SAND2015-2855C

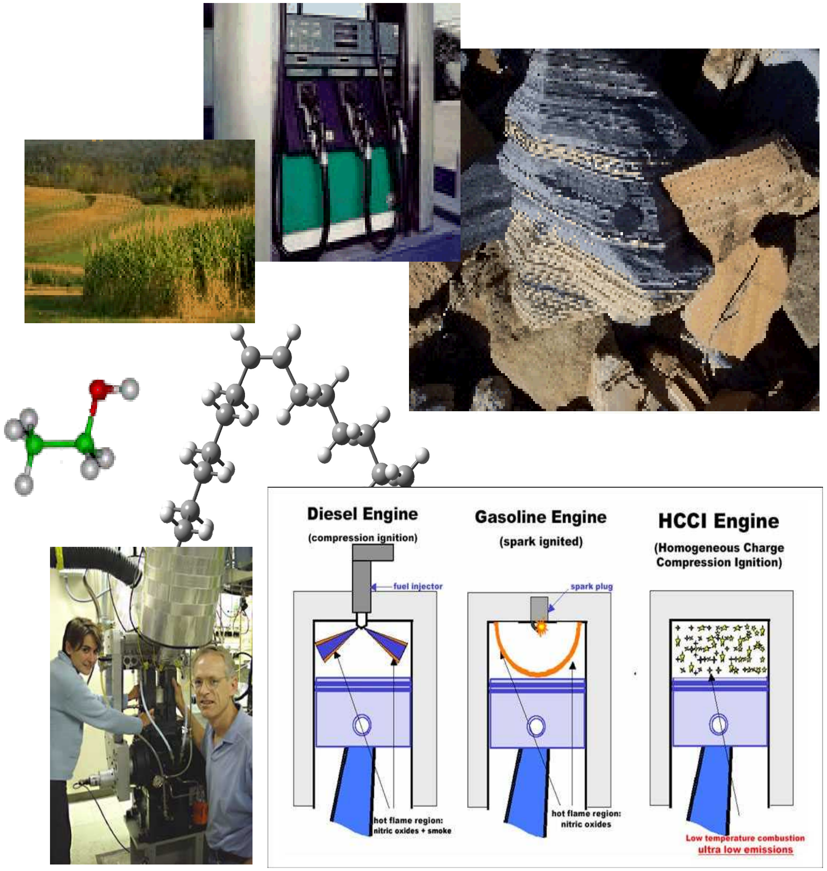
Legion as a Programming Model for Combustion Simulation at the Exascale

Jacqueline Chen
Sandia National Laboratories
Livermore, CA
jhchen@sandia.gov
www.exactcodesign.org

Mini-Symposium: Towards Exascale Simulation of Turbulent Combustion
NC15 April 19-22, 2015
Avignon, France

Why Exascale Combustion

- Predict behavior of new fuels in different combustion scenarios at realistic pressure and turbulence conditions
  - Develop new combustor concepts
  - Design new fuels
- Co-design center is focusing on high-fidelity direct numerical simulation methodologies
  - Need to perform simulations with sufficient chemical fidelity to differentiate effects of fuels where there is strong coupling with turbulence
  - Need to address uncertainties in thermo-chemical properties
  - Not addressing complexity of geometry in engineering design codes

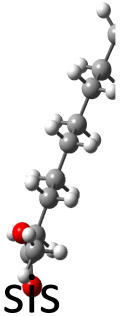

Diesel Engine
(compression ignition)
fuel injector
hot flame region: nitric oxides + smoke
Gasoline Engine
(spark ignited)
spark plug
hot flame region: nitric oxides
HCCI Engine
(Homogeneous Charge Compression Ignition)
Low temperature combustion
ultra low emissions

Fundamental Turbulence-Chemistry Interactions Motivated by Advanced Engines and Gas Turbines

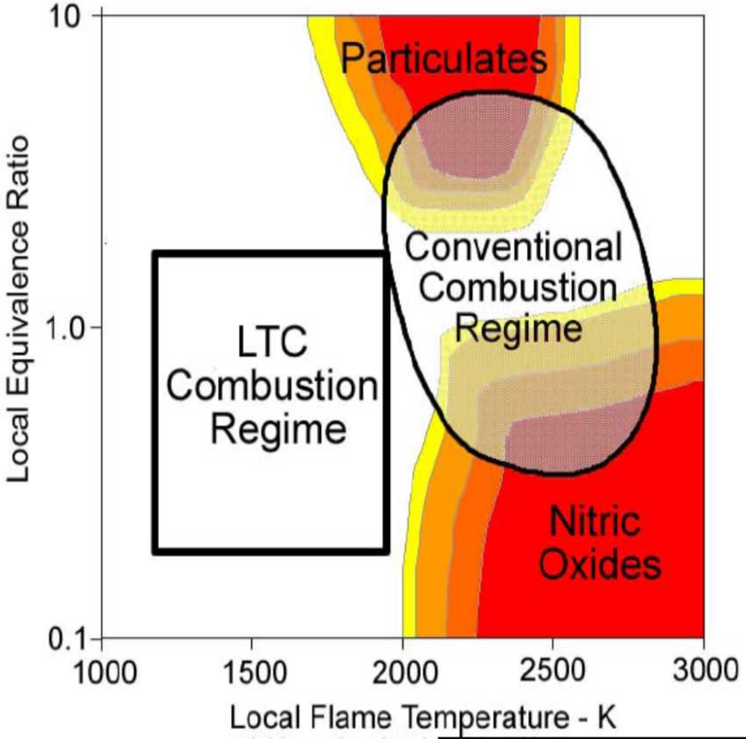
- Higher fuel efficiency and lower emissions driving combustion towards more **dilute, fuel lean, partially-premixed conditions**

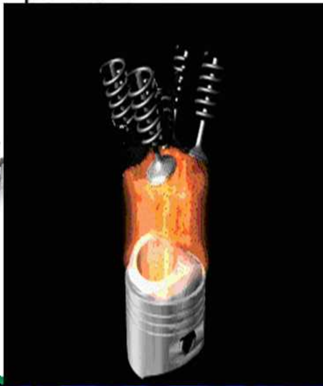
- New **mixed-mode combustion regimes**

- Strong **sensitivities to fuel chemistry**



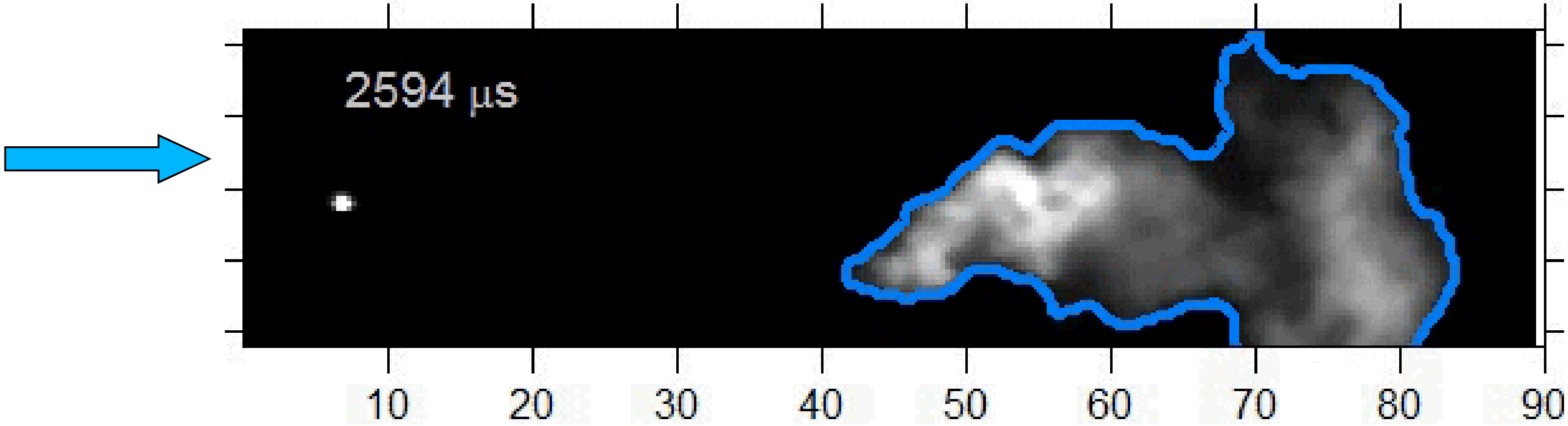
- **Preferential diffusion effects** – synthesis gases enriched with hydrogen for carbon capture storage in gas turbines for power



