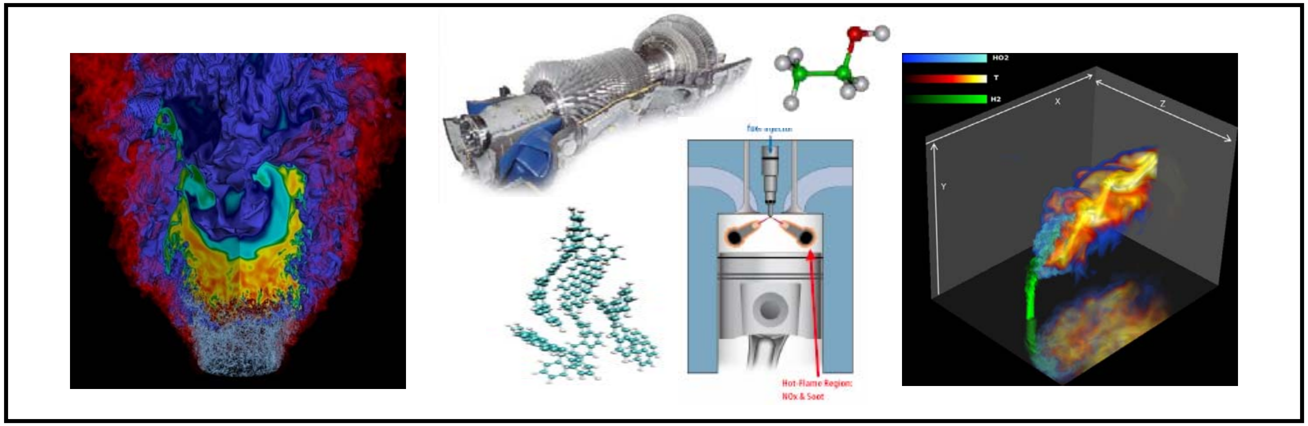
Combustion Exascale Co-Design Center

Center Director: Jacqueline Chen (SNL)
Deputy Director: John Bell (LBNL)

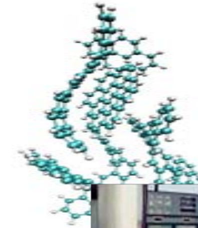


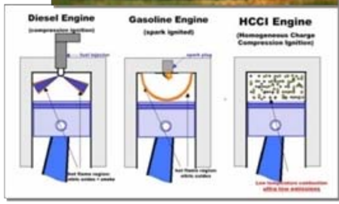
Members and Team Leads (bold): Janine Bennett (SNL), **Curtis Janssen (SNL)**, Arun Rodrigues (SNL), Omar Ghattas (UT Austin), **Robert Moser (UT Austin),** Valerio Pascucci (Utah), **Patrick McCormick (LANL),** Allen McPherson (LANL), Patrick Hanrahan (Stanford), Alexander Aiken (Stanford), Marc Day (LBNL), Michael Lijewski (LBNL), Paul Hargrove (LBNL), **John Shalf (LBNL),** David Donofrio (LBNL), Erich Strohmaier (LBNL), Samuel Williams (LBNL), Robert Falgout (LLNL), Ulrike Meier Yang (LLNL), **Daniel Quinlan(LLNL),** Ray Grout (NREL), **Scott Klasky (ORNL),** Karsten Schwan (Georgia Tech), Manish Parashar (Rutgers)

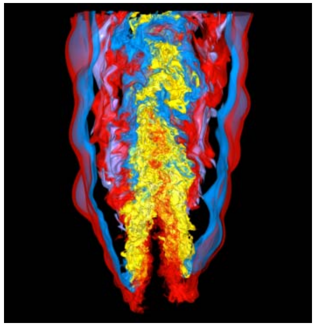
6th International Exascale Software Project Workshop
San Francisco, CA
April 6-7, 2011

Co-Design Will Enable Predictive Simulation and Modeling of Combustion with 21st Century Fuels

