Porting S3D to Titan (Turbulent Combustion)

OLCF
OAK RIDGE LEADERSHIP COMPUTING FACILITY

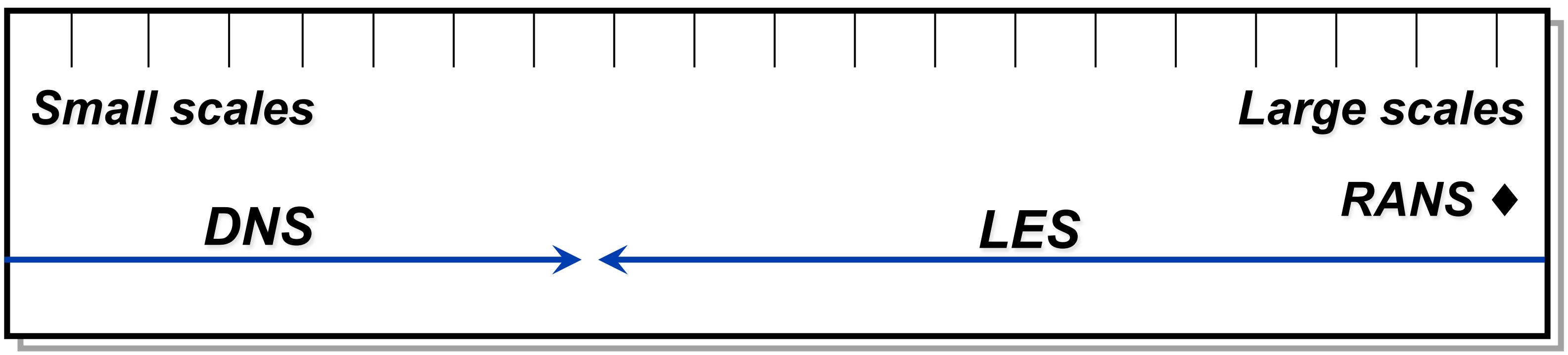
Application Readiness Team	
G. Bansal, J. H. Chen	SNL
R. W. Grout	NREL
R. Sankaran, K. Spafford, W. Wang	ORNL
M. Fatica, N. Juffa, G. Ruetsch	Nvidia
J. Levesque, J. Schwarzmeier	Cray

Presented by: Ramanan Sankaran

August 15, 2011

Direct numerical simulations (DNS)

- Turbulent combustion occurs over a wide range of scales
  - Device sizes are O(1m)
  - Diffusive scales and flame thickness O(10-100 μm)
  - Non-linear coupling and interaction among the entire range of scales
- Combustion CFD approaches