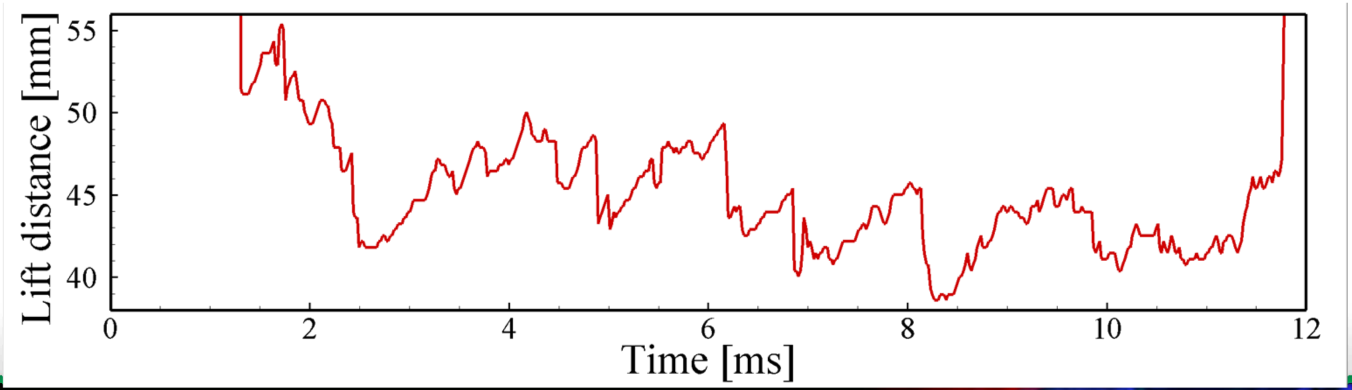


Motivation: Understanding Stabilization of Lifted Flames in Heated Coflow

What is the role of ignition in lifted flame stabilization?

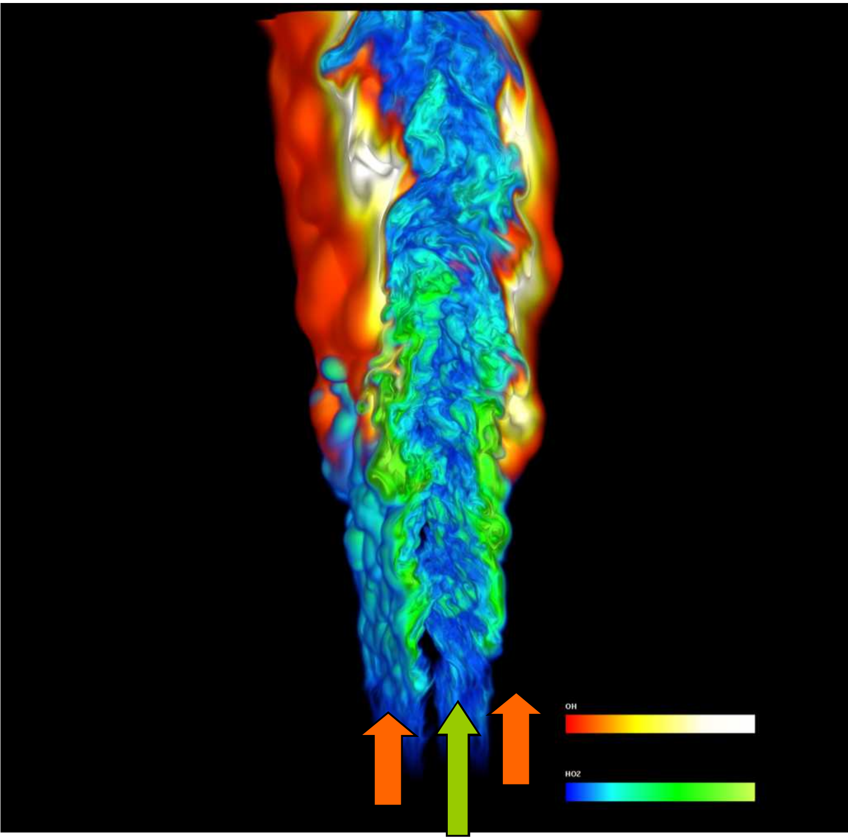


Chemiluminescence from diesel lift-off stabilization for #2 diesel, ambient 21% O_2, 850K, 35 bar courtesy of Lyle Pickett, SNL



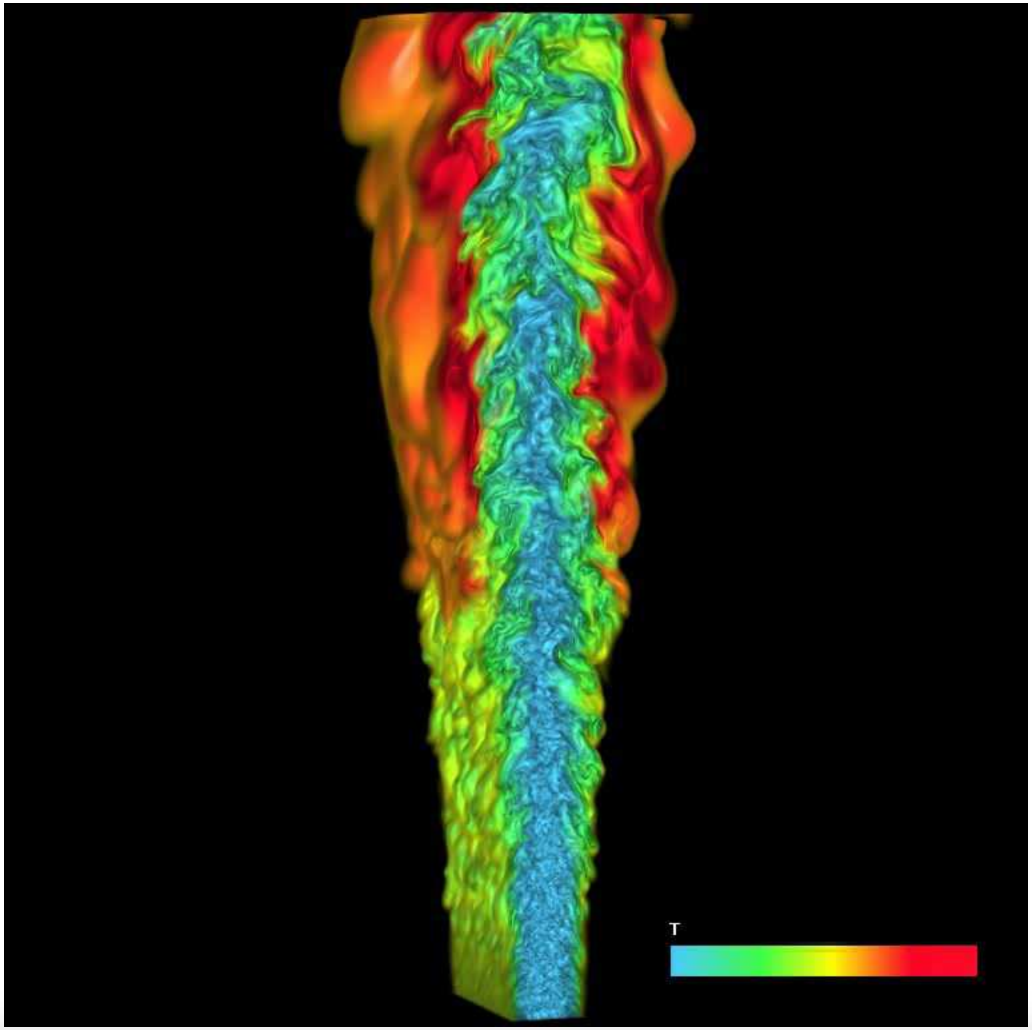
DNS of Lifted Ethylene-air Jet Flame in a Heated Coflow

- 3D slot burner configuration:
  - $L_x \times L_y \times L_z = 30 \times 40 \times 6$ mm^3 with
  - 1.28 billion grid points
  - High fuel jet velocity (204m/s); coflow velocity (20m/s)
  - Nozzle size for fuel jet, $H = 2.0$mm
  - $\mathrm{Re}_{jet} = 10{,}000$
  - Cold fuel jet (18% C_2H_4 + 82% N_2) at 550K, $\eta_{st} \approx 0.27$
  - Detailed C_2H_4/air chemistry, 22 species 18 global reactions, 201 steps
  - Hot coflow air at 1,550K



Ethylene-air lifted jet flame at Re=10000

Dynamics of lifted flame stabilization – Log(scalar dissipation) and Temperature



Why does this need exascale?