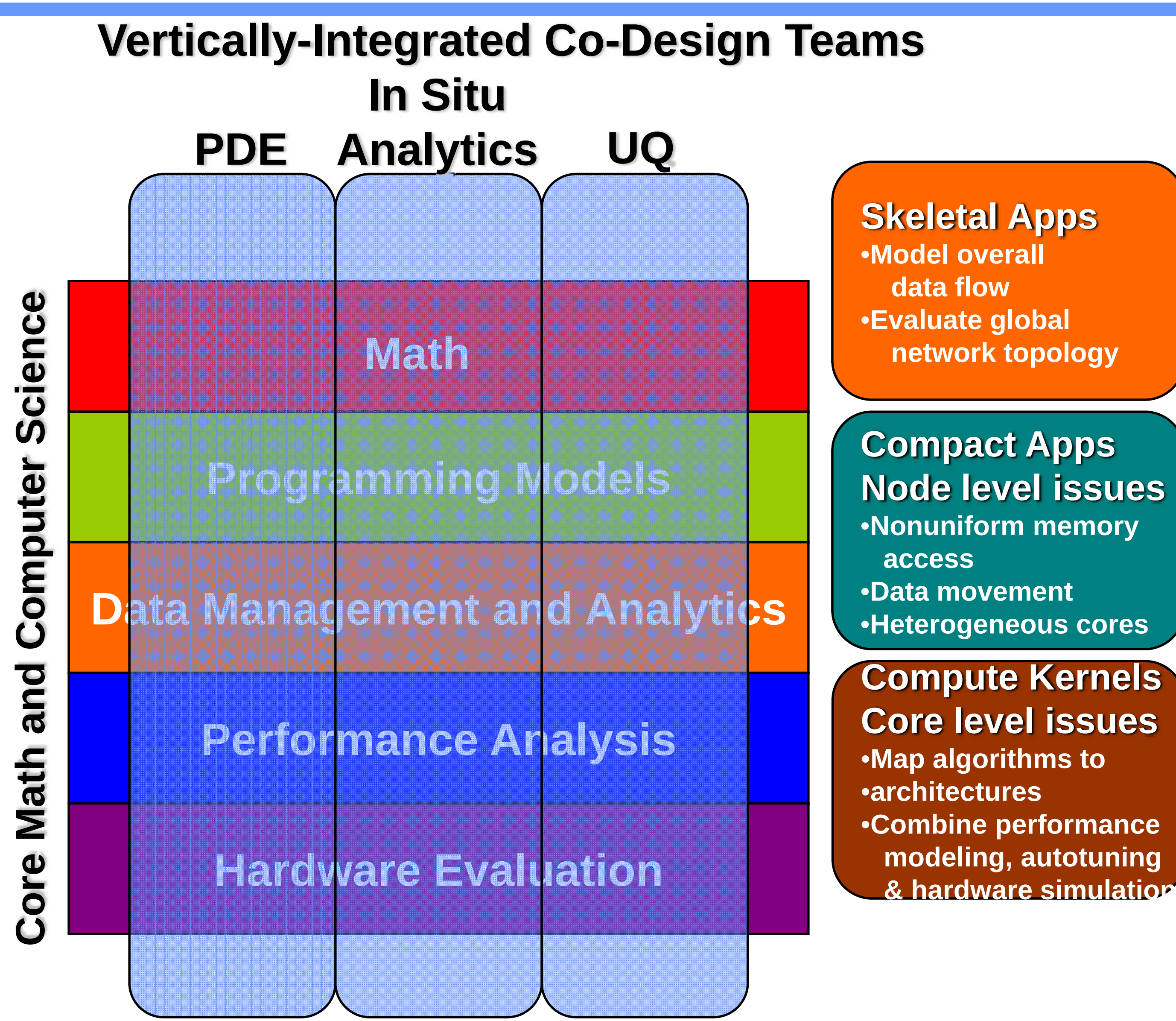
- **Motivation:**
  - Aggressive federal mandates to reduce greenhouse gas emissions by 80 percent by 2050 and petroleum consumption by 25 percent by 2020
- **Goals:**
  - Develop an exascale combustion modeling capability that combines high-fidelity direct numerical simulation, *in situ* analytics and embedded uncertainty quantification
  - Develop the necessary computer science tools and applied math methodology to facilitate the design and implementation of these applications
  - Quantify hardware constraints for an effective system
- **Impact:**
  - Simulations that differentiate alternative fuel effects on turbulence-chemistry interactions in high pressure regimes of practical combustors
  - Simulations will provide fundamental insight and validation data to guide development of predictive models used by industry to shorten the design cycle, promote economic competetiveness, reduce foreign oil dependence, and promote environmental stewardship
  - Insights will enable co-design of 21st Century fuels and fuel-flexible, efficient, low emissions combustion systems for transportation, power generation, and industrial processes







Co-Design Strategy



Requirements, Research, and Collaboration

Requirements:

PDE methodology for direct numerical simulation

- Require AMR to meet spatial resolution requirements
- Must support both low Mach number and fully compressible formulations

***In situ* data analytics**

- Data rates too high for deferred analysis – emphasis on data reduction and steering
- Support for data layout, volume and particle visualization, topological feature tracking, pathlines, local flame coordinates, feature-base statistics

Embedded uncertainty quantification (UQ)

- Impact of uncertainty of chemical parameters on predictive capability
- Quantitative comparisons with experimental data

Research issues and opportunities for collaboration

- How can programming models be used to: exploit fine-grained parallelism in PDE algorithms, express data movement vs. floating point operations in designing numerical algorithms, expose issues of fault tolerance and energy use?
- What is the most effective strategy for *in situ* data analysis? Shared work on nodes or staging? Use scratchpad memory for analysis? How do we balance simulation with analysis? What is the optimal data structure?
- How can we best formulate UQ problems for complex multiphysics problems with "chaotic" dynamics such as turbulent combustion? Hardware support?

