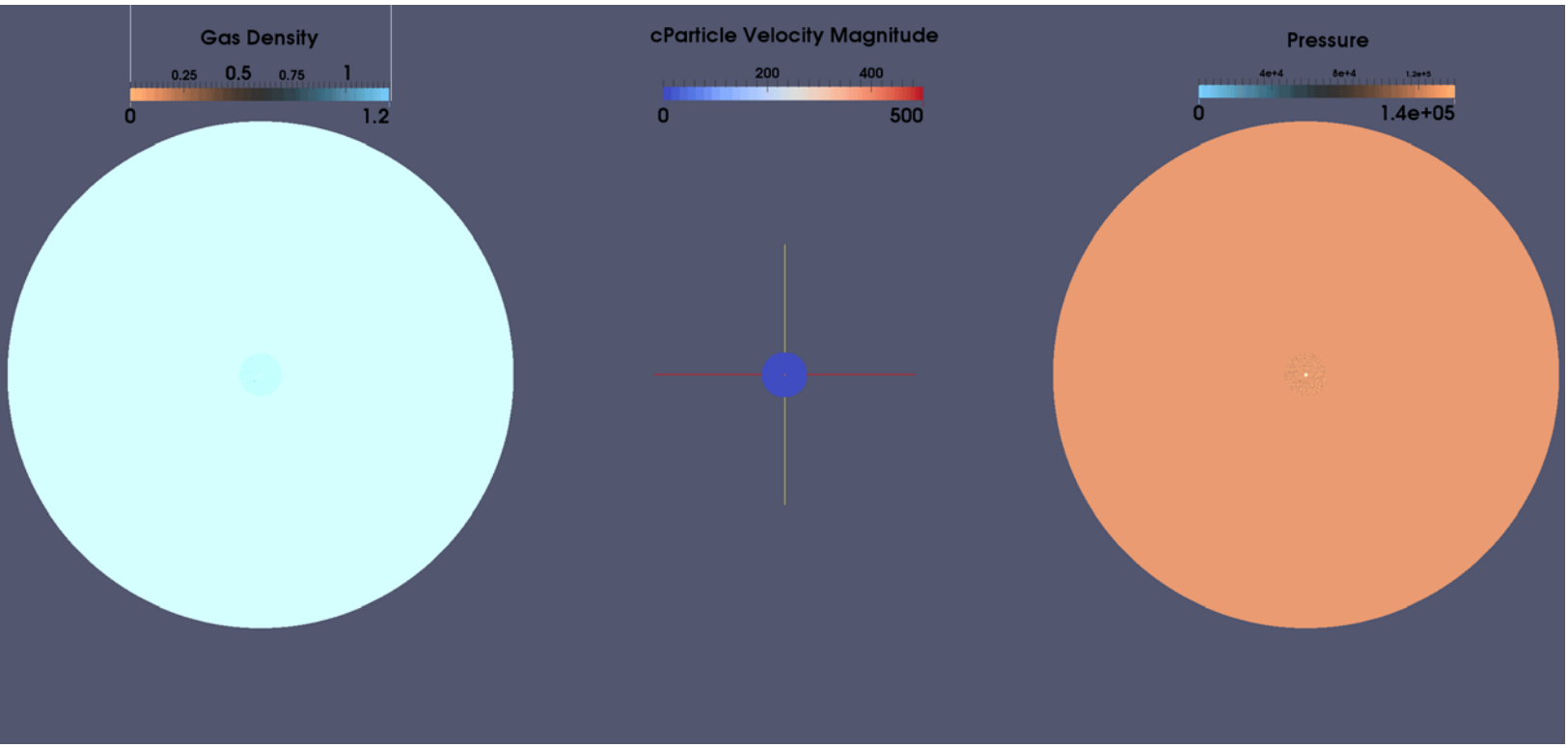
- 30 Millions computational cells
- 5 Millions computational particles
- r_{max} = 0.30cm



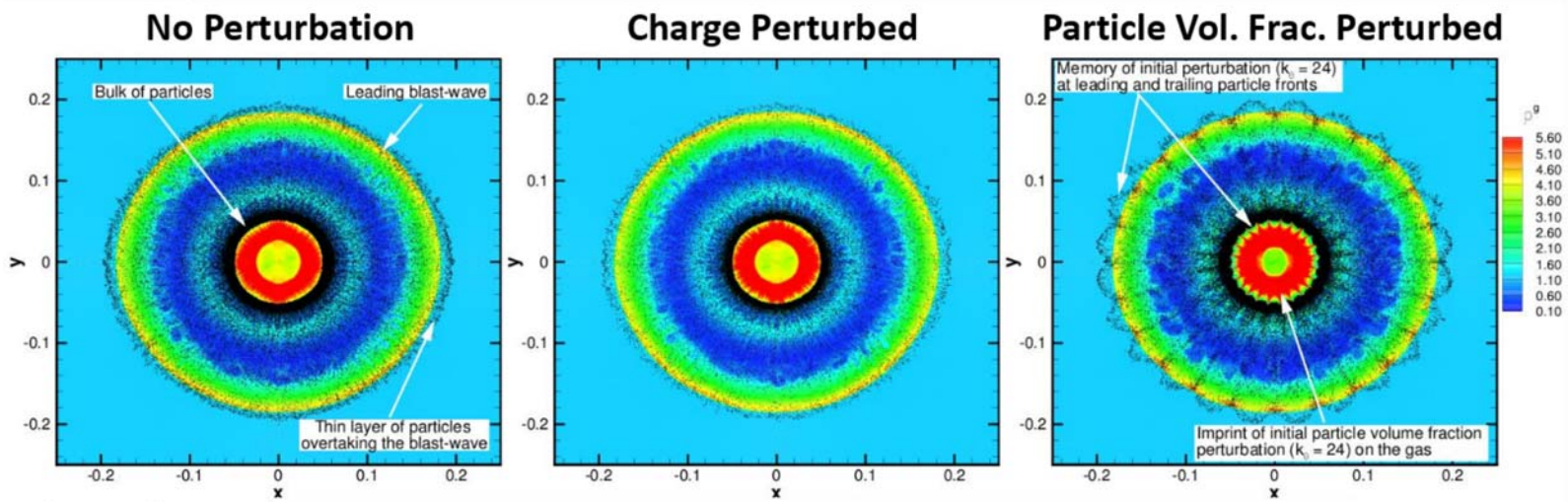
Cylindrical Explosive Dispersal of Particles up to 1ms

Features:

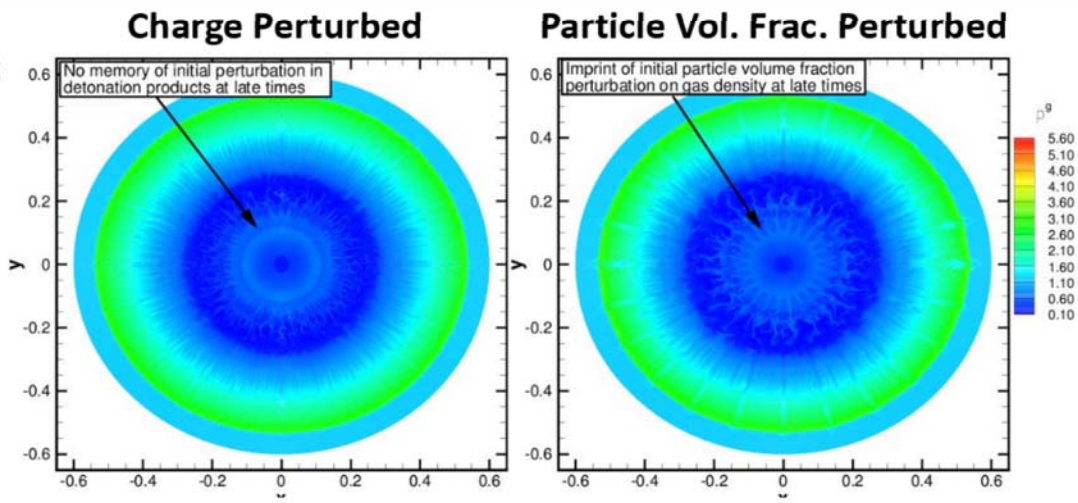
- **2.5 Millions computational cells in a (r,θ) plane**
- **1 Millions computational particles**
- **r_{max} = 0.60cm**



Uncertainty Quantification of Explosive Dispersal



Later time, looking only at The gas phase



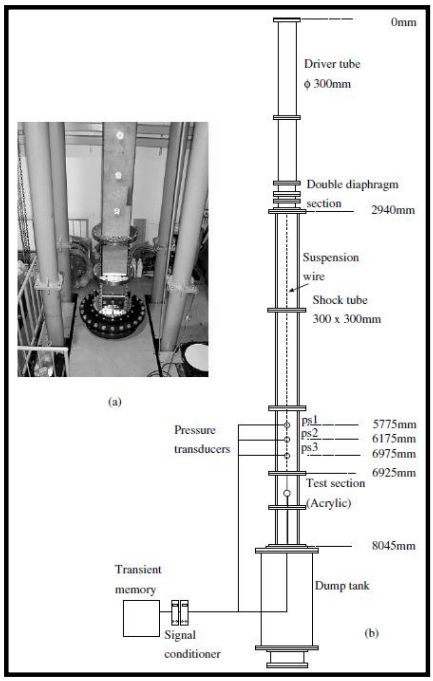
A small perturbation in the charge has no influence on the particle dispersal

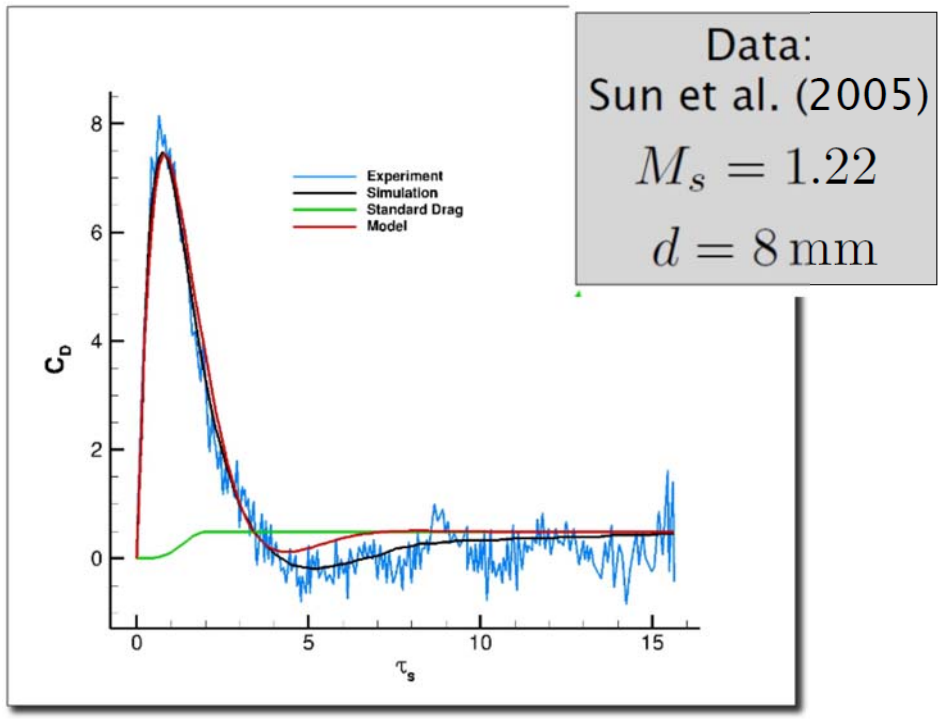
Uncertainty Reduction – Microscale Exp/Sim

