ExaCT CENTER FOR EXASCALE SIMULATION OF COMBUSTION IN TURBULENCE

A Brief Introduction to the ExaCT Co-Design Center

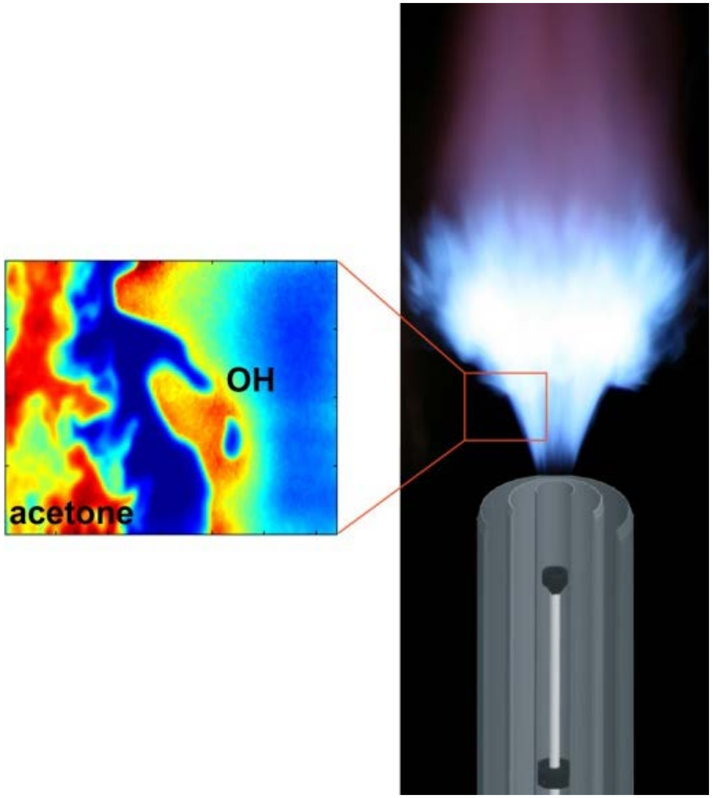
John Bell
Deputy Director
Lawrence Berkeley National Laboratory
Berkeley, CA

X-Stack Kickoff Meeting
Hillsboro, Oregon
Sept 18-19, 2012

Physics of Gas-Phase Combustion represented by PDE's

- Focus on gas phase combustion in both compressible and low-Mach limits

- Fluid mechanics
  - Conservation of mass
  - Conservation of momentum
  - Conservation of energy
- Thermodynamics
  - Pressure, density, temperature relationships for multicomponent mixtures
- Chemistry
  - Reaction kinetics
- Species transport
  - Diffusive transport of different chemical species within the flame



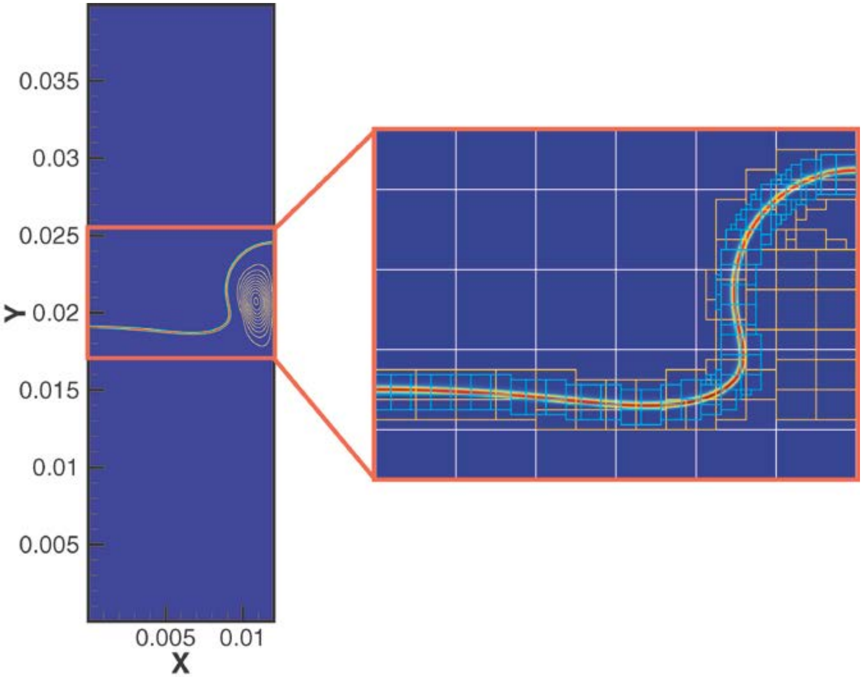
Stratified burner and OH/acetone PLIF imaging