Starting from Petascale Combustion Codes: LMC

- Adaptive Mesh Refinement code
  - John Bell’s group at LBNL
  - ~750,000 lines of code
  - C++/F90
    - Runs on Crays, IBM Powers, clusters – failed on BG/L
  - MPI, OpenMP (mostly for performance)
    - MPI is hidden, OpenMP is not
  - Use (mostly) their own libraries (BoxLib, AmrLib)
  - Used up to 50,000 cores (could go higher)
    - Strategy: Problem size -> Memory -> # of nodes
  - Up to ~60TB output right now

Combustion: LMC

- AMR code with:
  - Structured Grids
  - Adaptive Multigrid
  - Sparse linear algebra (iterative methods)
  - Local ODEs ((mostly serial)
    - Solve for Chemistry
    - Highly variable load depending on physical conditions
      - load balancing issue

Combustion: LMC

- Dynamic list of structured grids of various sizes
  - Limited by architectures and math requirements
  - Physical locality is not necessarily preserved between different sub-grids
  - Potential data affinity and communication issue
- Data layouts with space filling curves
- AMR has inherent, dynamic load-balancing problem
  - Thinking about multi-level strategies to match future architectures – inter-node and intra-node.

Combustion: S3D

- Cartesian structure mesh compressible N-S solver for reacting flows (high order 9 pt stencils, explicit time integration)
  - Jackie Chen's group at SNL
  - ~250,000 lines of code
  - F90/F77
    - Runs on CrayXT5, CrayXE6, IBM BG/L, clusters
  - MPI code scales to 250,000 cores,
  - MPI+OpenMP and MPI+CUDA (OLCF-3 CAAR program, hybrid 2012)
  - Use (mostly) their own libraries (ACML)
  - Used up to 250,000 cores (could go higher), production runs on 144,000 cores
    - Strategy: Problem size -> Memory -> # of nodes
  - Up to ~3/4 petabyte output/run 50M cpu-hr on Jaguar
  - No out of core (100 MB), small memory usage
  - No global synchronization needed (monitoring and async I/O)
  - In-situ visualization and topological feature extraction
  - MPI-IO and ADIOS for I/O and data staging
  - Want to explore DSLs for hiding low level constructs for parallelism

Math

•Examine basic PDE discretization approaches from the perspective of minimizing memory requirements and data movement (managing power) instead of optimizing floating point efficiency

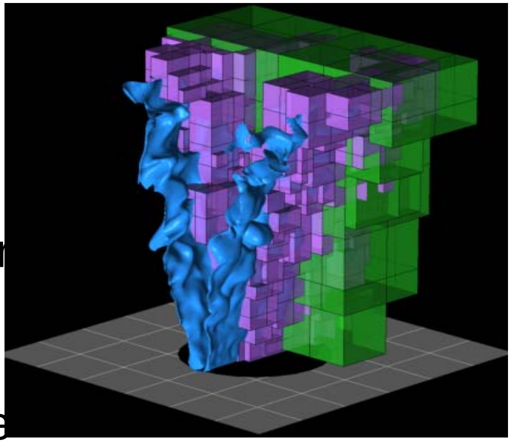
•Explore algorithms and hardware support for structured linear solvers

•Explore algorithms for verification and UQ including hardware acceleration (adjoints in transient problems, PC expansions)

•Investigate hierarchical scalable approaches to AMR that minimize global communication and data movement

•Investigate alternative programming models that facilitate expressing parallelism and controlling data layout in structured grid discretizations

•Explore the effect of architecture variation on performance of stencil operation kernels using a combination of modeling, autotuning and hardware simulations