❑ Physics – Informed Hybrid Surrogate Modeling

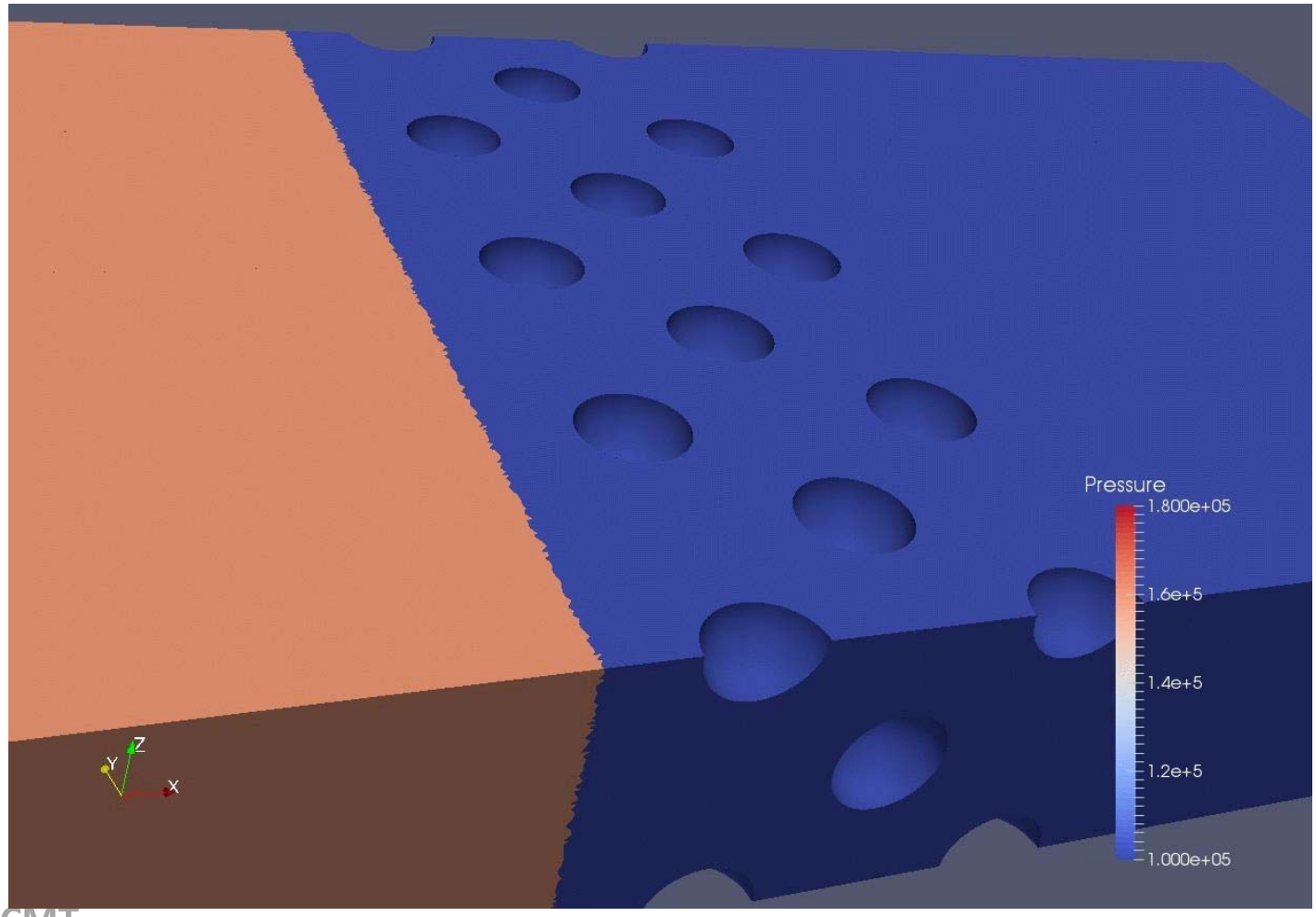
$$F(t) = 6\pi\mu a\left(\overline{u}^S - u^P\right) + V\left(\overline{\rho\frac{Du}{Dt}}^V\right) + V\int_{\xi=-\infty}^{t}\left(K_{IU}\left((t-\xi)\frac{C_0}{a}\right)\left[\overline{\frac{d\rho u}{dt}}^V\right]\frac{C_0}{a}\right)d\xi$$

$$+\int_{\xi=-\infty}^{t}\left(K_{IU}\left((t-\xi)\frac{C_0}{a}\right)\left[\overline{\frac{d^2x\rho}{dt^2}}^V\right]\frac{C_0}{a}\right)d\xi$$