- Turbulent combustion consists of phenomena occurring over a wide range of scales that are closely coupled
  - More grid points needed to resolve larger dynamic range of scales
  - More time steps needed for better statistics and less dependence on initial condition
- Complex fuels require higher number of equations per grid point
- In situ uncertainty quantification with adjoint sensitivity – reverse causality – uncertainties in chemical inputs
- In situ analytics/visualization
- Coupled execution (hybrid Eulerian-Lagrangian particle solver, or lockstep DNS/LES)

Why do we need to do co-design?

Old Constraints

- **Peak clock frequency:** as primary limiter for performance improvement
- **Cost:** *FLOPs* are biggest cost for system: *optimize for compute*
- **Concurrency**: Modest growth of parallelism by adding nodes
- **Locality***: MPI+X model (uniform costs within node & between nodes)*
- **Memory Scaling:** maintain byte per flop capacity and bandwidth
- **Uniformity**: Assume uniform system performance

New Constraints

- **Power:** primary design constraint for future HPC system design
- **Cost:** Data movement dominates: optimize to minimize data movement
- **Concurrency:** Exponential growth of parallelism within chips
- **Locality***:* must reason about data locality and possibly topology
- **Memory Scaling**: Compute growing 2x faster than capacity or bandwidth, no global hardware cache coherence
- **Heterogeneity:**Architectural and performance non-uniformity increase

- Future algorithms, programming environments, runtimes, hardware need to:
  - Express data locality (sometimes at the expense of FLOPS) and independence
  - Allow expression of massive parallelism
  - Minimize data movement and reduce synchronization
  - Detect and address faults

ExaCT Vision and Goal

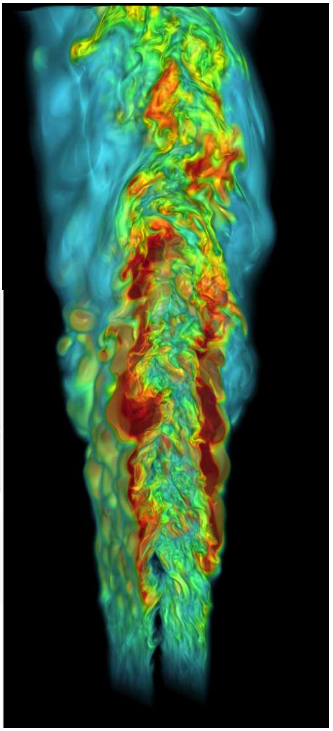
- Goal of combustion exascale co-design is to consider all aspects of the combustion simulation process from formulation and basic algorithms to programming environments to hardware characteristics needed to enable combustion simulations on exascale architectures
  - Interact with vendors to help define hardware requirements, computer scientists on requirements for programming environment and software stack, and applied mathematics community locality-aware algorithms for PDE's, UQ, and analytics

- Combustion is a surrogate for a much broader range of multiphysics computational science areas

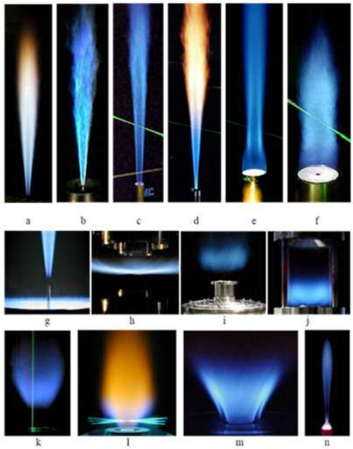
Petascale codes provide starting point for co-design process

- S3D
  - Compressible formulation
  - Eighth-order finite difference discretization
  - Fourth-order Runge-Kutta temporal integrator
  - Detailed kinetics and transport
  - Hybrid parallel model with MPI + OpenMP
  - MPI+ OpenACC (directives for GPU's)
  - Legion (deferred execution hides latencies)
- LMC
  - Low Mach Number model that exploits separation of scales between acoustic wave speed and fluid motion
  - Second-order projection formulation
  - Detailed kinetics and transport
  - Block-structure adaptive mesh refinement
  - Hybrid parallel model with MPI + OpenMP

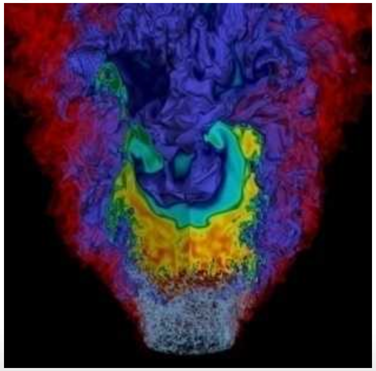
Expectation is that exascale will require new code base



S3D simulation of HO_2 ignition marker in lifted flame



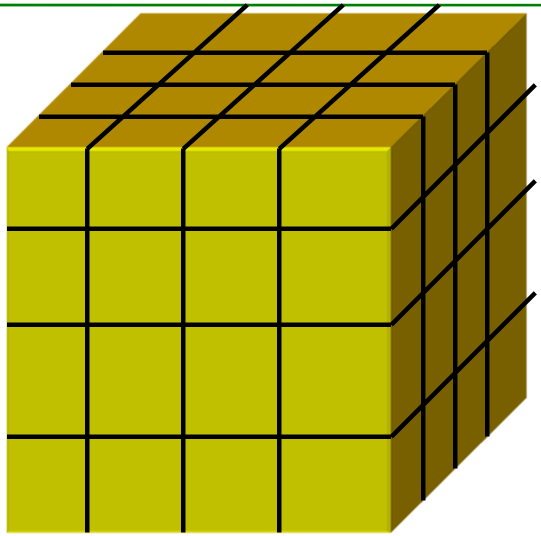
Laboratory scale flames

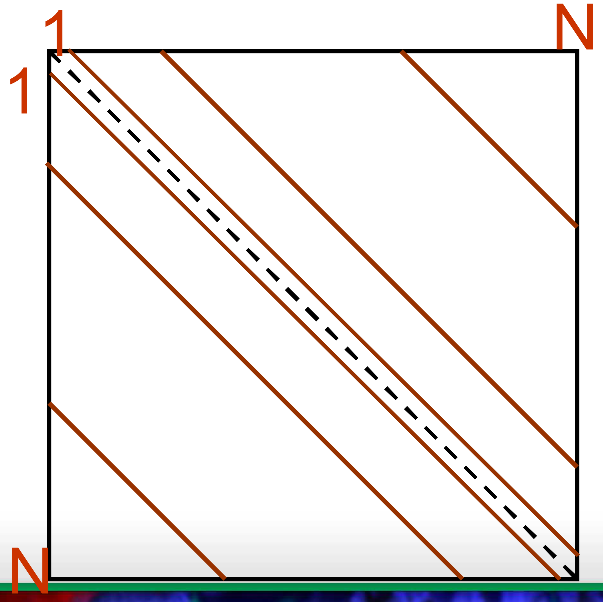


LMC simulation of NOx emissions from a low swirl injector

S3D MPI Parallelism

- 3D domain decomposition.
  - Each MPI process is in charge of a piece of the 3D domain.
- All MPI processes have the same number of grid points and the same computational load
- Inter-processor communication is only between nearest neighbors in 3D topology
  - Large message sizes. Non-blocking sends and receives
- All-to-all communications are only required for monitoring and synchronization ahead of I/O
- Good parallel scaling on Titan





What happens in the main solver?