Multi-level hierarchy of block-structured state data

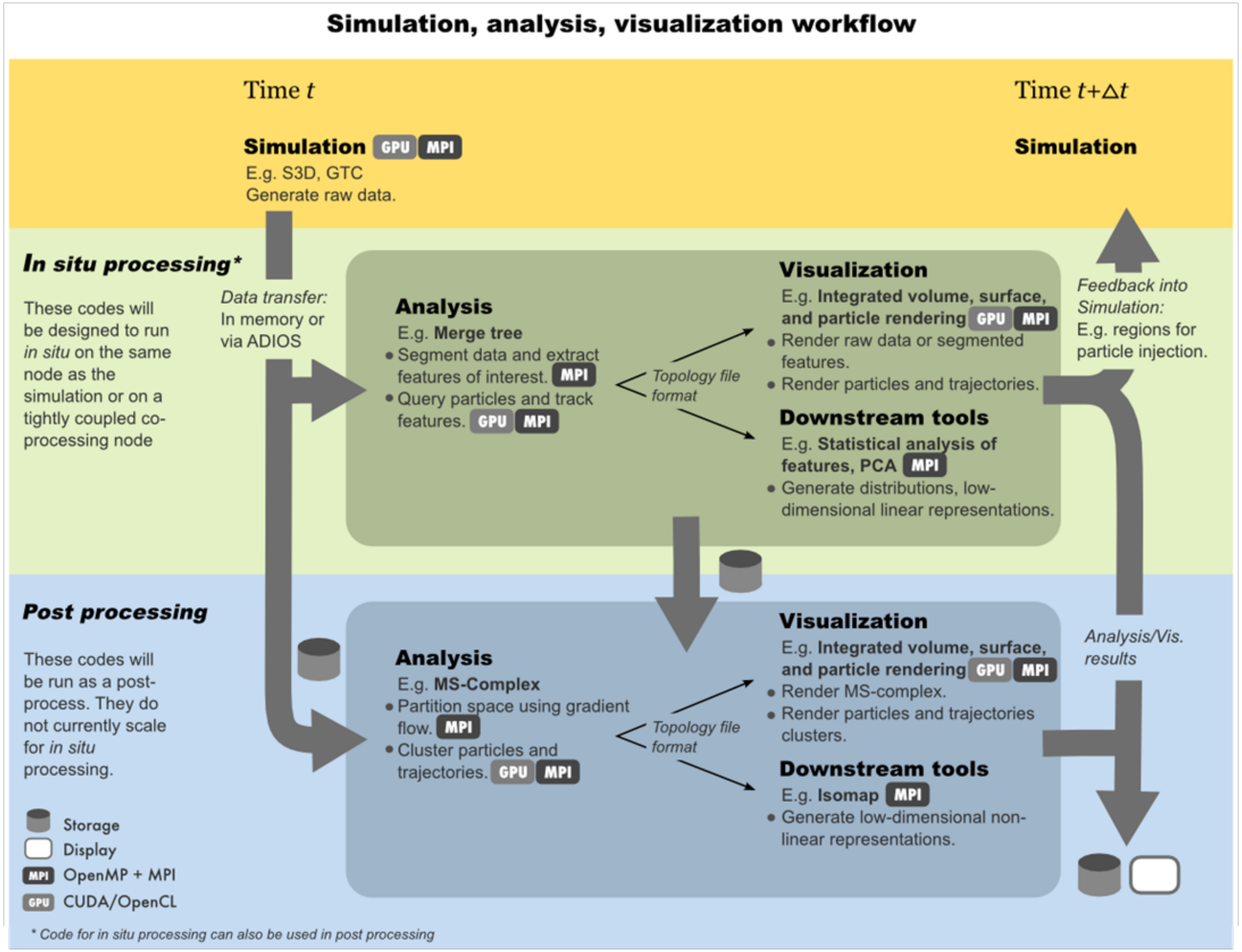
Programming Models

- Develop set of combustion-centric abstractions to unify simulation, in situ UQ and analytics/viz
  - Multi-level APIs for different levels of use: proof-of-concept, performance-focused development, and DSL-compiler driven optimizations
- Evaluate programming models and develop extensions to PM to map key algorithms to exascale hardware –
  - initial efforts support PM that address both distributed and local memory architectures (MPI + OpenMP + GPU) for encapsulating combustion domain
  - later extensions to PM for restricted forms of cache coherency (i.e. support forms of parallelism requiring data sharing)
- Design domain specific languages (DSL’s) and compiler support tailored to combustion needs along with technologies required for DSL’s to be developed - embedded DSL compilers and run time systems that support scheduling and interoperability

Data Management and Analytics

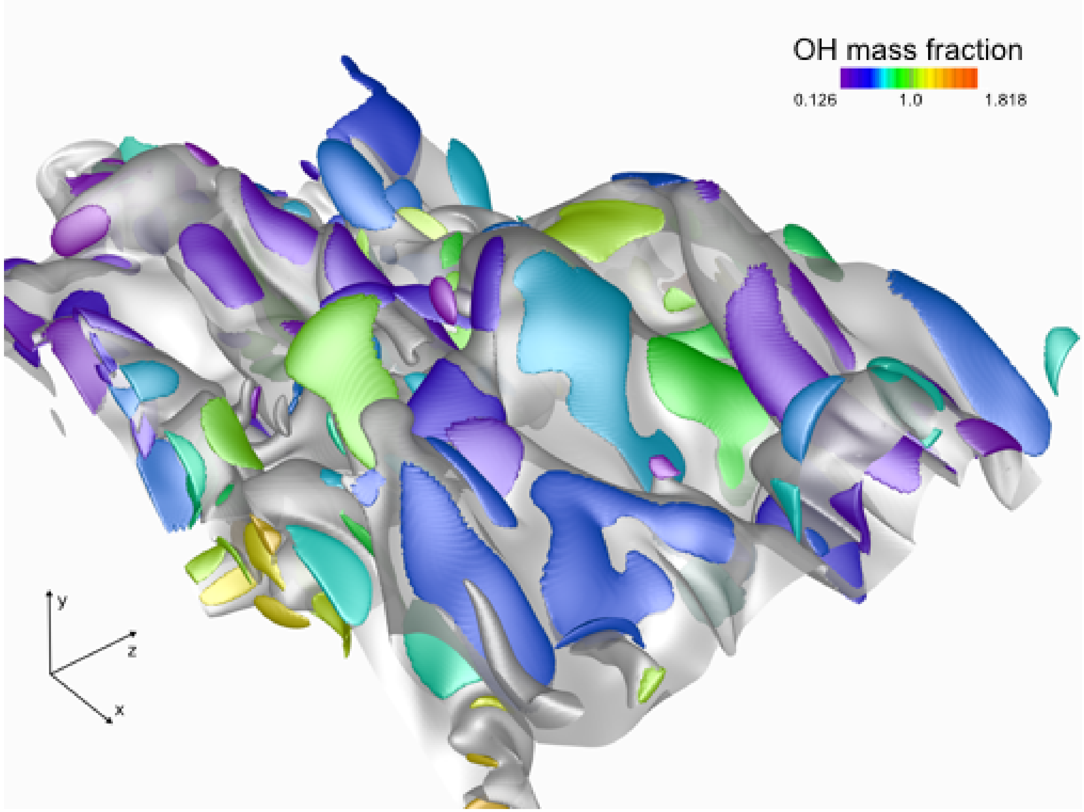
- Goals: Develop resilient algorithms and middleware to suport in situ data management, minimize data movement and storage requirments and to reduce power and manage the volume of data for future exascale combustion simulations
- ADIOS middleware: streaming middleware for in-transit processing; adaptive and dynamic approachs for in-transit processing to deal with variable data volumes AMR; accelerator support on compute or staging nodes via a PM
- Extreme distributed graphs (merge trees, MS complexes)
- Exploit intra-node parallelism for parallel image compositing