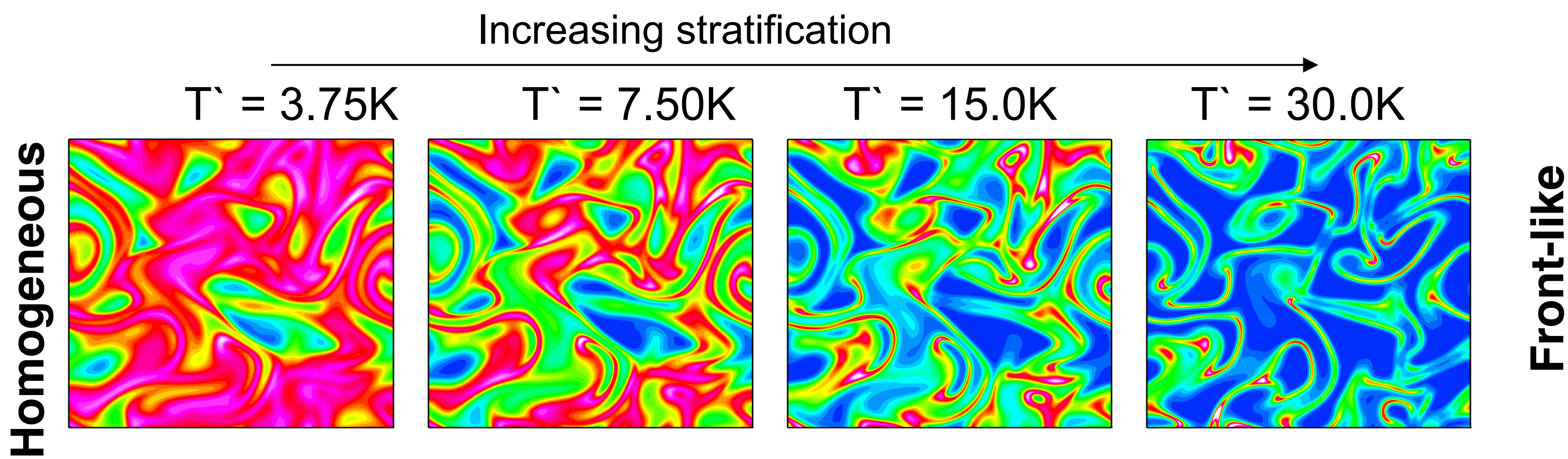


- Direct numerical simulation (DNS)
  - No sub-grid models, but limited on range of scales
  - Simulations limited to canonical research configurations

DNS is hard

- DNS of combustion is highly floating point intensive
- To increase the range of scales captured by a factor ‘X’, the computational work increases by ~X^3
- More time steps needed for better statistics and less dependence on initial condition
- Complex fuels require higher number of equations per grid point
- Device scale simulations are intractable and beyond the reach of DNS

What do we want to simulate on Titan?



Results from a 2D parametric study with hydrogen chemistry (9 chemical species), Chen *et al.* 2003.

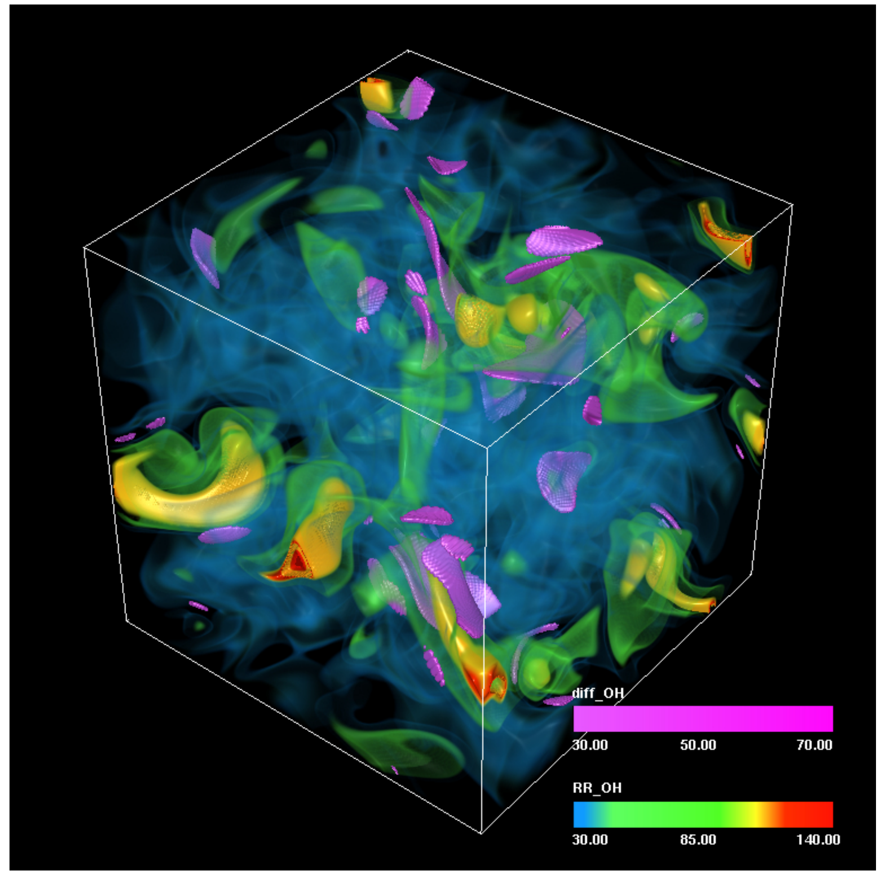
- Objective: 3-dimensional DNS of HCCI combustion in a high-pressure stratified turbulent dimethyl ether (DME) blended iso-octane/air mixture using detailed chemical kinetics (60 chemical species)

  Grid: 2D $O(10^6)$ ➟ 3D $O(10^9)$. Chemical complexity: 9 ➟ 60 species.