Code base

- S3D
  - Fully compressible Navier Stokes
  - Eighth-order in space, fourth order in time
  - Fully explicit, uniform grid
  - Time step limited by acoustics / chemical time scales
  - Hybrid implementation with MPI + OpenMP
  - Implemented for Titan at ORNL using OpenACC
- LMC
  - Low Mach number formulation
  - Projection-based discretization strategy
  - Second-order in space and time
  - Semi-implicit treatment of advection and diffusion
  - Time step based on advection velocity
  - Stiff ODE integration methodology for chemical kinetics
  - Incorporates block-structured adaptive mesh refinement
  - Hybrid implementation with MPI + OpenMP
- Target is computational model that supports compressible and low Mach number AMR simulation with integrated UQ

Adaptive Mesh Refinement

- Need for AMR
  - Reduce memory
  - Scaling analysis – For explicit schemes flops scale with memory ^ 4/3
- Block-structured AMR
  - Data organized into logically-rectangular structured grids
  - Amortize irregular work
  - Good match for multicore architectures
- AMR introduces extra algorithm issues not found in static codes
  - Metadata manipulation
  - Regridding operations
  - Communications patterns



Preliminary observations

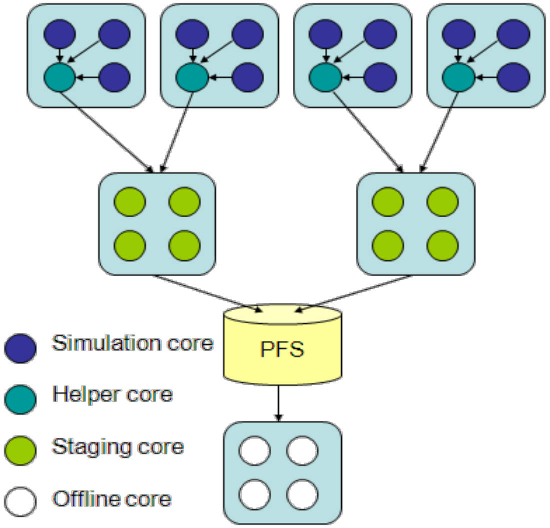
- Need to rethink how we approach PDE discretization methods for multiphysics applications
  - Exploit relationship between scales
  - More concurrency
  - More locality with reduced synchronization
  - Less memory / FLOP
  - Analysis of algorithms has typically been based on a performance = FLOPS paradigm – can we analyze algorithms in terms of a more realistic performance model
- Need to integrate analysis with simulation
  - Combustion simulations are data rich
  - Writing data to disk for subsequent analysis is currently near infeasibility
  - Makes simulation look much more like physical experiments in terms of methodology
- Current programming models are inadequate for the task
  - We describe algorithms serially and add things to express parallelism at different levels of the algorithm
  - We express codes in terms of FLOPS and let the compiler figure out the data movement
  - Non-uniform memory access is already an issue but programmers can't easily control data layout
- Need to evaluate tradeoffs in terms of potential architectural features

How core numerics will change

- Core numerics
  - Higher-order for low Mach number formulations
  - Improved coupling methodologies for multiphysics problems
  - Asynchronous treatment of physical processes
- Refactoring AMR for the exascale
  - Current AMR characteristics
    - Global flat metadata
    - Load-balancing based on floating point work
    - Sequential treatment of levels of refinement
  - For next generation
    - Hierarchical, distributed metadata
    - Consider communication cost as part of load balancing for more realistic estimate of work (topology aware)
    - Regridding includes cost of data motion
    - Statistical performance models
    - Alternative time-stepping algorithm – treat levels simultaneously