Simulation with in-situ analysis and visualization workflow



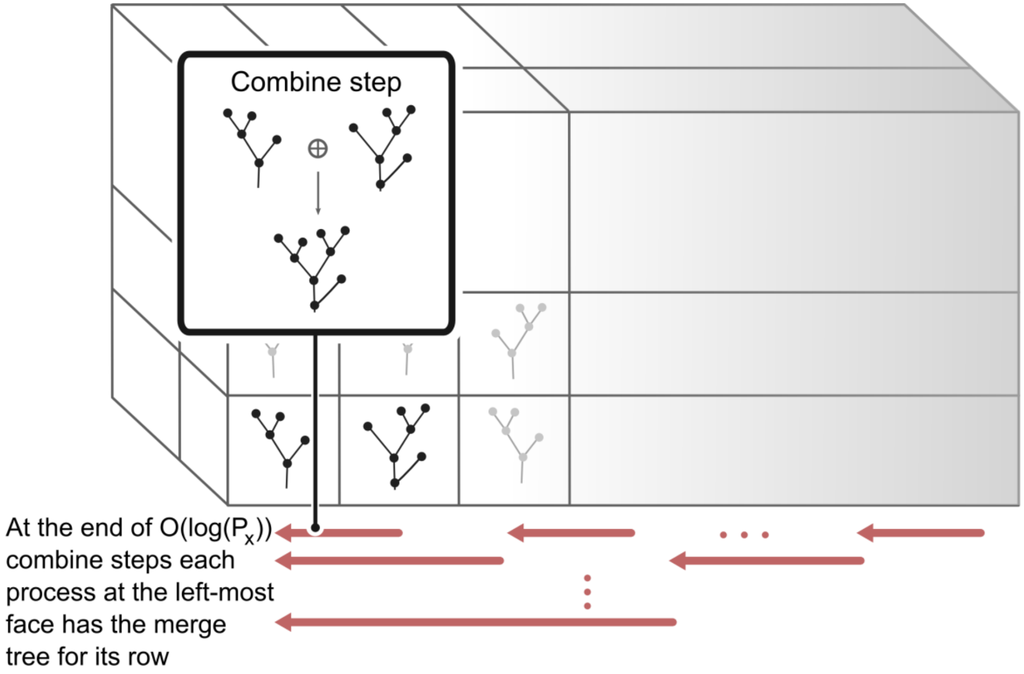
Distributed parallel computation of merge tree

- The *Merge tree* is a topological structure that can be used to segment and analyze scientific simulation data sets.



Distributed parallel computation of merge tree

- We have developed a distributed parallel algorithm to compute the merge tree of massive combustion data.
  - Tests on S3D data achieve run times < 10 seconds.

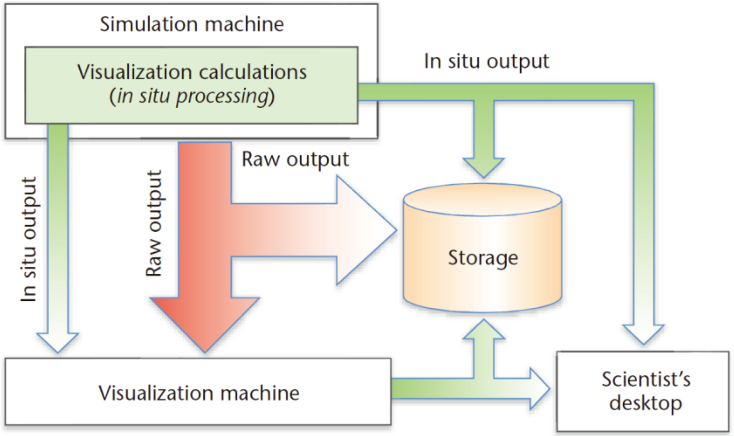


Alg. Phase	2.5K	10K	30K	129.6K
SINE				
Loc. MT.	21.858	4.739	1.408	0.363
Bound. crit. pts.	0.944	0.370	0.167	0.070
Combine	0.127	0.323	0.612	1.787
Total	22.929	5.432	2.187	2.220
CHI				
Loc. MT.	0.356	0.221	0.106	0.053
Bound. crit. pts.	0.861	0.340	0.154	0.071
Combine	4.970	4.880	4.966	4.658
Total	6.187	5.441	5.226	4.782