- Goals: To investigate
  - Interaction of 3D turbulence with important chemical kinetic pathways leading to ignition
  - Effects of charge stratification on heat release modes, pressure rise rates, and pollutant formation
  - Generate a high-fidelity database for use as a benchmark to validate sub-grid combustion models for mixed-mode combustion in LES and RANS

Planning the science simulation

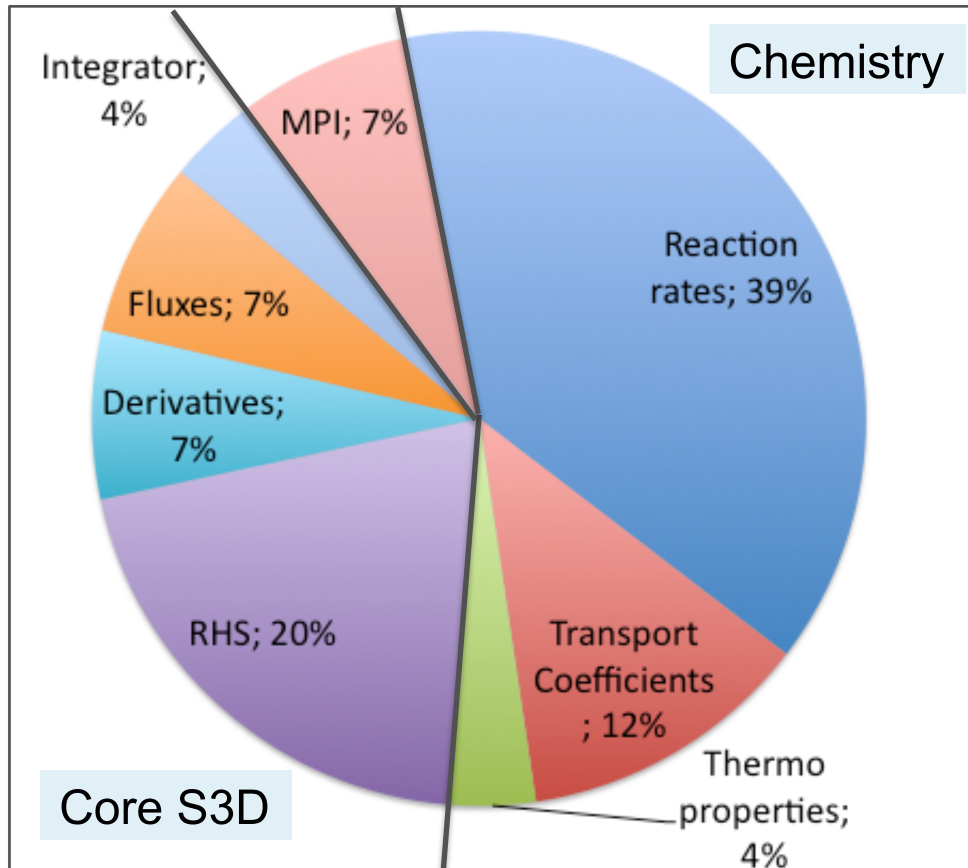
- Recent 3D simulation on Jaguar was used to extrapolate and plan a target Titan simulation
- Planned simulation will have more grid points and/or larger chemistry
- Will need a month on 12,000 hybrid nodes of Titan



Recent 3D DNS of auto-ignition with 30-species DME chemistry (Bansal *et al*. 2011)

