Transport Simulations beyond Petascale

Jing Fu (ANL)

A) Project Overview

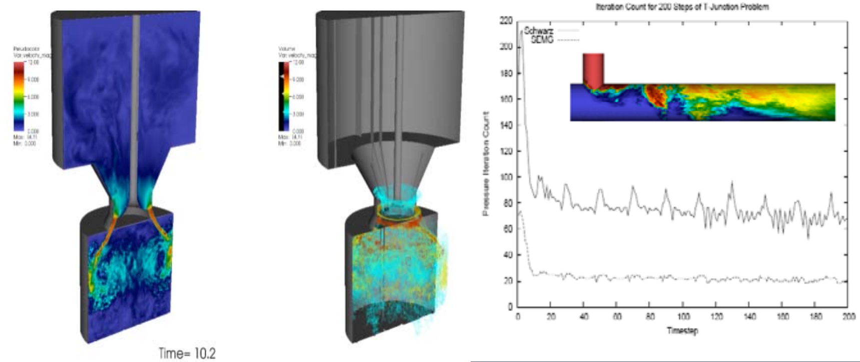
- The project: Peta- and exascale algorithms and software development (petascalable codes: Nek5000, NekCEM, NekLBM)
- Science goals: simulation-based investigations in nuclear energy, combustion, accelerator, and nanoscience applications
- The participants, description of team:

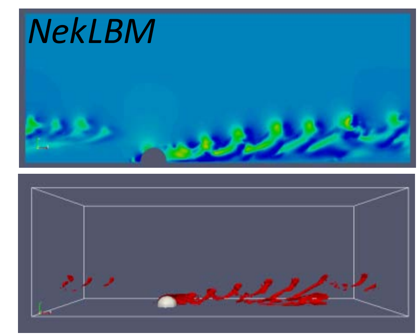
  Paul Fischer, Misun Min, Aleksandr Obabko, Yulia Peet, James Lottes, Katherine Heisey, Jing Fu
- Sponsor: DOE ASCR, DOE NE
- Goals: Development scalable, efficient, and accurate solvers for production runs on future generation architectures targeting billion cores

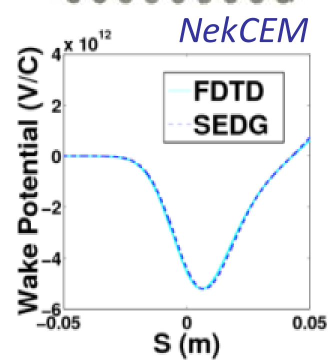
B) Science Lesson

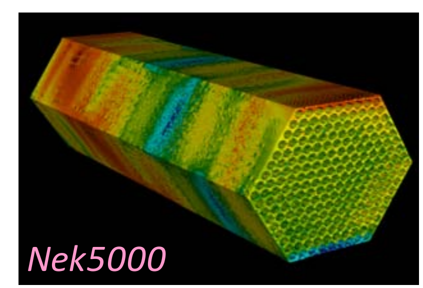
- Nek5000: Fluid dynamics
  - Nuclear reactor
  - Combustion
  - Turbulence
  - Ocean modeling
- NekLBM: Lattice Boltzmann fluid
  - Multiphase flows
- NekCEM: Electromagnetics
  - Accelerator modeling
  - Photovoltaic solar cells
  - Nanophotonic devices

Nek5000









C) Parallel Programming Model

- MPI, Hybrid (MPI +OpenMP/GPU/Pthreads)
- Languages: Fortran and C
- Runtime libraries, build requirements: LAPACK, BLAS, MOAB
- Other infrastructure: python for testing unit tool
- What platforms does the application currently run on?
  - ALCF (BG/P & BG/Q), OLCF (Cray XK6), ANL Fusion
- Current status & future plans for the programming model
  - hybrid: MPI + OpenMP (current speedup 5-10%)
  - hybrid: CPU + GPU (current speedup 20%)
  - To optimize further to reduce communication for data transfer

D) Computational Methods

- What algorithms and math libraries do you use? (PDE, FFT, etc)

  Use spectral element discretizations for solving PDEs
- Current status for your computation

  - Sustained 19% of peak performance on 262,000 cores on Juelich's IBM BG/P Eugene (Nek5000)
  - 90% parallel efficiency for electromagnetics code on BG/P and scalable runs on 262144 cores on Cray XK6 (NekCEM).
  - A new high-order lattice Boltzmann flow solver developed
  - A new high-order Schrodinger solver
- Future plans: Optimize/extend the codes to exascale; Apply for new applications in fuel cells, solar cells, boiling water reactors

E) I/O Patterns and Strategy

- Input I/O and output I/O patterns (one file per MPI process?, pNetCDF? HDF5?, etc)
  - Collective I/O, Reduced-blocking I/O, Threaded reduced-blocking I/O; one or multi-outputs
- Approximate sizes of inputs and outputs
  - Inputs: mesh (vertex data) and vertex connectivity
  - outputs: coordinates, cell connectivity, fields data
- Checkpoint / Restart capabilities: what does it look like?
  - checkpoint: capable of bandwidth 10s of GBs at best
  - restart: re-ordering element in ascending order with swap routine newly built; support restart on any number of cores
- Future plans for I/O: Extension to tetrahedral-mesh based outputs (currently, only for hexahedral-mesh based output)

F) Visualization and Analysis

- How do you explore the data generated?
  - Output in VTK format (ASCII/Binary)
  - Mainly, use Paraview for visualization (Visit is an alternative)
- Do you have a visualization workflow?
  - currently single step run for visualization
  - Further investigation needed for visualization in parallel
- Current status and future plans for your viz and analysis
  - Current visualization with Paraview is in serial
  - Further investigation for data structure changes to reduce the redundant data for each time step in current output

G) Performance

- What tools do you use now to explore performance:
  - Use hpm library on BG/P for performance measurement
- What do you believe is your current bottleneck to better performance?
  - Increase double-hammer instructions for higher peak
- What do you believe is your current bottleneck to better scaling?
  - hybrid programming might help further scaling at exascale
- What features would you like to see in perf tools (ease of instrumentation, different measurements, embedded vis, etc)
  - Different measurements and embedded vis might be helpful
- Current status and future plans for improving performance
  - Double-hammer and hybrid programming are on-going plan

H) Tools

- How do you debug your code?