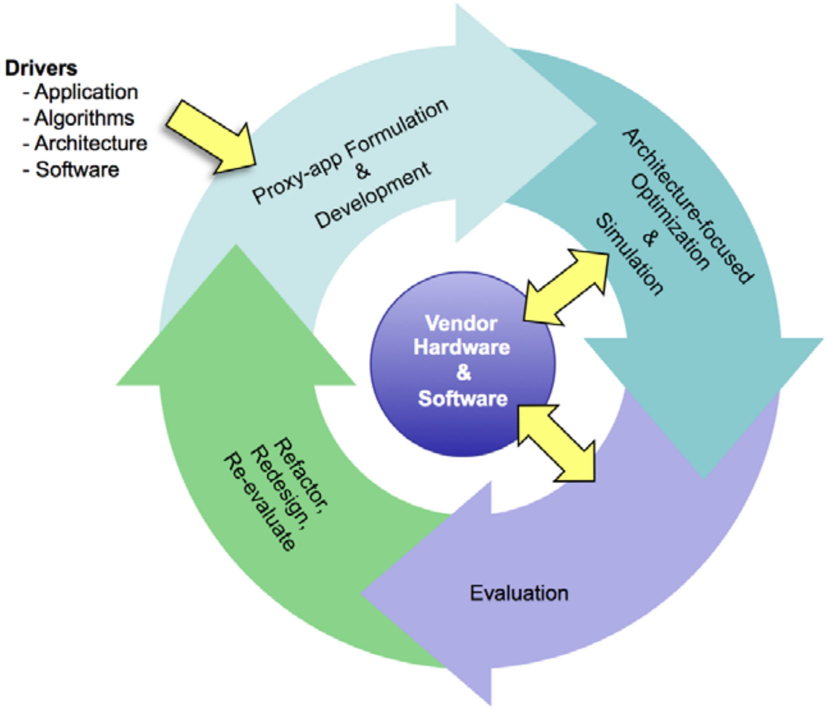
Data analysis

- Current simulations produce 1.5 Tbytes of data for analysis at each time step (Checkpoint data is 3.2 Tbytes)
  - Archiving data for subsequent analysis is currently at limit of what can be done
  - Extrapolating to the exascale, this becomes completely infeasible
- Need to integrate analysis with simulation
  - Design the analysis to be run as part of the simulation definition
    - Visualizations
    - Topological analysis
    - Lagrangian tracer particles
    - Local flame coordinates
    - Etc.
- Approach based on hybrid staging concept
  - Incorporate computing to reduce data volume at different stages along the path from memory to permanent file storage



Co-design Process

- Identify key simulation element
  - Algorithmic
  - Software
  - Hardware
- Define representative code (proxy app)
- Analytic performance model
  - Algorithm variations
  - Architectural features
  - Identify critical parameters
- Validate performance with hardware simulators / measurements
- Document tradeoffs
  - Input to vendors
  - Helps define programming model requirements
- Refine and iterate



Proxy Applications

- Caveat
  - Proxy apps are designed to address a specific co-design issue.
  - Union of proxy apps is not a complete characterization of application
  - Anticipated methodology for exascale not fully captured by current full applications
- Proxies
  - Compressible Navier Stokes without species
    - Basic test for stencil operations, primarily at node level
    - Coming soon – generalization to multispecies with reactions (minimalist full application)
  - Multigrid algorithm – 7 point stencil
    - Basic test for network issues
    - Coming soon – denser stencils
  - Chemical integration
    - Kernel test for local, computationally intense kernel
  - Others coming soon
    - Integrated UQ kernels
    - Skeletal model of full workflow
    - Visualization / analysis proxy apps

Visualization/Topology/Statistics Proxy Apps

- Proxies are algorithms with flexibility to explore multiple execution models
  - Multiple strategies for local computation algorithms
  - Support for various merge/broadcast communication patterns
- Topological analysis
  - Three phases (local compute/communication/feature-based statistics)
  - Low/no flops, highly branching code
  - Compute complexity is data dependent
  - Communication load is data dependent
  - Requires gather/scatter of data
- Visualization
  - Two phases (local compute/image compositing)
  - Moderate FLOPS
  - Compute complexity is data dependent
  - Communication load is data dependent
  - Requires gather
- Statistics
  - Two phases (local compute/aggregation)
  - Compute is all FLOPs
  - Communication load is constant and small
  - Requires gather, optional scatter of data

These are coming soon. Contact us for early access.

Summary / X-Stack Interactions

- Co-Design methodology
  - Identify hardware / software / application issue
  - Create proxy app to encapsulate the issue
  - Evaluate impact of algorithm and hardware variations on performance
    - Analytic models, measurement, simulation
  - Iterate
- Proxies do not provide complete coverage
  - If you would like to pursue a particular issue, we can make a suitable proxy app aimed at addressing that issue for combustion simulation
- New programming model is critical element
  - Ability to express information about application needed for performance
  - Access to machine characteristics needed to achieve performance
  - Level of abstraction to ensure portability while maintaining reasonable performance
  - Need to respect characteristics of "real" codes
- For additional information:
  - http://exactcodesign.org
  - Contact:
    - jbbell@lbl.gov for PDE solver aspects
    - jhchen@sandia.gov for SDMA aspects
- There will be a 3 hour deep-dive at the Exascale PI Meeting, October 1-3, 2012 to provide details about combustion simulation