Defined a target problem and profiled it

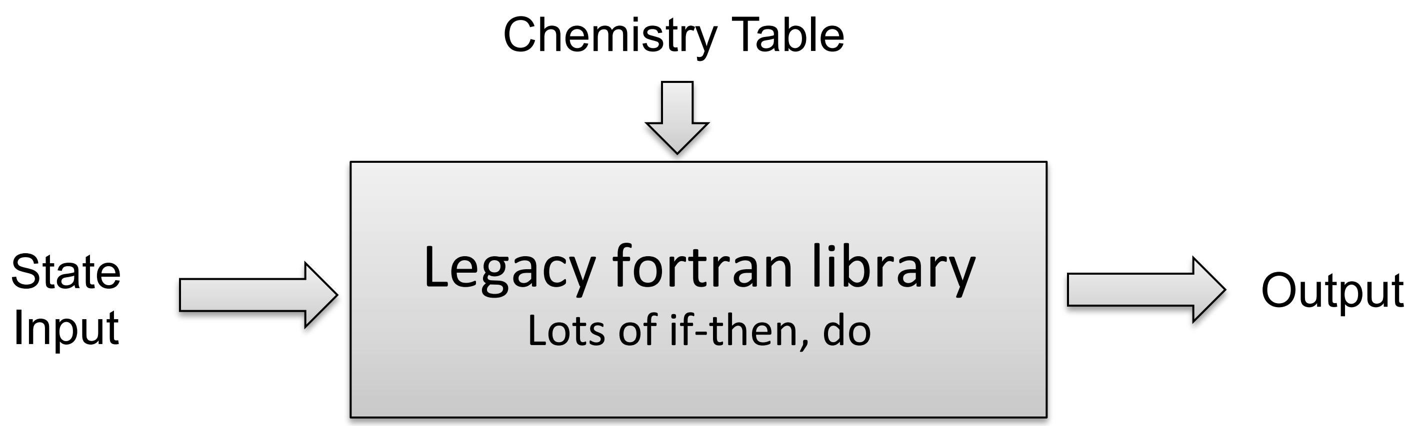
- A benchmark problem was defined to closely resemble the target simulation
  - 52 species n-heptane chemistry and 48^3 grid points per node
  - 48^3 * 12,000 nodes = 1.5 billion grid points
- Code was benchmarked and profiled on dual-hexcore XT5
- Several kernels identified and extracted into stand-alone driver programs

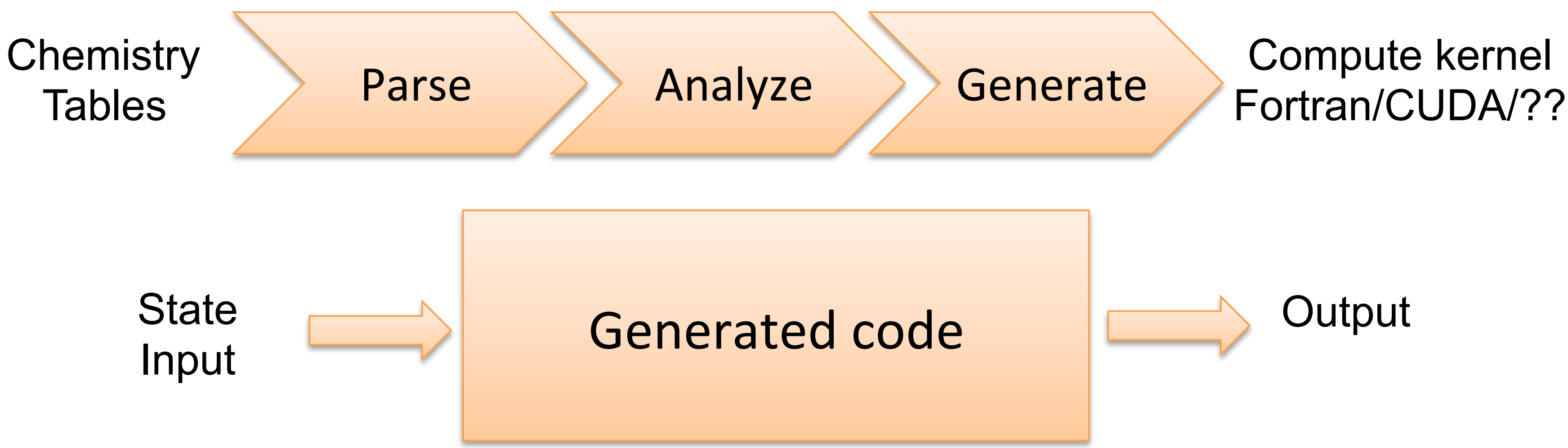


Chemistry Kernels

- Reaction rates, thermodynamic properties and transport coefficients account for 55% of time.
  - Complex chemical kinetic models needed to address multi-stage ignition and flame dynamics
- Point-wise functions that are independent of S3D's mesh data structure and MPI-layer
  - Efforts will pay for a long time irrespective of changes to S3D's algorithms, data structures and solver.
- Used by other combustion codes in the community.
  - Impacts other HPC and workstation-scale combustion applications.