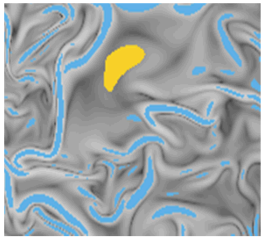
- Computes rate of change of N conserved quantities at every grid point
  - d/dt (Q_k) = (Advection) + (Diffusion) + (Source)
  - Sum of all the terms that contribute to the time derivative is called the RHS
- d/dt (Q_k) is integrated explicitly in time through Runge-Kutta
- RHS contains multiple terms that are functions of Q_k, variables derived from Q_k
- Advection and diffusion require finite differencing and MPI
- Source terms are point-wise functions
- Thermodynamic, chemical and molecular transport properties are point-wise functions of Q_k

Source term is the most compute intensive kernel

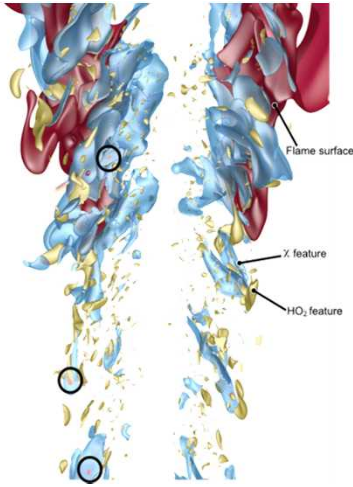
- Called as ckwyp or getrates
- Chemical reaction rate computed using Arrhenius model
  - A + B ⇔ C + D
  - Forward reaction rate = C*[A]*[B]*T^aexp(-Ta/T)
  - Equilibrium constant gives reverse reaction rates
  - More terms for third body efficiency, collision efficiency, pressure corrections …
- The source term for a species is the sum of the rates of all reactions in which it participates
- The kernel uses exp/log heavily

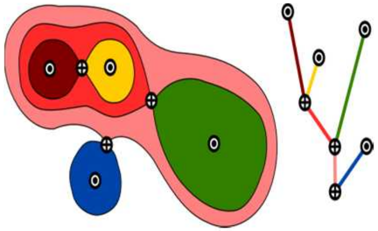
In Situu Uncertainty Quantification Guided by Analytics

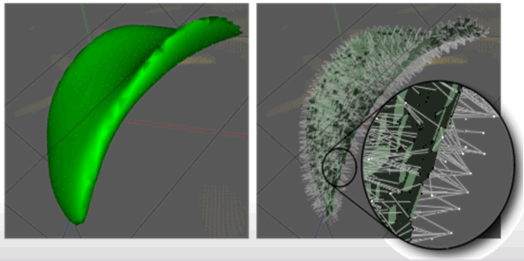
- Uncertainty in reaction rates characterizing ignition/extinction events that control fuel efficiency and emissions with respect to uncertainties in input chemical and transport parameters
- Solve adjoint equations backward in time: need the primal state at all times
- Exploit space-time locality guided by analytics to bound regions of interest



- Topological Segmentation and Tracking
  - ➢ Topological segmentation and tracking
  - ➢ Distance field (level set)
- Statistics
  - ➢ Filtering and averaging (spatial and temporal)
  - ➢ Statistical moments (conditional)
  - ➢ Statistical dimensionality reduction (joint PDFS)
  - ➢ Spectra (scalar, velocity, coherency)
  - ➢ Chemical Explosive Mode

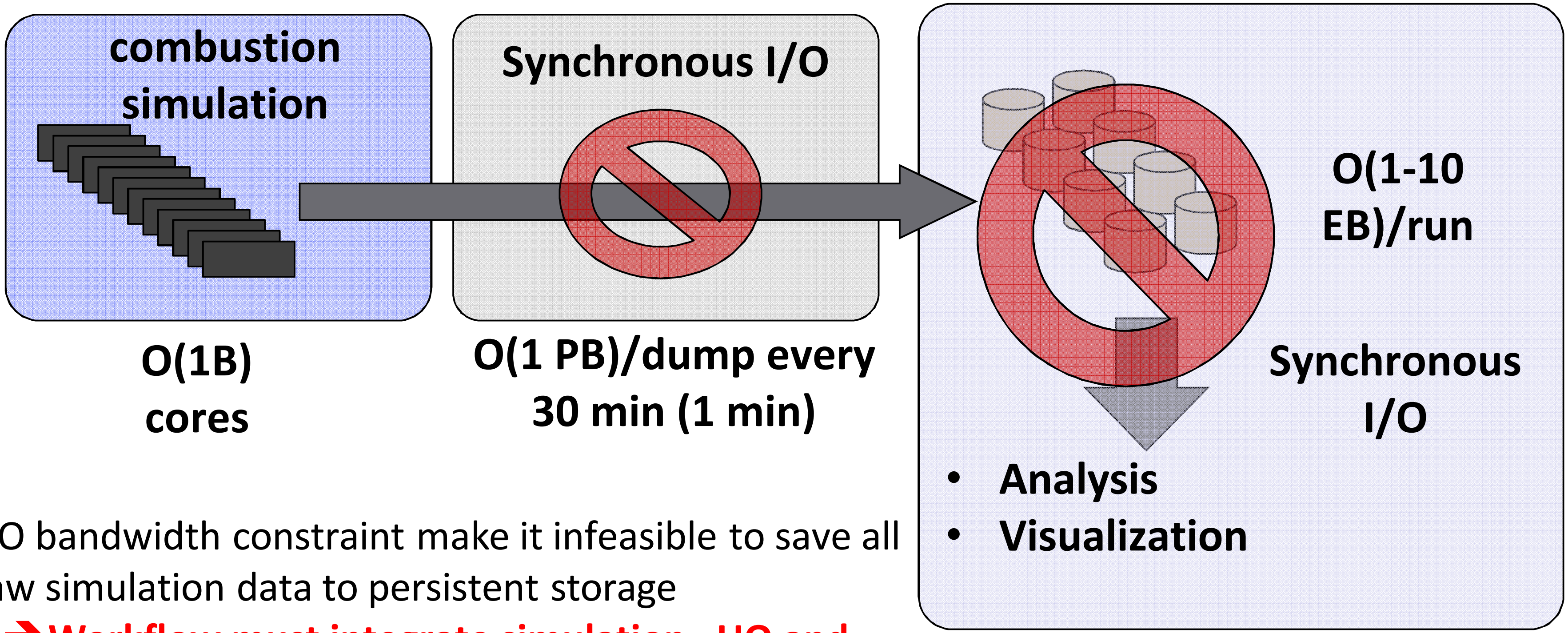






Petascale Workflow Model Won't Scale

Performing the simulation is not enough – need to analyze results



- I/O bandwidth constraint make it infeasible to save all raw simulation data to persistent storage

  ➔Workflow must integrate simulation , UQ and analysis !!!!
- Challenge: co-design a workflow that supports smart placement of solver + UQ, and analytics, reducing checkpointing size with in-situ and in-transit analytics

Programming Environment Critical to Performance

Effective use of exascale hardware will require programming environment that effectively maps algorithms to hardware

- Driven by programmability of combustion applications and characterization of algorithms on different designs of architectures
  - Simplify programming to express locality and independence
  - Simplify programming of block-structured PDE's, analytics, UQ for performance, scalability & portability on heterogeneous architectures with high variability and still maintain readability

Legion Approach

- Capture the structure of program data
- Decouple specification from mapping
- Automate data movement, parallelism discovery, synchronization, hiding long latency

Goal: A performance-oriented programming model with a raised abstraction for productivity and portability.

http://legion.stanford.edu

(Bauer, Treichler, Slaughter, Aiken – Structure Slicing: Extending Logical Regions with Fields, Thursday Nov 20 1:30-2:00, Rm 388-390)

Legion Programming Model

- Application logic expressed as hierarchy of tasks
- Application data expressed as hierarchy of logical regions
  - Subregions are subset of elements and/or subset of fields
- Application mapping computed by mapper object(s)
  - Selection of processors, memories, instance layouts
- Legion runtime handles:
  - Extraction of parallelism (task--- and data---)
  - Resource management and movement of data
  - Scheduling of execution on heterogenous processors
  - Hiding overhead of all this with use of deferred execution

Describing Data

- Logical regions (array of objects):
  - Have no implied layout
  - Have no implied location

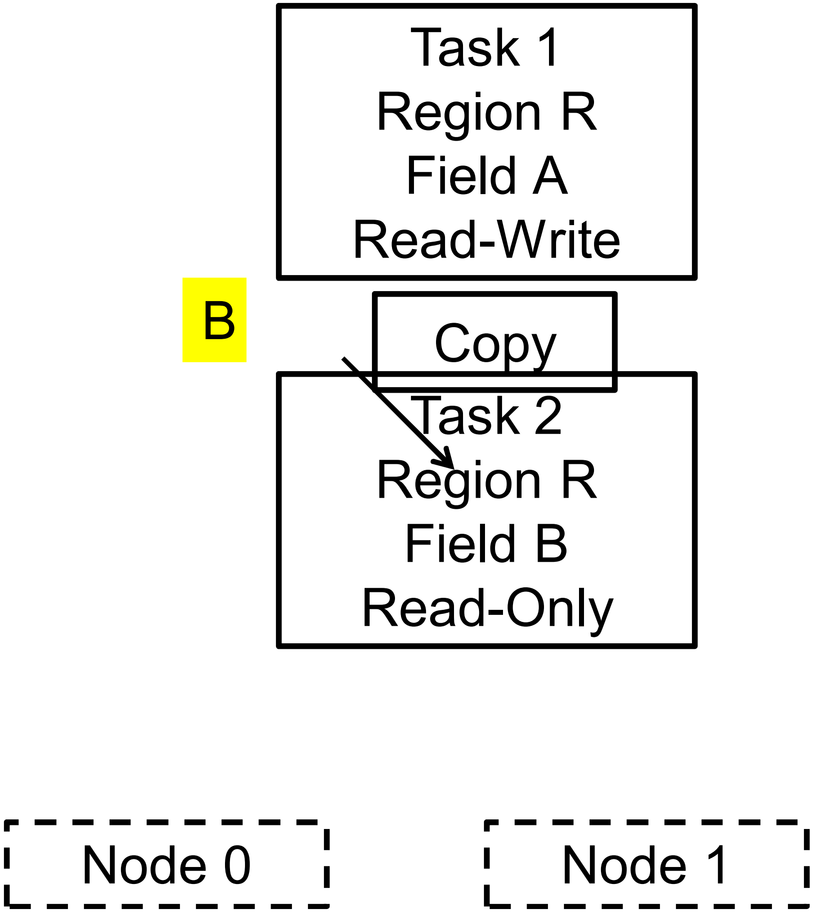
- Described by:
  - Index space (set of keys)
  - Field space (set of fields)