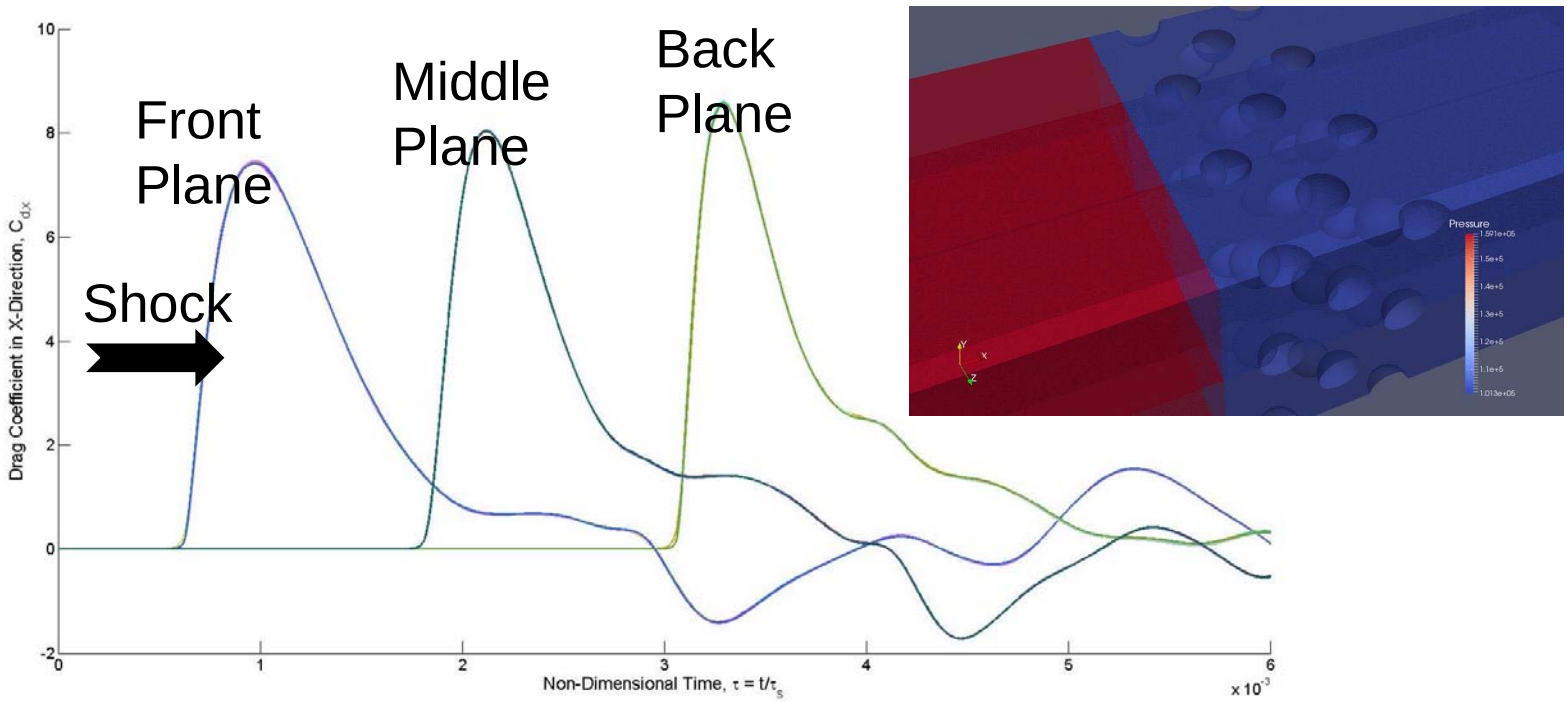


Multi-Particle (FCC) / Shock Interaction



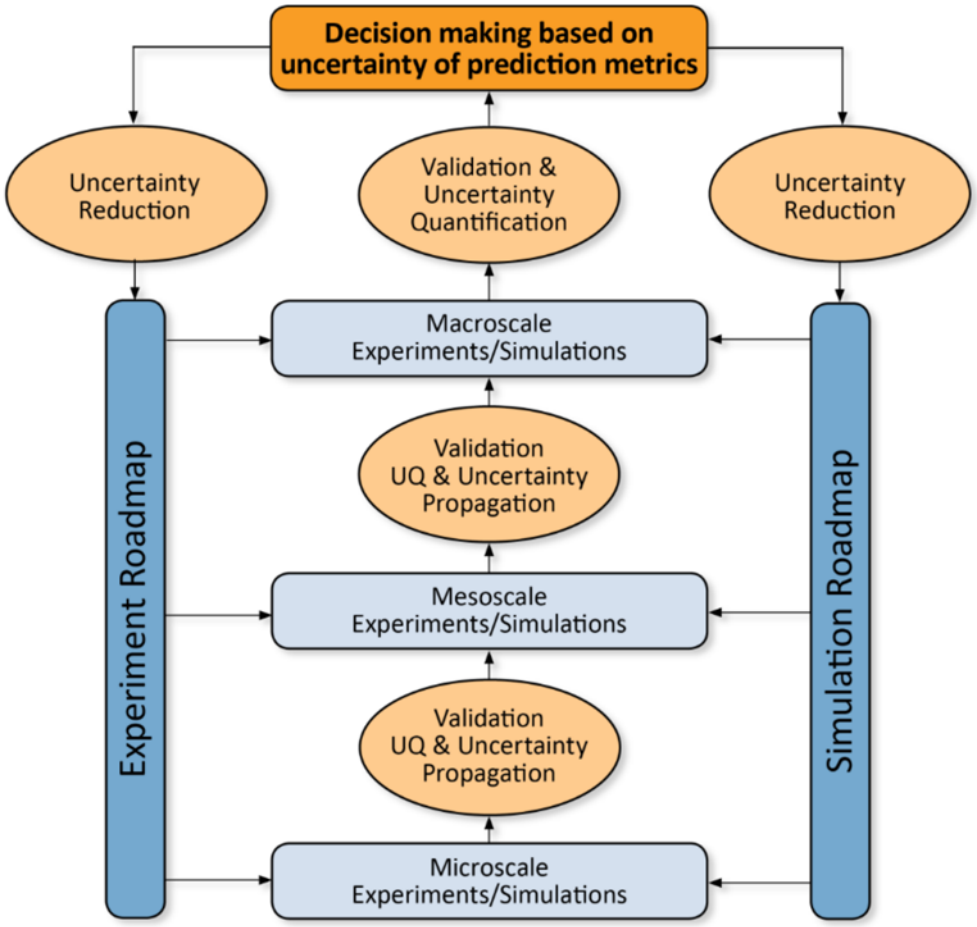
Multi-Particle Force Analysis

- Inviscid flow with a shock Mach number of 1.22
- 10% volume fraction (FCC arrangement)



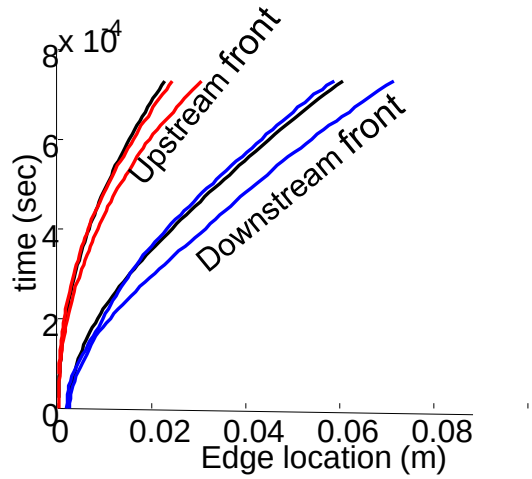
Force histories

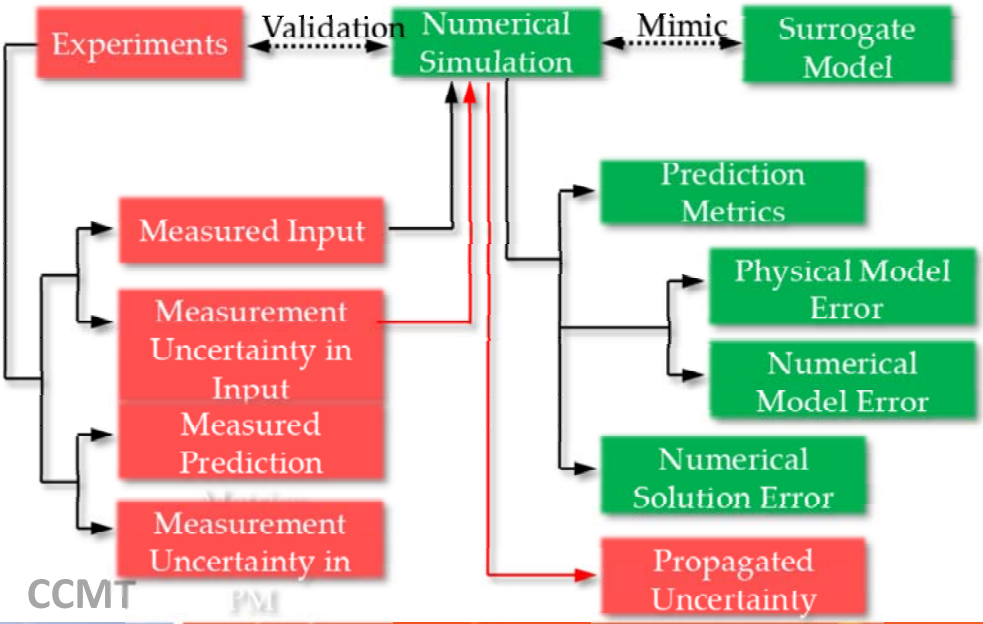
Decision Making with Uncertainty Budget

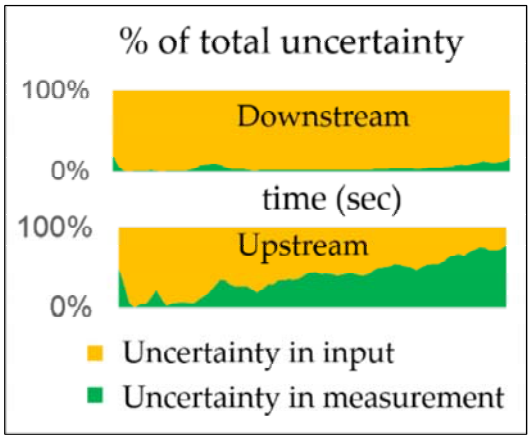


Validation and UQ Framework

Sandia Multiphase Shock Tube (Wagner et al. 2011)