General library vs model specific code



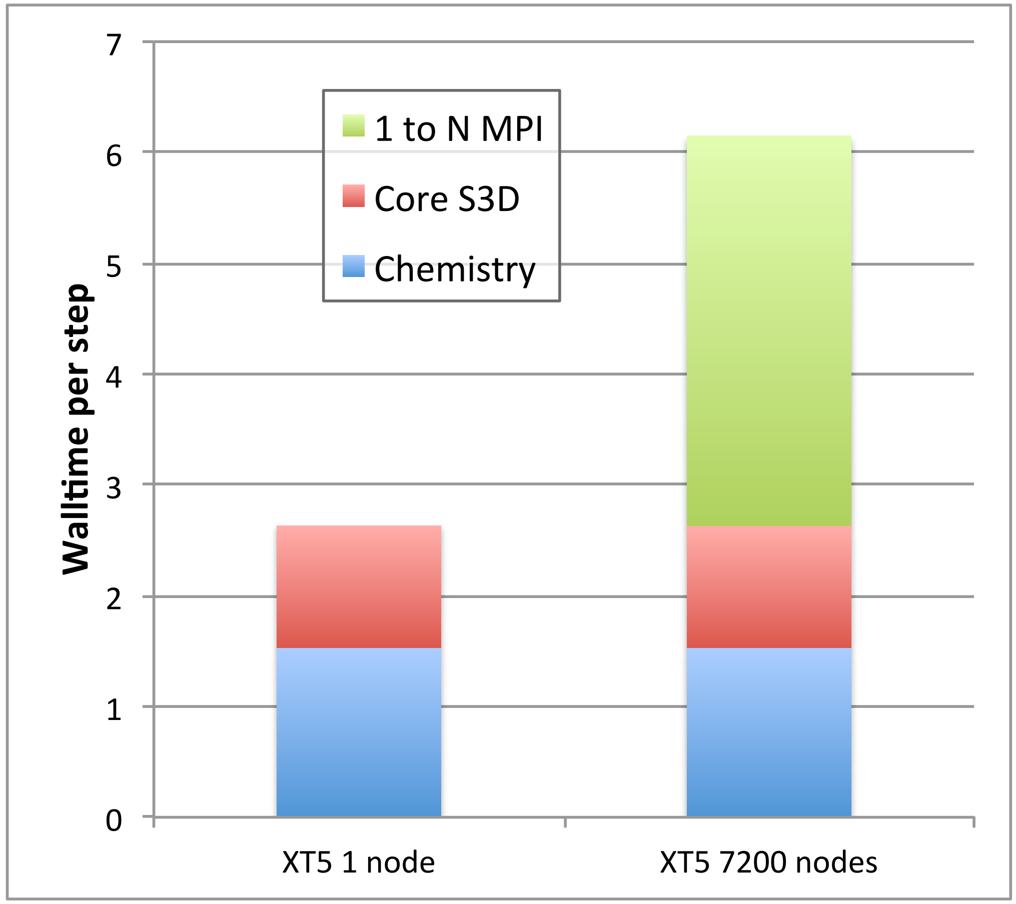


Chemistry Code Generator

- Code generation for the chemistry routines is in vogue
  - Allows non-standard and reduced chemistry models.
  - Branches are evaluated and loops are unrolled at generation time.
  - Model constants are in place avoiding pointer chases.
- Goals of the new code generator
  - Ensure accuracy and correctness for complex chemistry.
  - Ability to generate for newer architectures, programming models.
  - Target a wider audience using much more complex chemistry.
    - Multi-zone type calculations with thousands of species
  - Expose more levels of parallelism
    - Partition and parallelize the reaction network

Need to accelerate more than chemistry

- Amdahl's law: If you don't accelerate everything, soon you won't have accelerated any