In-situ Visualization

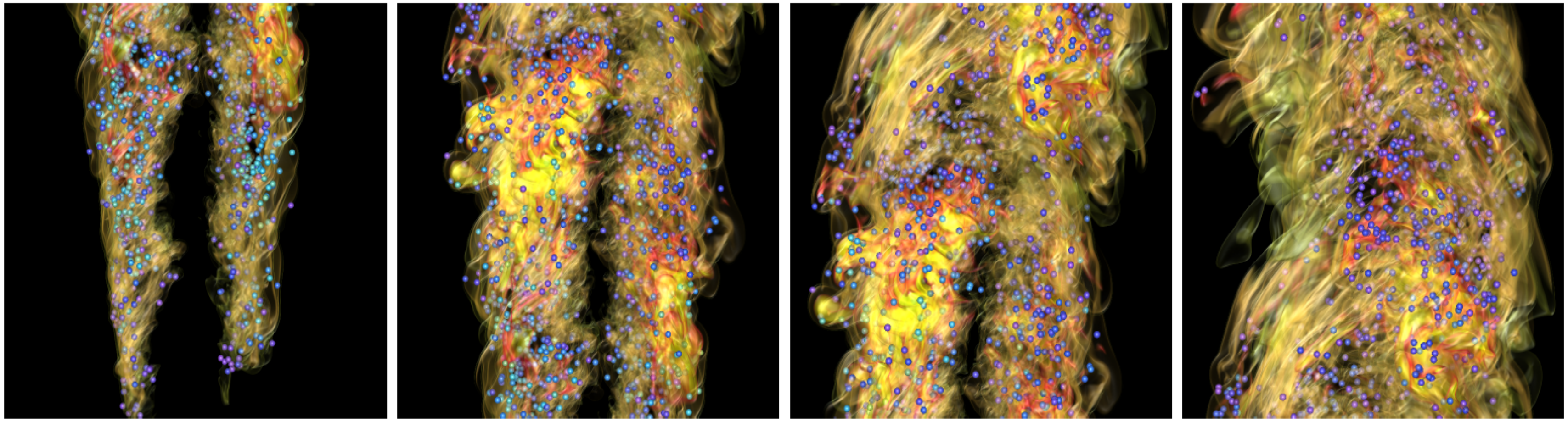
- Motivation
  - Scientists need efficient and effective solutions to manage and study their increasing amount of data
  - Traditional data analysis and visualization methods suffers from I/O and network bandwidth bound
  - In-situ processing is to transform or reduce the data on the same machine as the simulation runs to minimize the amount of data need to be stored or transferred

- Challenges
  - Integrate visualization code and simulation code to share same data structure and optimize memory usage
  - Make visualization as scalable as simulation with simulation data partitioning and distribution
  - Perform visualization with a low cost that is only a very small fraction of simulation time



In-situ Visualization

- In-situ visualization for S3D Combustion simulations
  - Design grid adaptor mechanism to ease the integration
    - Simulation only provides data partition and pointer of field and particle data to grid adaptor
    - Visualization directly takes data regions from grid adaptor
  - Highly scalable parallel volume rendering, particle rendering, and image compositing
  - Visualization time is less than 1% of simulation time if visualization is performed every 10th time step *(based on the experimentation results with 15,360 cores, 1620x1280x320 volume size, and 1024x1024 image size on JaguarPF at ORNL)*



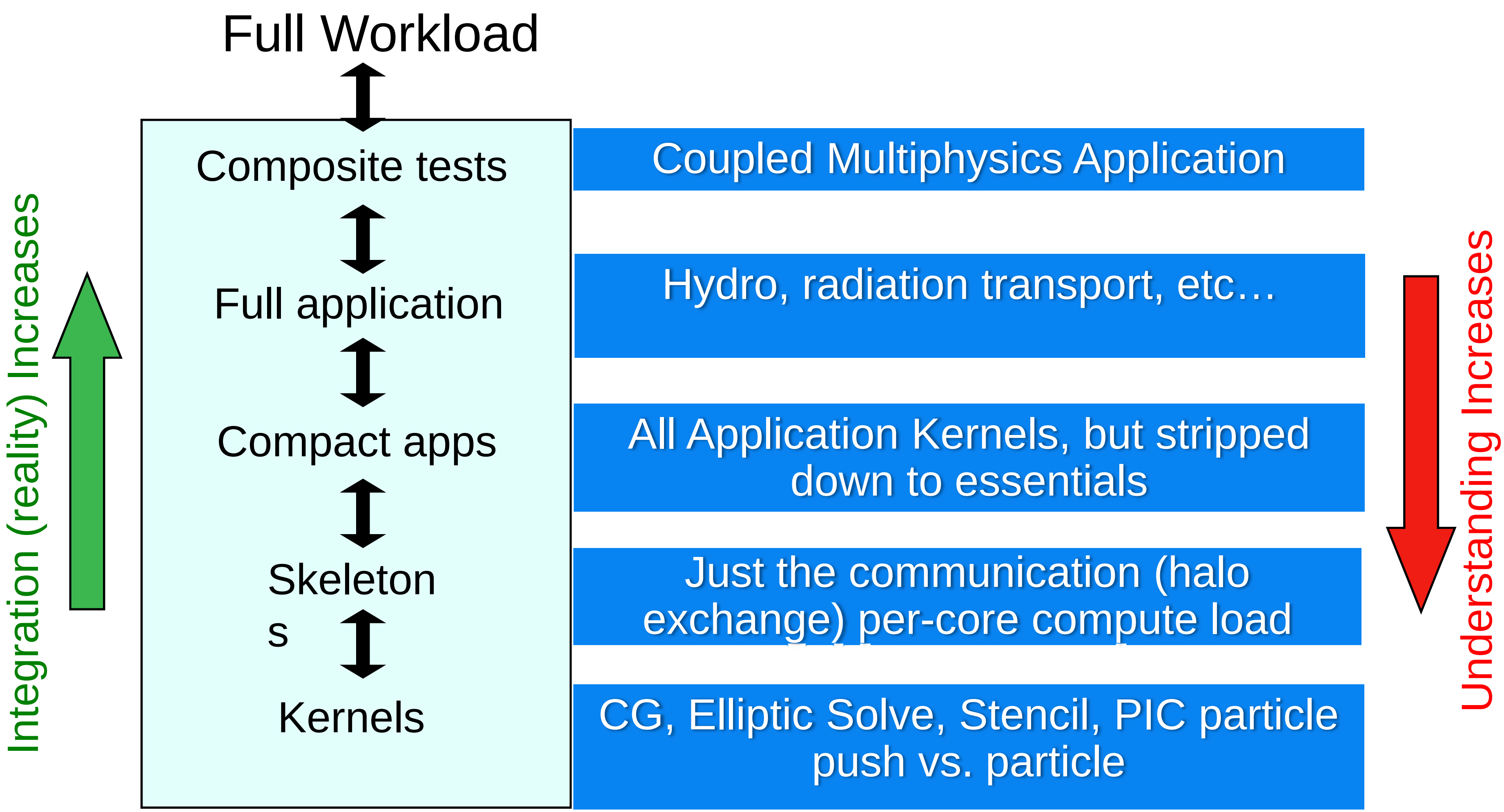
Selected zoomed-in views of mix rendering of volume and particle data (volume variable CH2O and particle variable HO2)