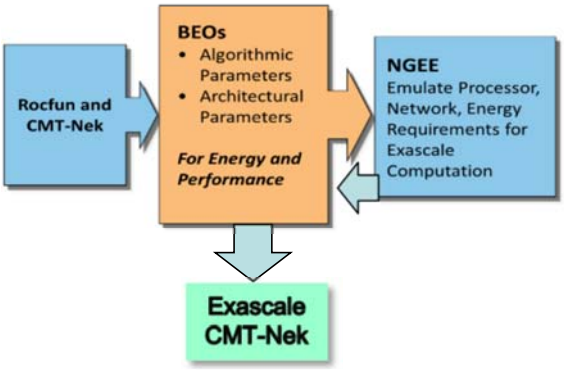


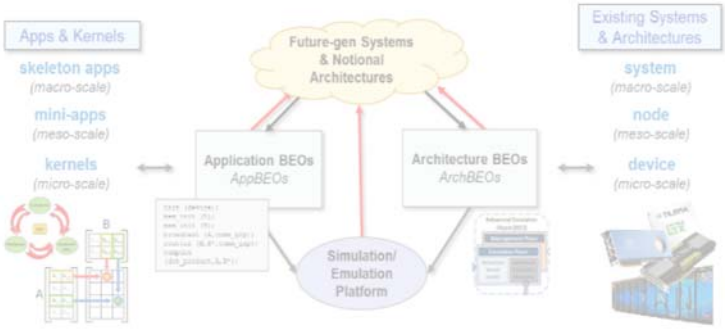
Update

- First-year accomplishments:
  - Fully staffed by April 2014
  - Executed large simulations on several DOE Platforms
  - Dakota bundled runs for UQ
  - UB framework and demonstration for shock-curtain interaction problem
  - Improved physics-based hybrid surrogate models from micro and mesoscale simulations
- Going forward
  - Hero runs of demonstration problem with extreme resolution and additional physics
  - Vigorous uncertainty reduction through micro/meso modeling
  - Emulation-informed CMT-Nek development for near-future architectures

CS Research @ CCMT

- Hardware-software co-design of CMT applications (performance, energy, and thermal issues)





- Behavioral emulation for design-space exploration of CMT applications

Goal & Research Thrusts

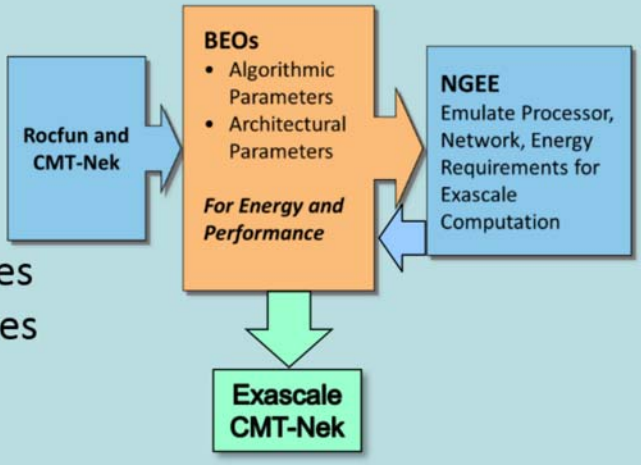
$10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^4$ cores

Goal: Perform multi-objective optimization of CMT apps to provide inputs

- For Exascale CMT app development
- For NGEE Behavioral Emulation: co-design, design-space exploration

Research thrusts

- Multi-objective optimization
  - Energy, performance
  - Based on thermal constraints
- Code generation for hybrid multicores
  - Support for multiple types of cores
  - Support for vectorization
- Load balancing