Weak scaling of target problem on Jaguar XT5

Core S3D Time Advance Loop

- Flux, derivatives, RHS and integrate account for 40% time
  - d/dt (Q_k) = (Advection) + (Diffusion) + (Source)
  - Evaluation and accumulation of the terms that are on the right hand side of the equation
  - Time integration
- Memory intensive 3, 4 and 5 dimensional loops and arrays
  - Low computation intensity
  - Strided and non-contiguous memory access
  - Representative of other CFD codes
- MPI communication between nearest neighbors in S3D's 3D grid topology for halo exchange

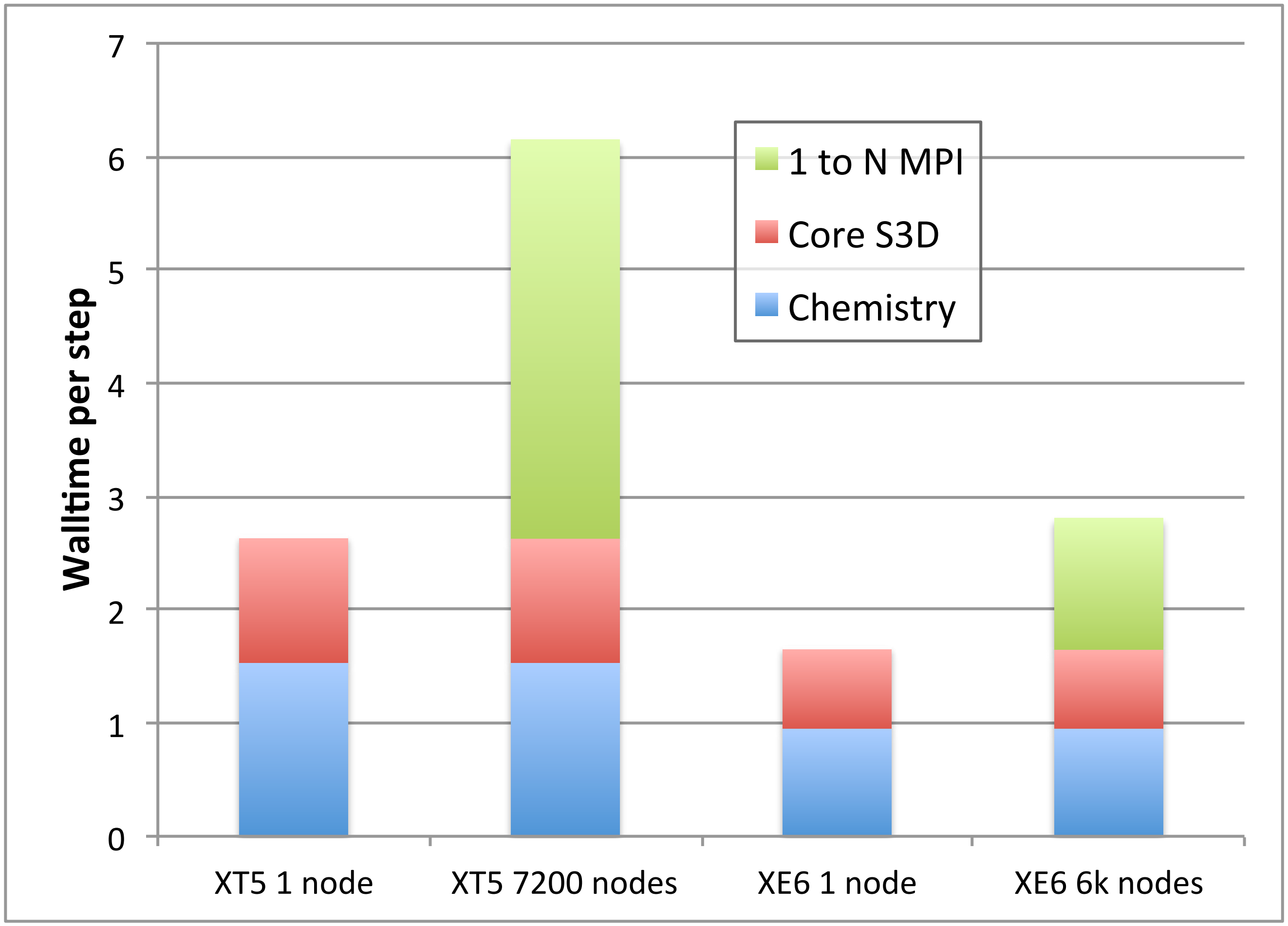
Hybridization of S3D

- Turn S3D from pure MPI code to a hybrid MPI+OpenMP
  - Improve MPI scaling
  - Use OpenMP extensions for porting all of S3D to accelerator