- Operations include:
  - Partitioning into subregions
  - Slicing by fields

		Field Space			
		f0	f1	f2	f3
Index Space	p0				
	p1				
	p2				
	p3				
	p4				
	p5				
	p6				
	p7				
	p8				
	p9				
	p10				
	p11				

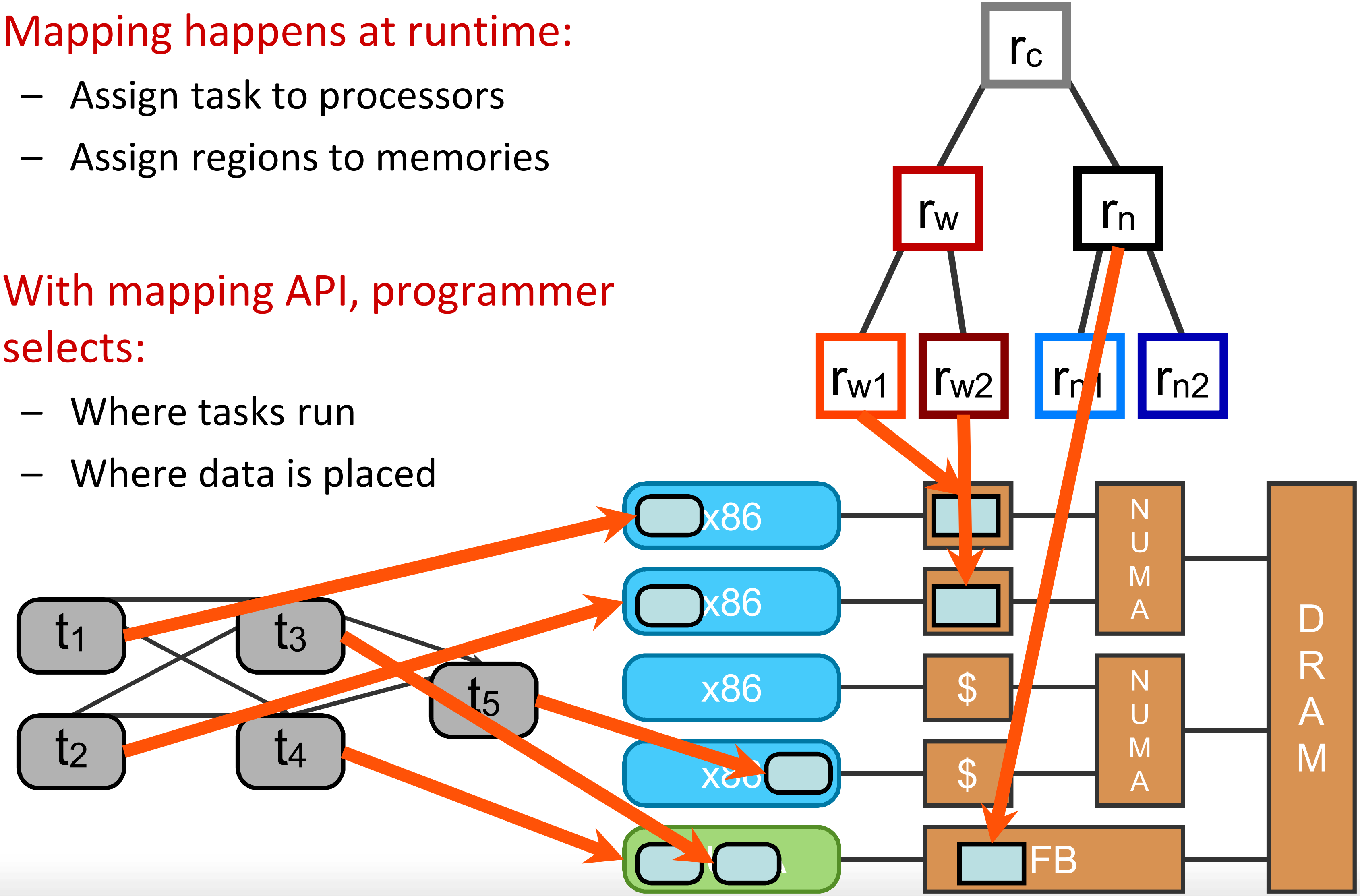
Legion Tasks

- Hierarchical Tree of Tasks
- Legion tasks specify:
  - Region usage
  - Field usage
  - Access modes
- Legion runtime:
  - Infers data dependences
  - Inserts copies



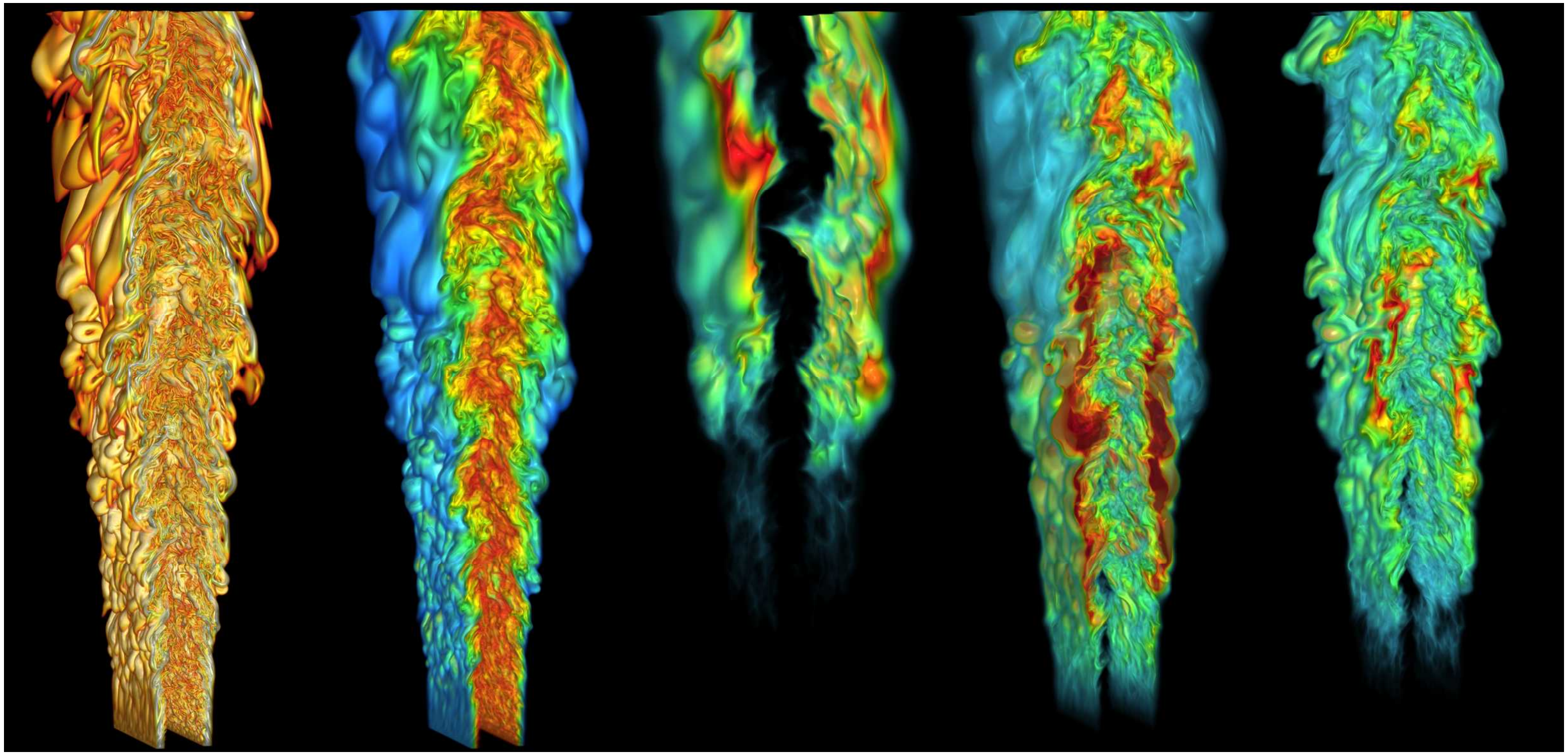
Legion Mapping

- Mapping happens at runtime:
  - Assign task to processors
  - Assign regions to memories
- With mapping API, programmer selects:
  - Where tasks run
  - Where data is placed