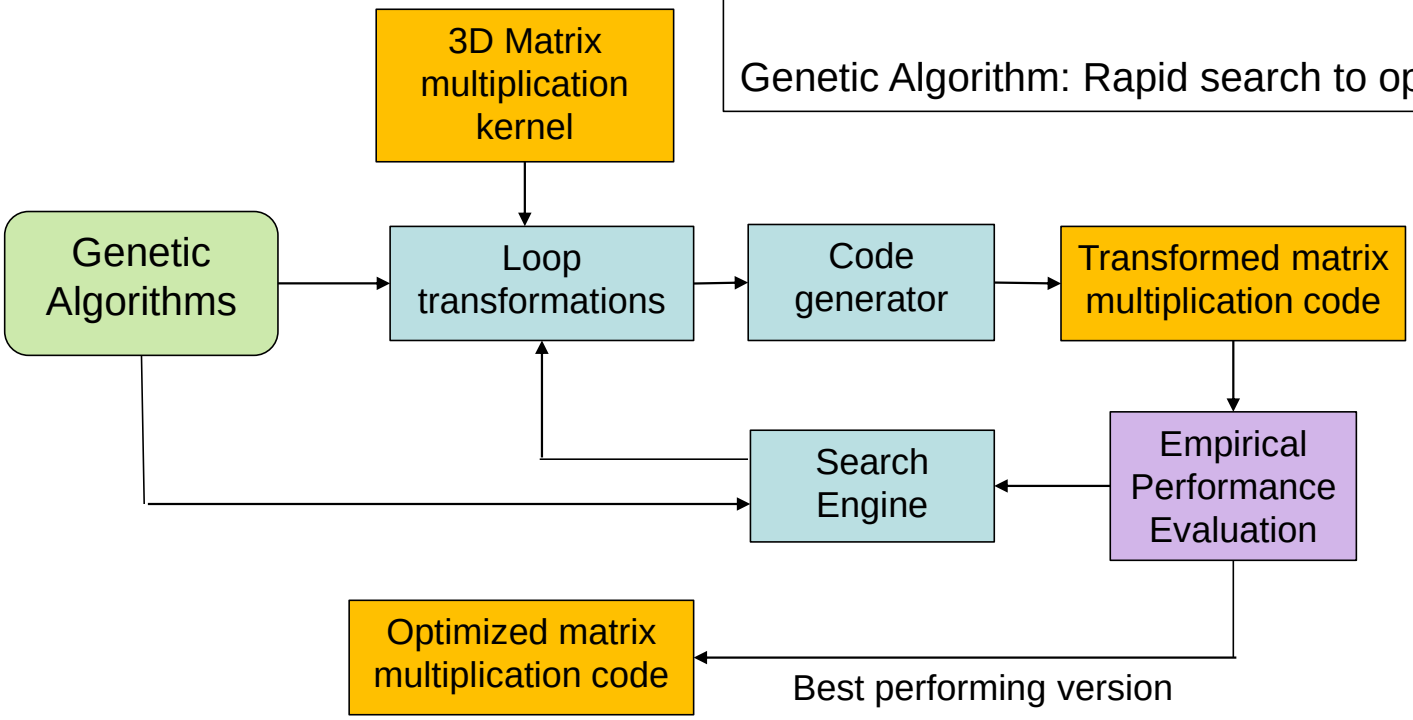


CMT-NEK - Autotuning

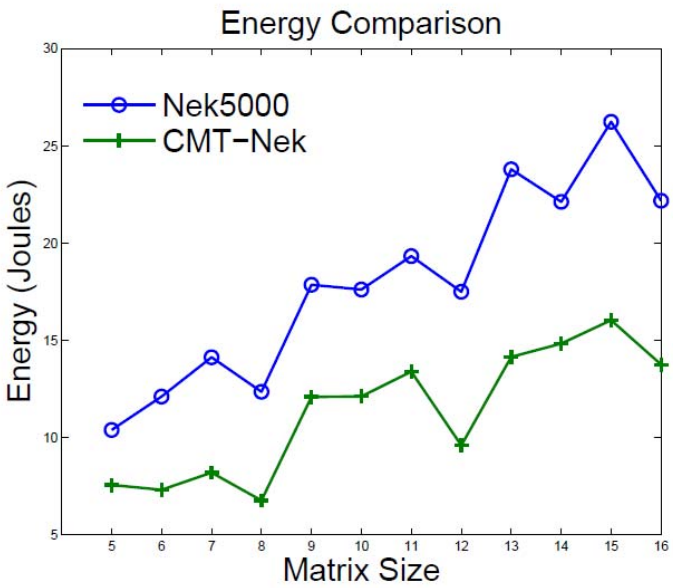
Characteristics of CMT-Nek: Nested loops

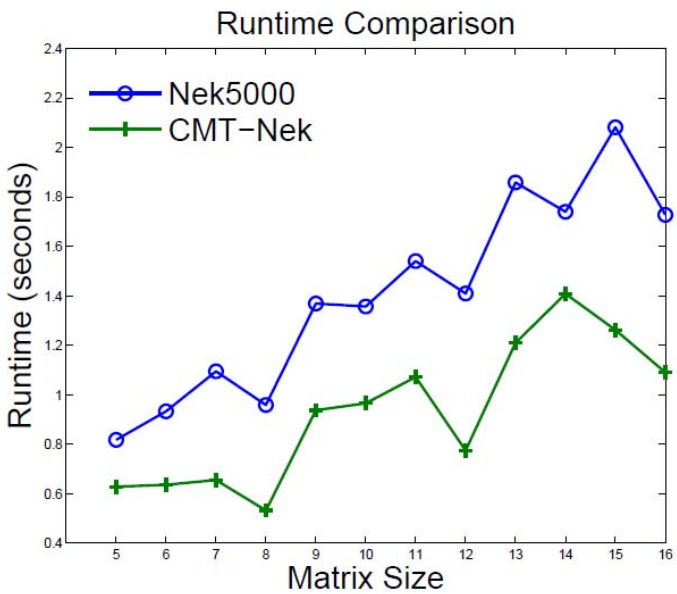
Loop Transformation: CHiLL
- Unroll & Permutations

Genetic Algorithm: Rapid search to optimum



Results For Derivative Operation





- Up to 45% improvement in energy and performance
- Platform: AMD Fusion @ Sandia National Laboratory
- Similar improvement observed on other platforms such as IBM BG/Q and AMD Opteron.

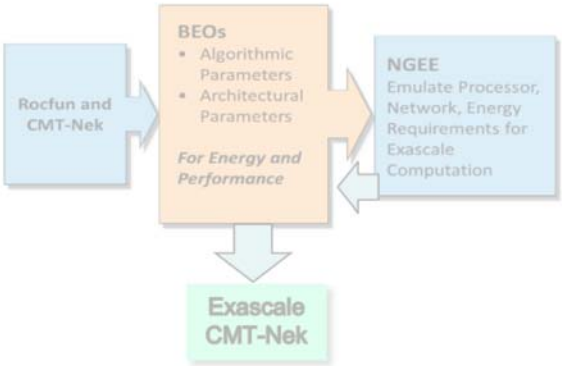
Update: CS Research for CMT-Nek

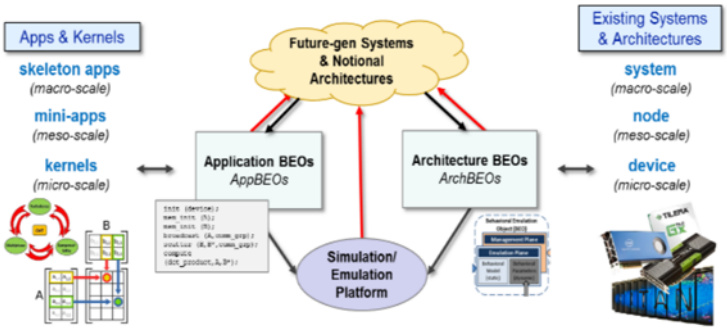
- First-year accomplishments:
  - Autotuning framework based on genetic algorithms
    - 3D matrix multiplication for derivatives and interpolation

- Going forward
  - Autotune other key computational and communicational kernels of CMT-Nek
  - Investigate performance, energy and power models for hybrid multicores
  - Investigate thermal models and constraints