In-situ Visualization

- Future Work
  - Continue studying in-situ processing for selected applications to understand the impact on simulations, subsequent visualization tasks, and scientists' work processes
    - In-situ methods to prepare data for portraying time-varying particles and how they are clustered into groups with distinct characteristics
    - In-situ methods to generate compact data representations of each time step at simulation time that can be used for subsequent operations on the opacity and color mappings after simulation
    - In-situ methods to compute and visualize metrics suitable for quantifying variable correlations using statistical analysis techniques

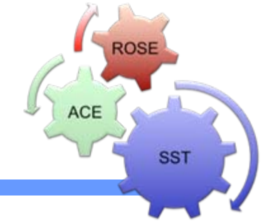
Hierarchy of Application Proxies

(enable hierarchy of understanding for co-design process)



Wasserman 2006

Architectural Simulation for CoDesign

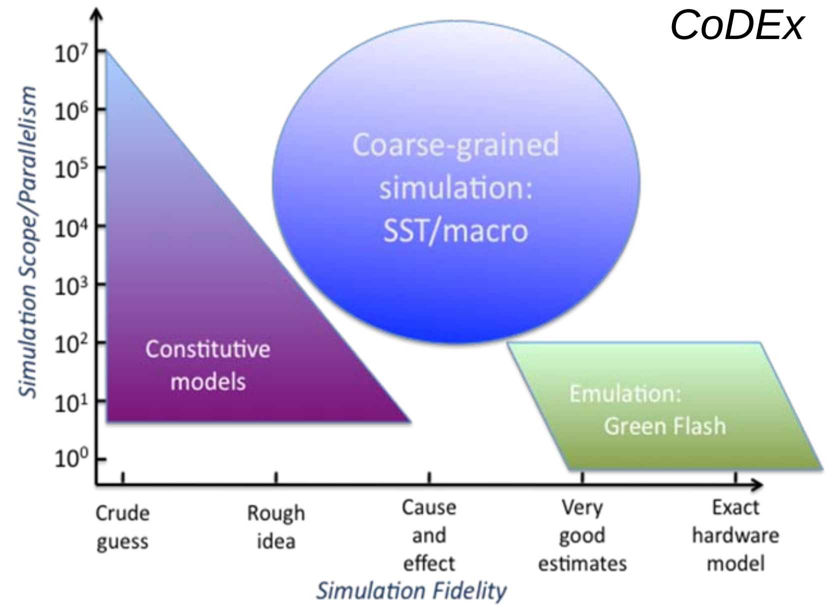


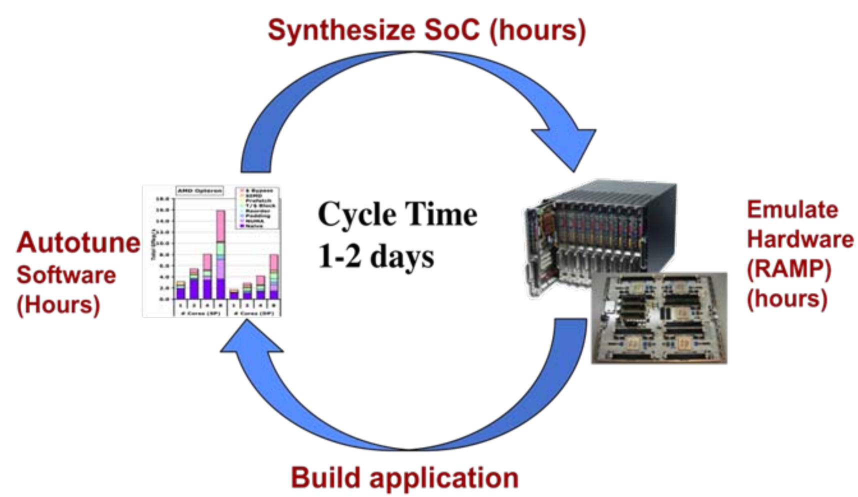
ROSE Compiler: **Enables deep analysis of application requirements, semi-automatic generation of skeleton applications, and code generation for ACE and SST.**