Application: S3D

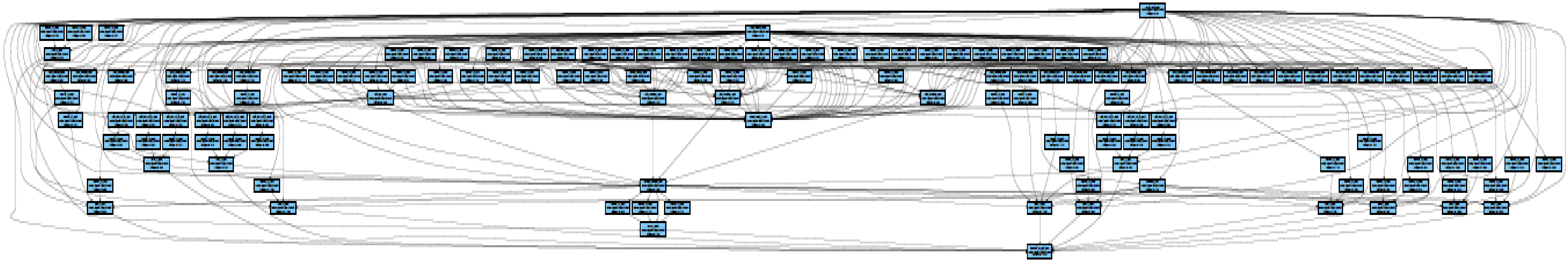
- Current state-of-the-art combustion DNS simulation
  - Written in 200K lines of Fortran
  - Direct numerical solver using explicit methods
  - Many phases, lots of parallelism



χ ξ OH HO_2 CH_2O

S3D Task Parallelism

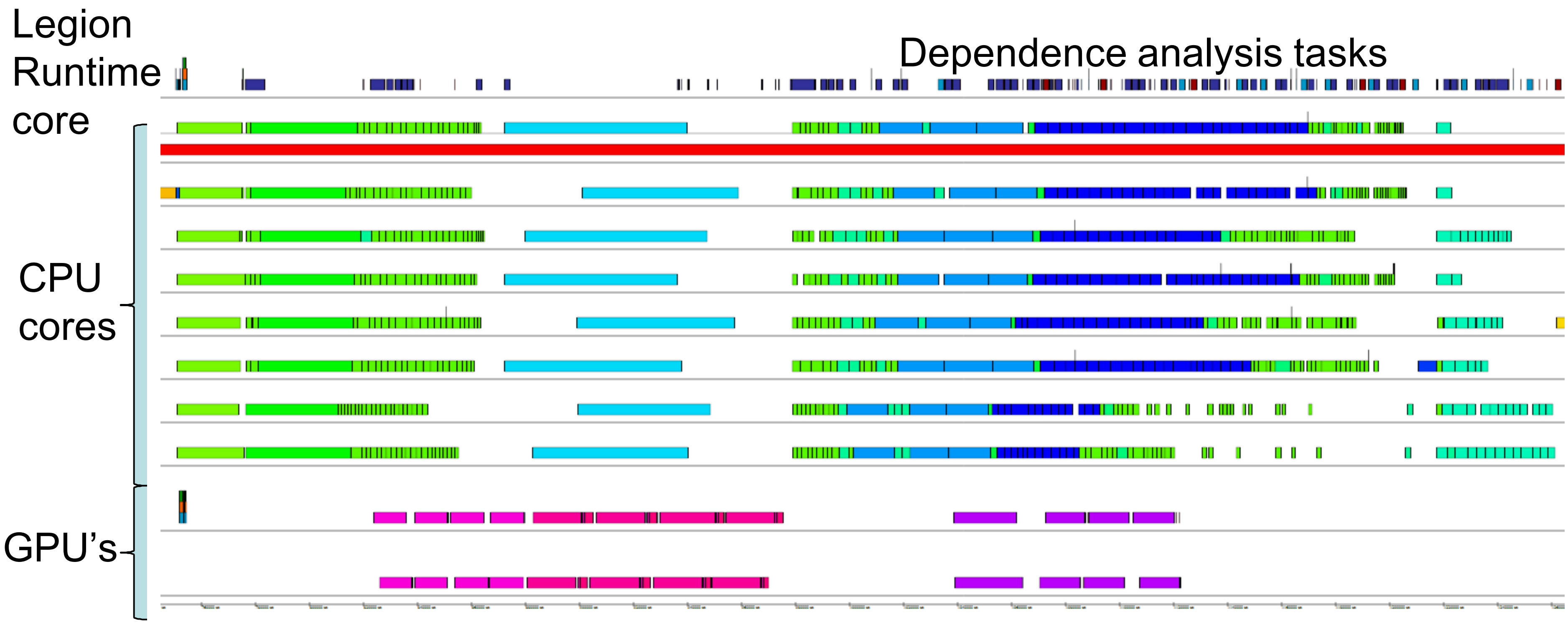
- One call to Right-Hand-Side-Function (RHSF) as seen by the Legion runtime
  - Called 6 times per time step by Runge-Kutta solver
  - Width == task parallelism
  - H2 mechanism (only 9 species)
  - Heptane (52 species) is significantly wider
- Manual task scheduling would be difficult!



S3D Execution Profile

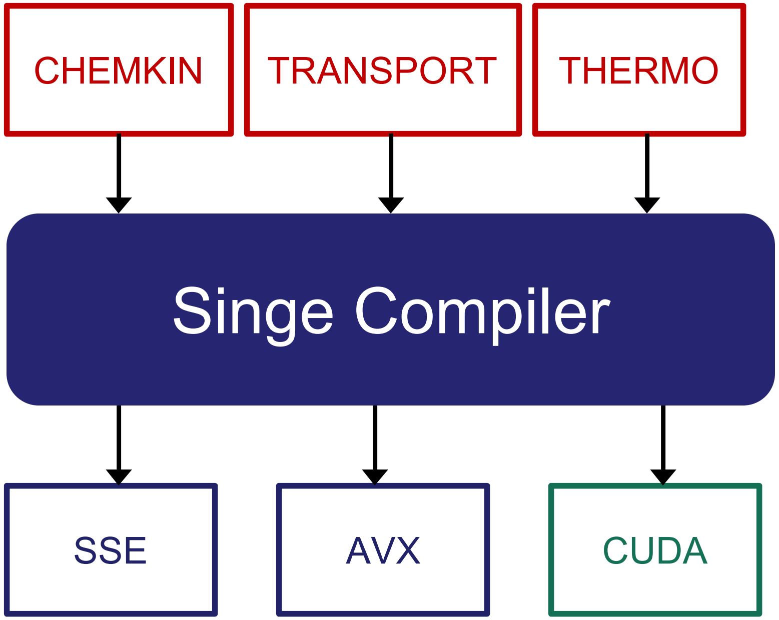
- Two custom mappers
  - Schedule tasks onto CPUs and GPUs
  - Prioritize tasks on critical path of RHSf

Legion execution of RHSF on one node



Leaf Tasks

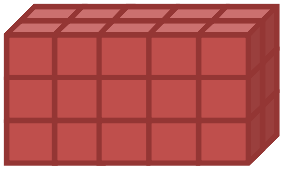
- Legion treats tasks as black boxes
  - Doesn’t care how tasks are written
- Still need fast leaf tasks for computationally expensive chemistry, diffusion, viscosity
  - For CPUs & GPUs
  - For multiple mechanisms
- Singe* is a DSL compiler for chemistry kernels