CS Research @ CCMT

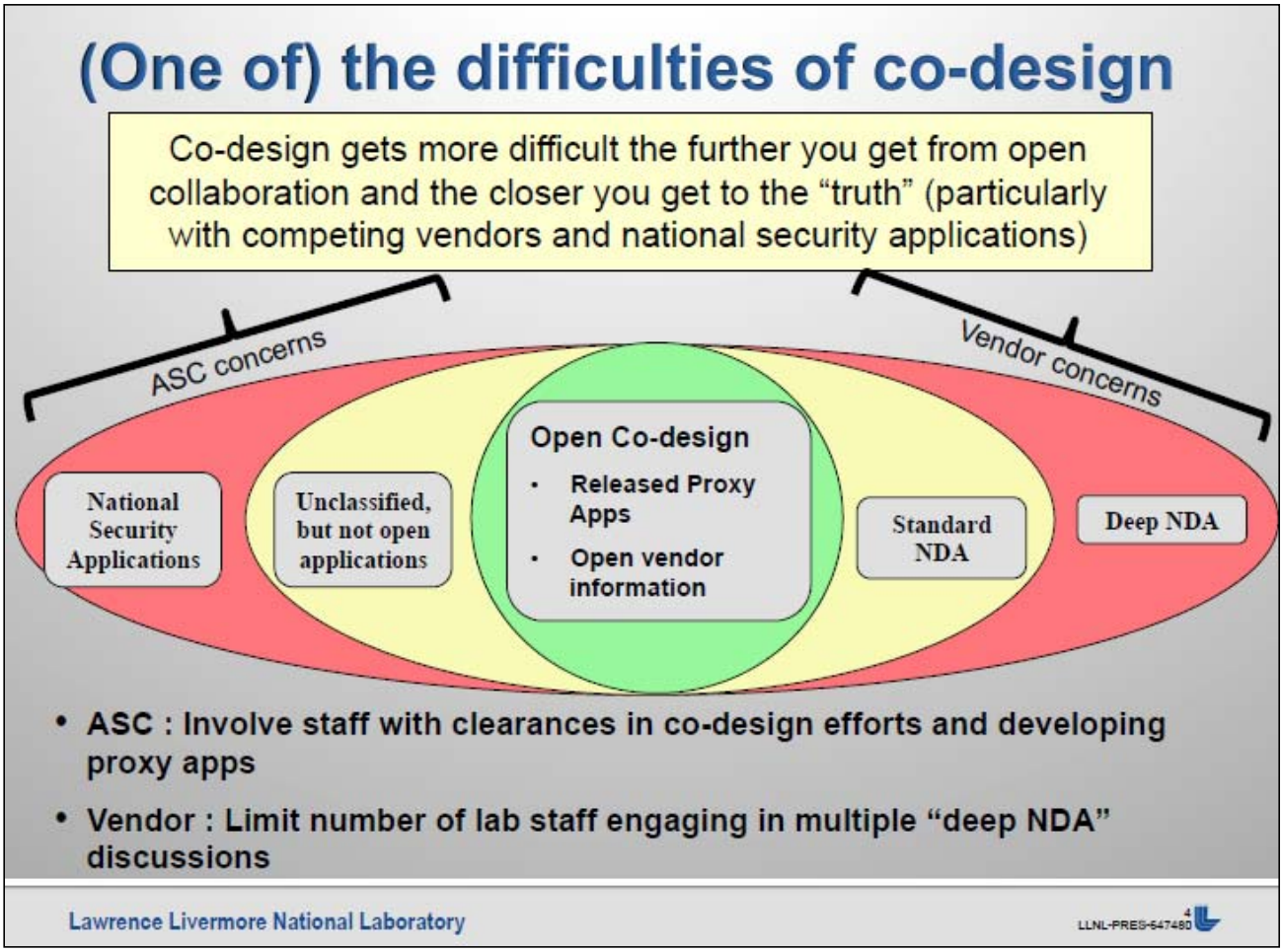
- Hardware-software co-design of CMT applications (performance, energy, and thermal issues)





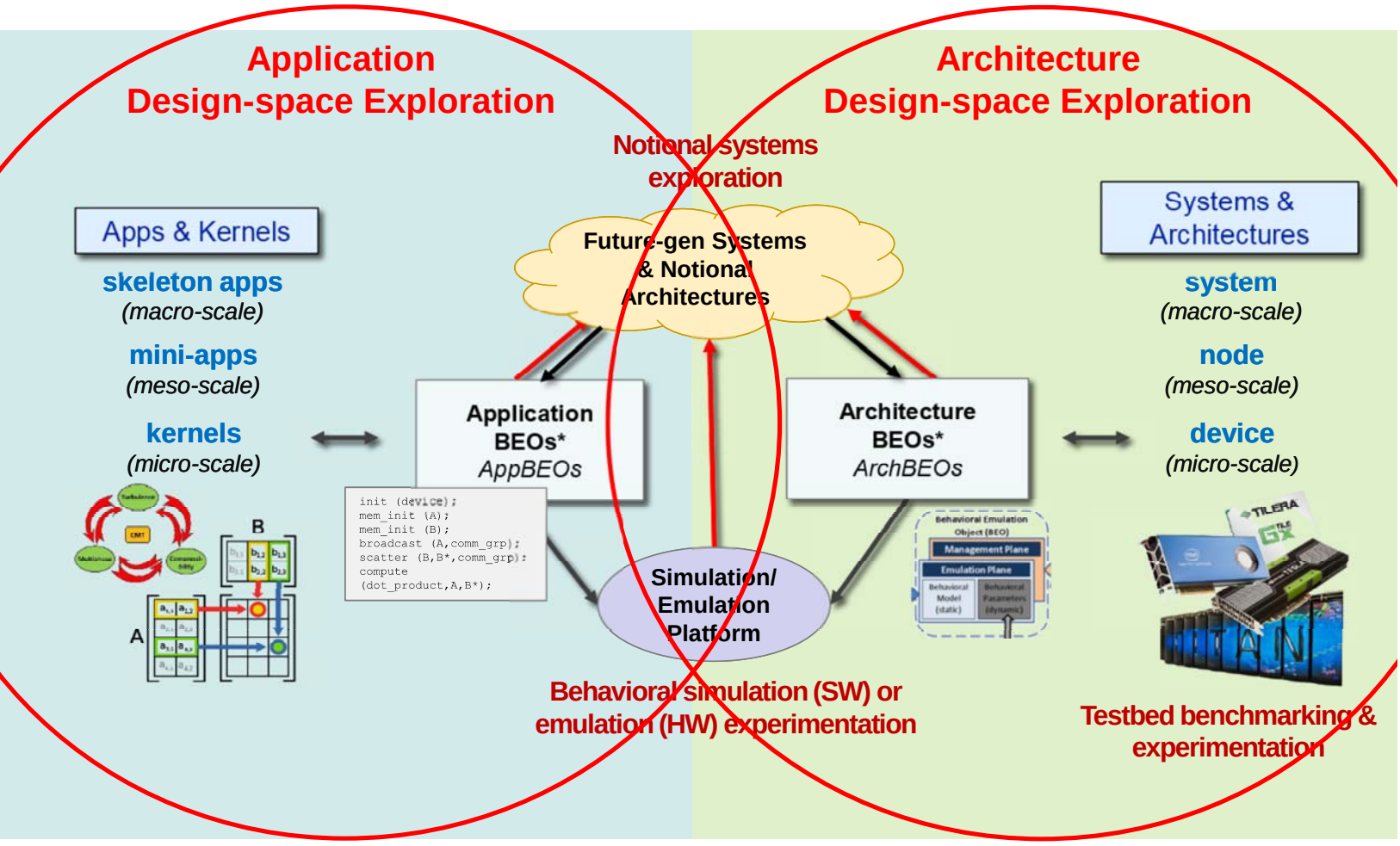
- Behavioral emulation for design-space exploration of CMT applications