ACE Node Emulation: **Rapid design synthesis and RAMP/FPGA-accelerated emulation for rapid prototyping cycle accurate models of manycore node designs.**

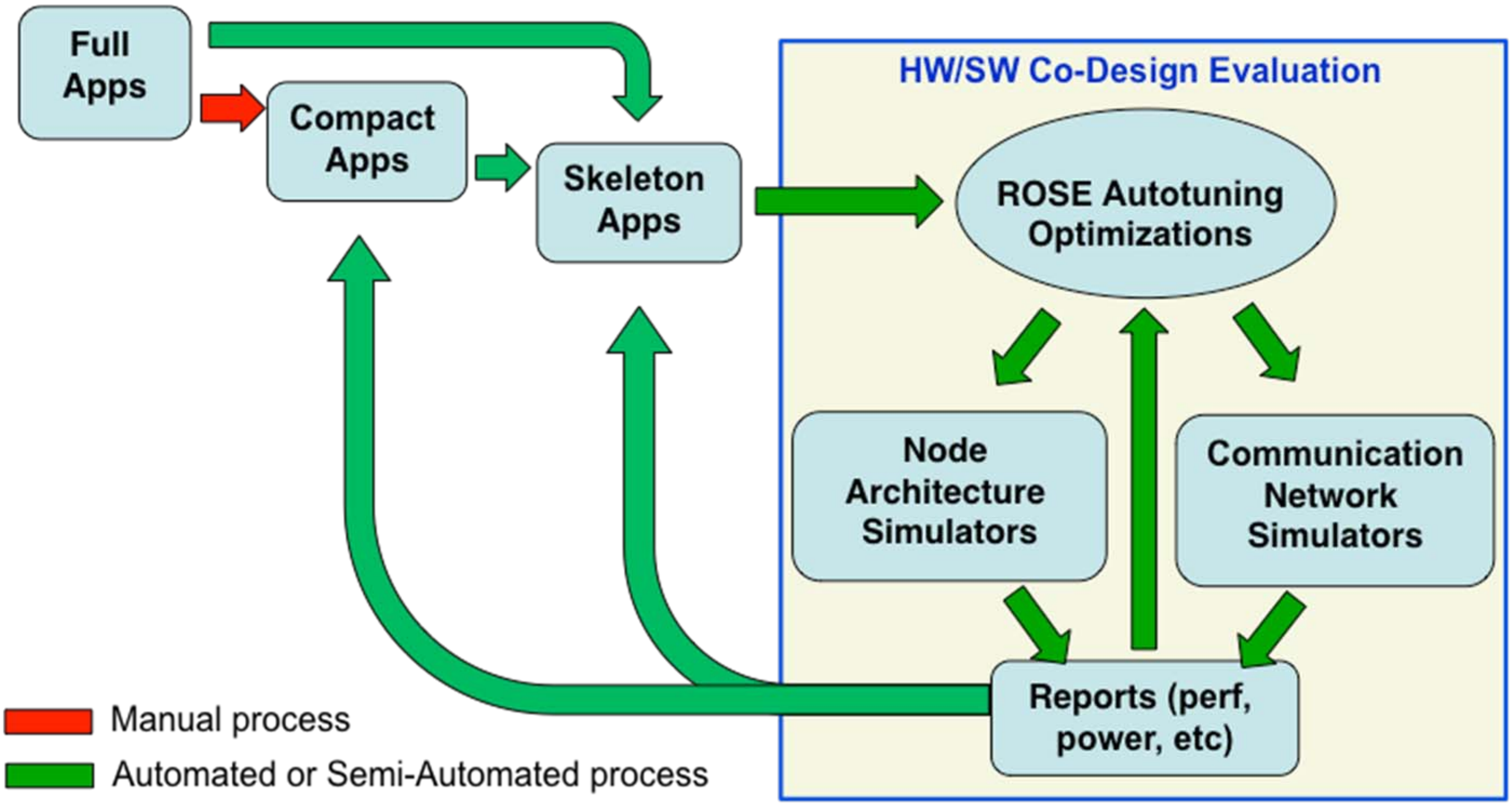
SST System Simulation: **Enables system-scale simulation through capture of application communication traces and simulation of large-scale interconnects.**

- **Simulate hardware *before* it is built!**
- **Break slow feedback loop for system designs**
- **Protect vendor IP**
- ***Insert applications and algorithms into the tightly coupled HW/SW CoDesign process***





CoDesign Tool Flow



Dan Quinlan 2011