- S3D code revisions
  - Rearranging the compute loops and derivatives to have large compute regions interleaved with communication
  - Variable scoping - distinguishing shared global variables from thread private variables
  - Overlapping computation and communication by identifying opportunities for preposting halo communication

While we were hybridizing S3D

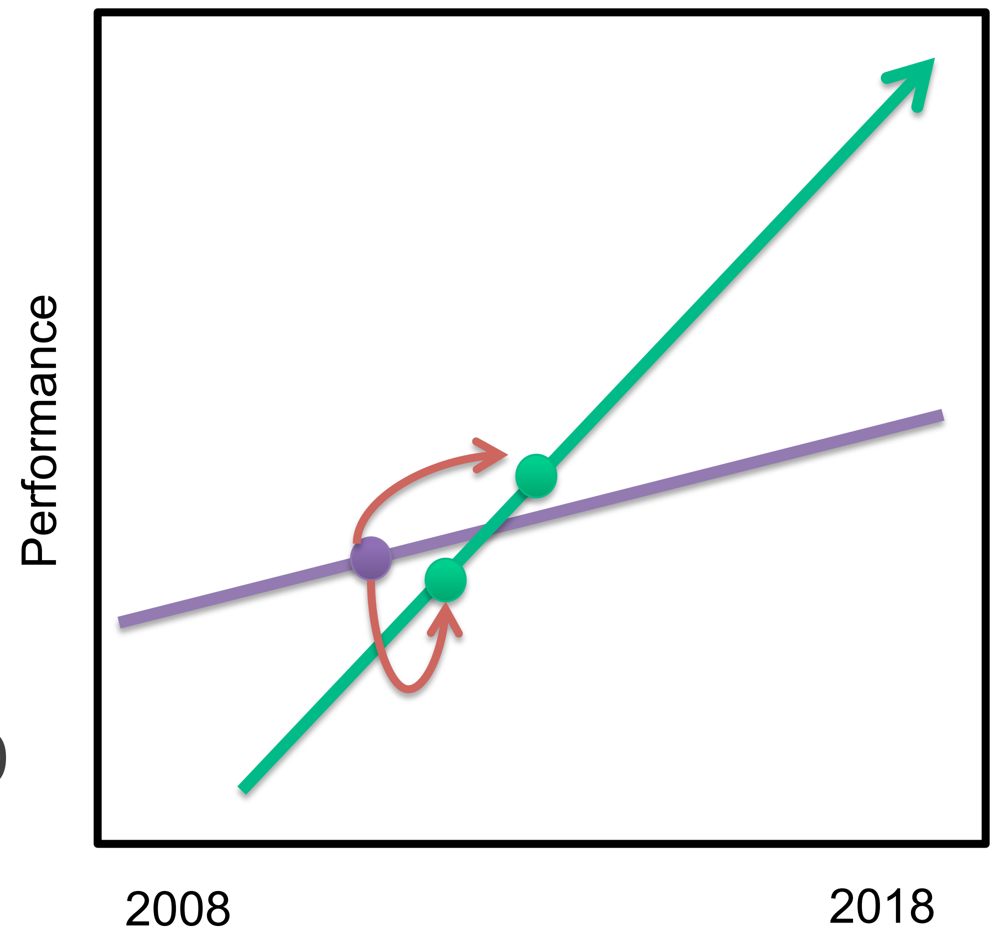
- XE6 came with Gemini interconnect



Scaling of original code (XT5 vs XE6)

Performance of hybrid code

- Currently we cannot beat pure-MPI performance
  - Scales better, but poor in absolute time
- Needs more tuning
  - Memory affinity
  - Placement
  - Optimum number of threads
- Our vision in hybridizing S3D is to be on the right curve for exascale



S3D rewrite is in progress

- Programming models being explored
  - CUDA (Fortran)
  - CCE !$omp directives
  - PGI !$acc directives

- We anticipate 4X better performance on XK6 Titan nodes than XT5 Jaguar
  - Combination of on-node GPU acceleration and improved parallel scaling