Context: DOE Co-design



Bob Neely, "Proxy Applications: Vehicles for Co-design and Collaboration, PSAAP II Kick-off Meeting, Albuquerque, Dec. 10, 2013

CCMT Behavioral Emulation Flow



* BEO – Behavioral Emulation Object

Behavioral Emulation (BE)

- ***Component-based, coarse-grained*** **simulation**
  - Fundamental constructs called BE Objects (BEOs) act as surrogates
  - BEOs characterize & represent behavior of app, device, node, & system objects as fabrics of interconnected ArchBEOs (with AppBEOs) up to Exascale
- ***Multi-scale*** **simulation**
  - Hierarchical method based upon experimentation, abstraction, exploration
- ***Multi-objective*** **simulation**
  - Performance, power, reliability, and other environmental factors

	Apps	Arch	BEO Models
Macro Level	Skeleton-apps	***System*** BEO fabrics	▪ Models abstracted from Meso-scale ▪ Testbed experimentation in support ▪ Notional ***Exascale system*** exploration
Meso Level	Mini-apps	***Node*** BEO fabrics	▪ Models abstracted from Micro-scale ▪ Testbed experimentation in support ▪ Notional ***Exascale node*** exploration
Micro Level	Kernels	***Device*** BEO fabrics	▪ Architectural studies ▪ Testbed experimentation as foundation ▪ Notional ***Exascale device*** exploration

Research Thrusts

BE Modeling Research

1. **Behavioral Characterization**
   - How do we build, calibrate, then validate performance?
2. **Parameter Estimation**
   - How do we efficiently capture behavior in surrogates?
3. **Synchronization & Congestion**
   - How do we handle sync and congestion at scale?
4. **Resilience & Energy**
   - How do we extend beyond performance attributes?

Platform Research

5. **Management & Visualization**
   - How do we measure & analyze massive systems & apps?
6. **Reconfigurable Architectures**
   - How do we exploit FPGA hardware for speed & scale?

Behavioral Emulation Workflow (to date)

Step 1: Calibration

- BEOs: computation & communication
- Performance models
  1. Sample on target platforms for interpolation
  2. Use Kriging method for multi-dim interpolation
  3. Evaluate & recalibrate, if necessary

Step 2: Validation

- Microbenchmarks
  - Computation
  - Communication
- Kernels
  - 2D matrix multiply
  - Sobel filtering
  - 3D matrix multiply
- Platforms
  - Tile-Gx36
  - Xeon Phi

Step 3: Prediction

- Kernels on
  - Next-gen Tile-Gx72
- Kernels on
  - Anticipated Intel Knight's Landing

Behavioral Emulation Platforms

- Software PDES behavioral simulator
  - Initial prototype: In-house developed SMP simulator
  - V2: Leverage existing PDES simulators (e.g., SST, ROSS)
- Hardware-accelerated behavioral emulator
  - FPGA-based reconfigurable computing
  - Leverage emerging reconfigurable supercomputing advances (e.g., UF's Novo-G, Microsoft's Catapult)

Software PDES Simulator: Sample Results

Tile 6x6 (existing sys.)	Simulated Time	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 36 cores	2.82×10^9 ns	-0.35%	35.7us	4.82ms	~135x
Sobel 800x600 image across 36 cores	9.27×10^7 ns	-2.61%	54.2us	8.08ms	~149x

Next-gen

Tile 9x8	Simulated Time	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 72 cores	1.66×10^9 ns	To be determined	81.1us	10.24ms	~126x
Sobel 800x600 image across 72 cores	2.58×10^7 ns	To be determined	102.0us	17.94ms	~176x

Anticipated

KNL 9x8	Simulated Time	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 72 cores	5.87×10^8 ns	To be determined	81.1us	10.24ms	~126x
Sobel 800x600 image across 72 cores	1.37×10^7 ns	To be determined	102.0us	17.94ms	~176x

Single-FPGA Results: 3 Data Points

Tile 6x6 (existing sys)	Simulated Time (Consistent with SMP results)	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 36 cores	2.82×10^9 ns	-0.35%	35.7 us	4.82ms	~135x
Sobel 800x600 image across 36 cores	9.27×10^7 ns	-2.61%	54.2 us	8.08ms	~149x

Next-gen

Tile 9x8	Simulated Time (Consistent with SMP results)	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 72 cores	1.66×10^9 ns	To be determined	81.1us	10.24ms	~126x
Sobel 800x600 image across 72 cores	2.58×10^7 ns	To be determined	102.0us	17.94ms	~176x

Anticipated

KNL 9x8	Simulated Time (Consistent with SMP results)	Prediction Error	FPGA Simulation Time	SMP Simulation Time	Speedup
2D MM 1024x1024 across 72 cores	5.87×10^8 ns	To be determined	81.1us	10.24ms	~126x
Sobel 800x600 image across 72 cores	1.37×10^7 ns	To be determined	102.0us	17.94ms	~176x

Update: BE Modeling Research

- First-year accomplishments:
  - Demonstrated successful device-level calibration, validation, & prediction
  - On existing (Xeon Phi, Tilera) & notional devices
- Going forward
  - Extend methodology beyond device level (node, system)
  - Abstraction, scalable synchronization and congestion issues

❖ **CMT-centric**

- **CMT kernel, proxy apps (Nekbone)**
- **Questions to be answered** for application design-space exploration ("knobs" for tunable design parameters)

Update: Platform Research

- First-year accomplishments:
  - Software PDES behavioral simulator
    - Proof of concept prototype: in-house developed SMP simulator
  - Hardware-accelerated behavioral emulator
    - Single-FPGA prototype: feasibility study with promising results
- Going forward
  - Software PDES behavioral simulator
    - V2: Leverage existing PDES simulators (e.g., SST, ROSS)
  - Hardware-accelerated behavioral emulator
    - Extend to multiple FPGAs on Novo-G#
    - Leverage emerging reconfigurable supercomputing advances (e.g., IBM's CAPI coherent accelerator interface, Microsoft's Catapult, Micron's HMC)

Questions?