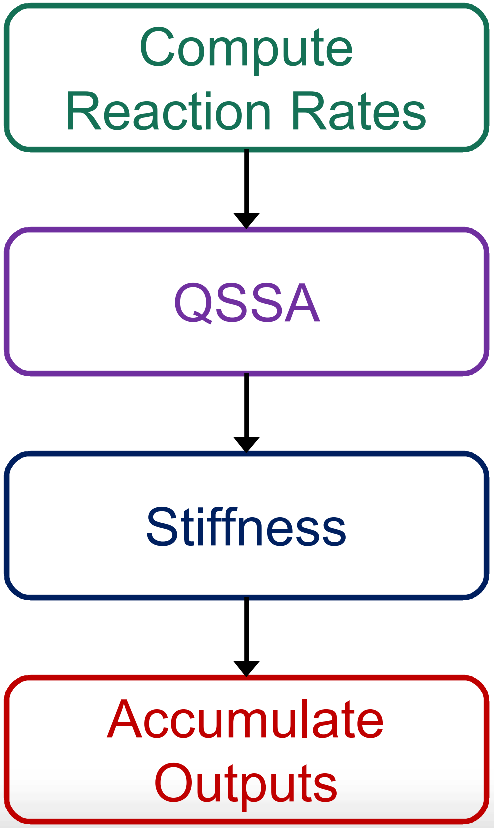


*Bauer et al. PPoPP’14

Combustion Challenges

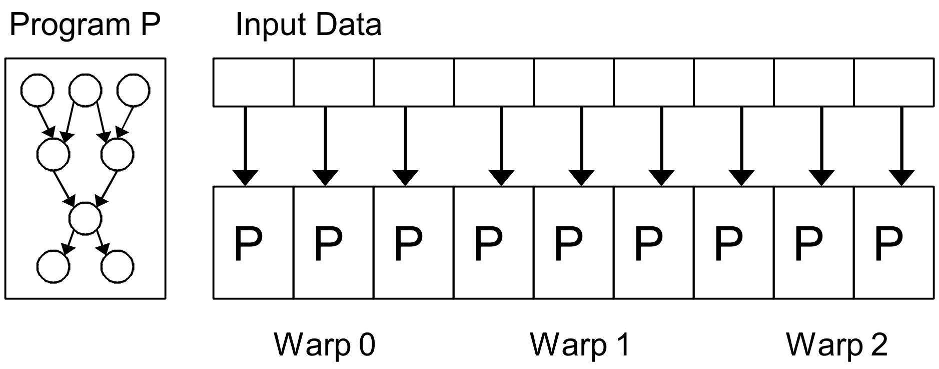
- GPU programming models emphasize data parallelism
  - Not always the best choice for performance
- Large working sets (per point)
  - nHeptane chemistry needs 566 double precision reaction rates (per point)
  - GPU register file only store 128 per thread
- Multi-phase computations
  - Fissioned kernels limited by memory bandwidth, slow



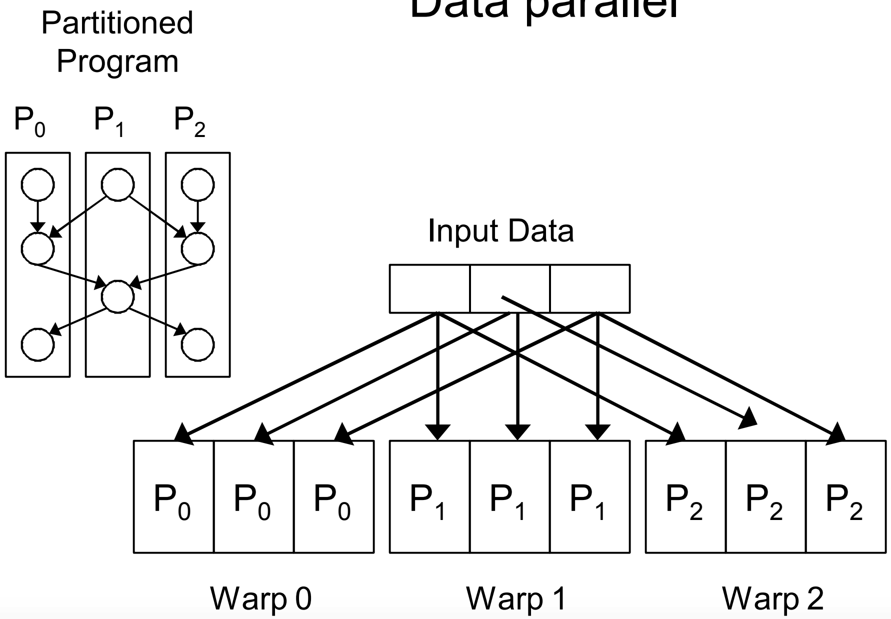


Warp Specialization

- Leverage knowledge of underlying hardware
  - GPUs execute warps: streams of 32-wide vector instructions
  - All threads in warp execute the same program (data parallel unit)
- Each warp can run different computation
  - Generate code that specializes each warp,(leverage task parallelism)
  - Different warps do different computations on the same data
  - Allows much better use of memory while keeping processors busy
  - Fit large working sets on chip

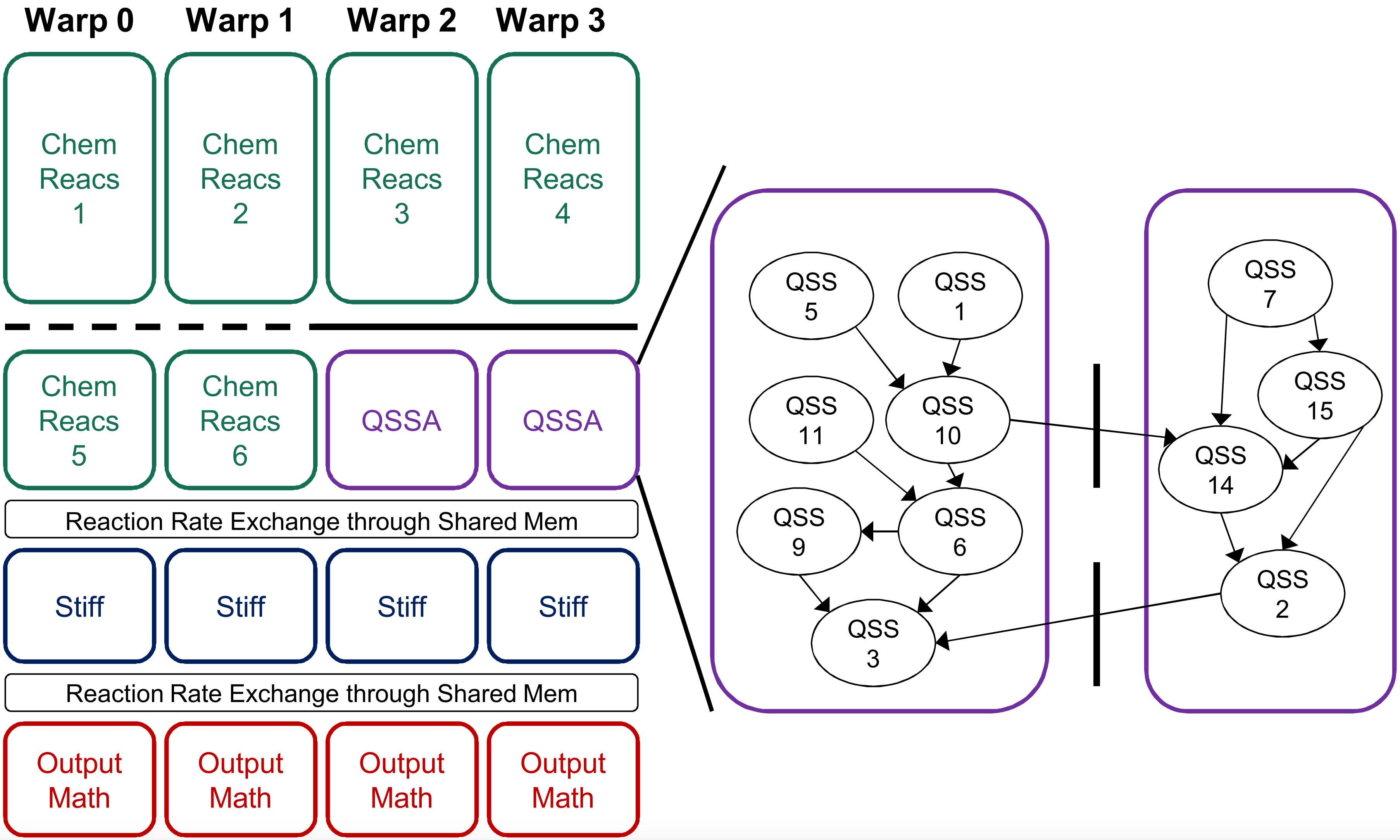


Data parallel



Warp specialization

Singe Warp-Specialized Chemistry Kernel



Warp 0	Warp 1	Warp 2	Warp 3
Chem Reacs 1	Chem Reacs 2	Chem Reacs 3	Chem Reacs 4
Chem Reacs 5	Chem Reacs 6	QSSA	QSSA
Reaction Rate Exchange through Shared Mem			
Stiff	Stiff	Stiff	Stiff
Reaction Rate Exchange through Shared Mem			
Output Math	Output Math	Output Math	Output Math