  - Not intensive use of debugging tools
- What other tools do you use?

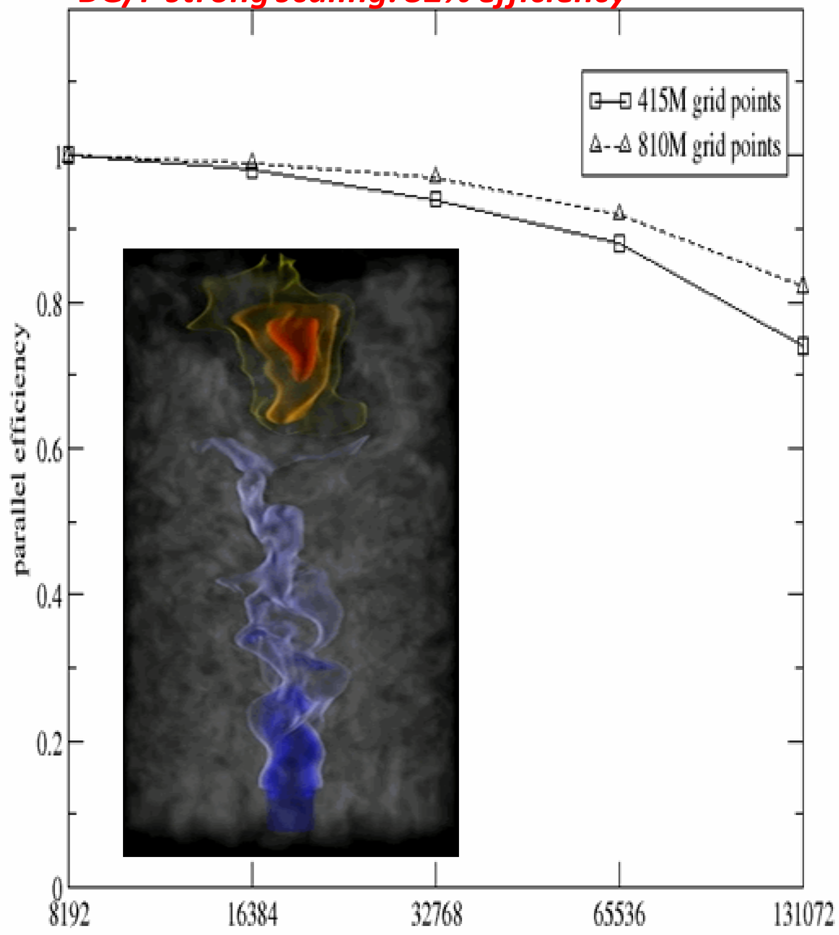
  - Not intensive use of other tools other than performance measure
- Current status and future plans for improved tool integration and support

  - Integration of MOAB (meshing) tool is on-going work.

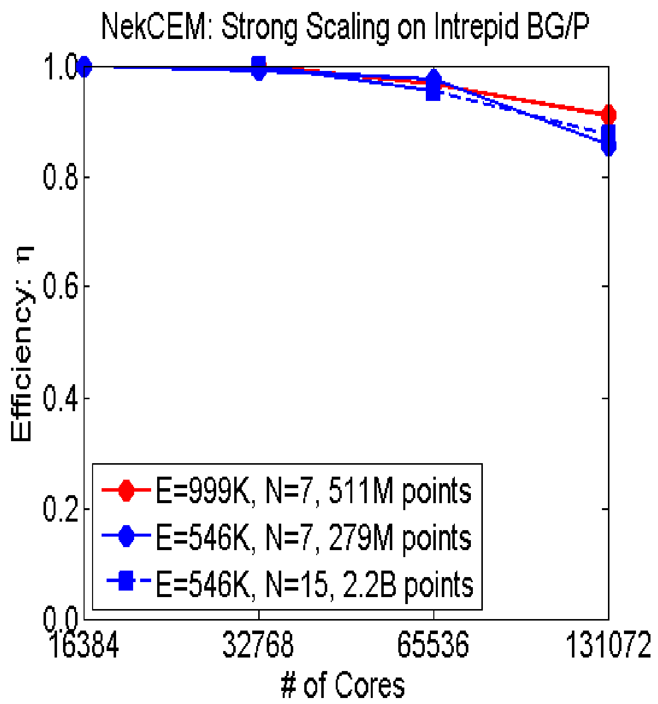
I) Status and Scalability

Nek5000: Turbulent Combustion

BG/P strong scaling: 82% efficiency



Electromagnetics: 90% Efficiency



J) Roadmap

- Where will your science take you over the next 2 years?
  - Continue to develop further in the current applications
  - Expansion to the applications areas of energy efficient fuel cells, solar cells for solar energy harvesting, and nanosensor for molecule detection in medical applications
- What improvements will you need to make (algorithms, I/O, etc)?
  - Peak performance/hybrid programming
  - Performance measure on exascale platforms
- What are your plans?
  - further investigation for advanced numerical algorithms for realistic simulations for exascale-enabled scientific problems