Performance Results

- Chemistry Kernel
  - All Singe kernels significantly faster than current production versions
  - Warp specialized SINGE code is up to 3.75 times faster than previously optimized data-parallel CUDA kernels

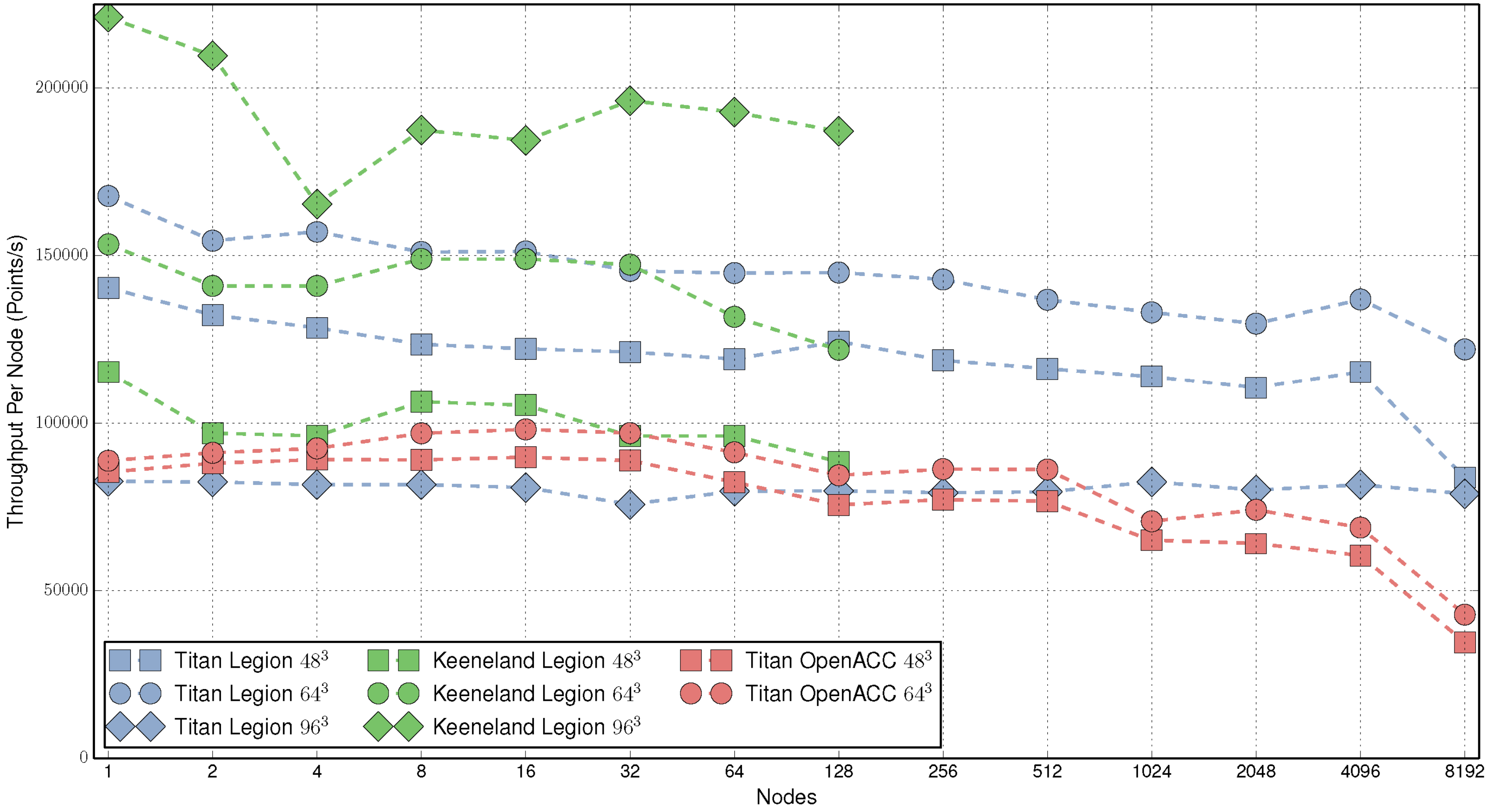
- Multi-Node Heterogeneous Testbeds S3D Legion:
  - Keeneland: 128 nodes, 16 Sandy Bridge cores, 3 Fermis
  - Titan: 18K nodes, 16 Interlagos cores, 1 K20 GPU

S3D Performance Comparison

- Compare against MPI+OpenACC code on Titan
  - Tuned by Cray and NVIDIA engineers with ORNL/NREL domain experts
- OpenACC runs 48^3 and 64^3 for DME and heptane
  - Fixed mapping (most compute on GPU)
- Legion runs 48^3, 64^3, and 96^3 for any mechanism (DME, heptane, PRF)
  - Try both All-GPU and mixed CPU+GPU mappings
- Use re-ranking script for runs >= 1024 nodes

Legion S3D Performance on Titan (weak scaling)

N-heptane 52 species

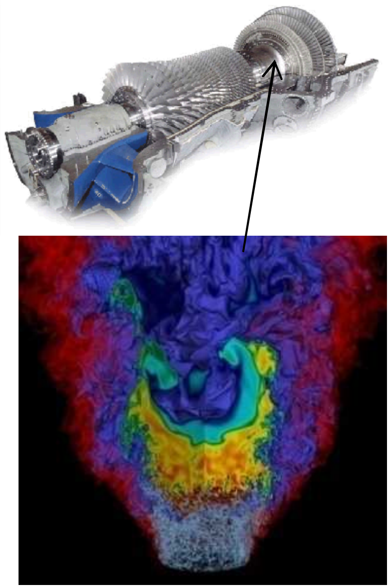


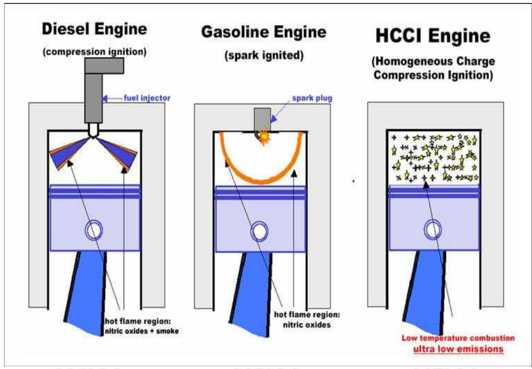
1.71X - 2.33X faster between 1024 and 8192 nodes DME

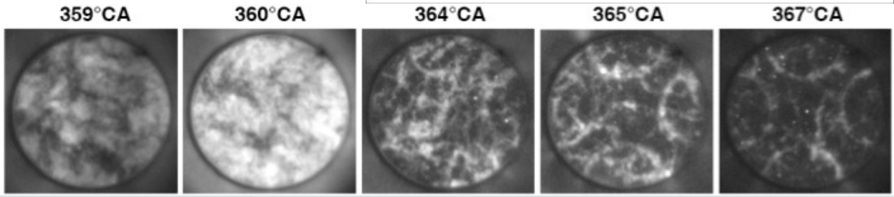
1.88-2.85X faster between 1024 and 8192 nodes n-heptane (64*3)

Exascale Use Cases: Science at Relevant Conditions

- **Homogeneous Charge Compression Ignition (HCCI engine combustion)** – ‘Chemical’ engine with high diesel-like efficiency without NOx and soot, tailor the charge stratification to control ignition and burn rate
- **Turbulent Jet Flames (Swirl, transverse, cavity)** – low-swirl Injector gas turbines with staged lean premixed combustion, flame stabilization, emissions
- **Lifted Diesel Jet flames –**lifted autoignitive diesel jet flame stabilization with multi-stage ignition fuels
- **Need to include UQ with respect to chemistry and transport properties**
- **Extrapolation of current capability show that these cases will require exascale-level resources**




Diesel Engine
(compression ignition)
fuel injector
hot flame region: nitric oxides + smoke
Gasoline Engine
(spark ignited)
spark plug
hot flame region: nitric oxides
HCCI Engine
(Homogeneous Charge Compression Ignition)
Low temperature combustion
ultra low emissions



Thanks!

- Questions?

- jhchen@sandia.gov
- www.exactcodesign.org