SAND2015-3757C

Mini-PIC
A Next Generation PIC Testbed

Matthew T. Bettencourt[1]

[1]Electromagnetic Theory,

Sandia National Laboratories,

Albuquerque, New Mexico, USA

International Conference on Plasma Science

Antalya Turkey

May 27, 2015

National Nuclear Security Administration

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

SAND 2011-4138 C

Sandia National Laboratories

Outline

- Background
- EMPIRE – ElectroMagnetic Plasma In Radiation Environments
- Mini-PIC
  - Kokkos
- Results
- Summary

Background

- US DoE Advanced Scientific Computing (ASC) is refreshing their major computing resources
  - FY16 Trinity – Self booted Xeon Phi cluster
  - FY18 Sierra – IBM Power with Nvidia chips -150 PF
  - FY21 ATS-3 – In bidding process
- Current software stack cannot take advantage of these platforms
- New program element to develop new applications
  - Advanced Technology Demonstration and Mitigation
- Sandia National Laboratories is developing a new code for plasma simulation EMPIRE
  - ElectroMagnetic Plasma In Radiation Environments

EMPIRE

- EMPIRE is a new open source application being developed for next generation platforms
- Physics and simulation goals –
  - Electromagnetic and electrostatic
  - Kinetic and fluid based plasma descriptions
  - Radiation transport and gas chemistry - DSMC
  - Beyond forward simulation – Sensitivities, UQ, optimization, …
- Computation capabilities –
  - Hybrid structured/unstructured mesh
  - Hybrid PIC-fluid description
  - In-situ meshing, mesh refinement
  - In-situ analysis, visualization and adjoint methods
  - Asynchronous Multi-Tasking (AMT), dynamic balancing

Mini-PIC

- Goal – Develop a reduced physics application to act as a testbed for development of next generation PIC models
- Continuum models in FEM, and solver technologies are being examined by a large number of people
  - Focus on the PIC part of the solve
  - MPI+X
    - Domain Decomposition for MPI
    - Kokkos for cross platform threading
  - Uses unstructured tet or hex elements
  - Developed ES-PIC, adding EM-PIC
    - High order current weighting
  - DSMC under development
  - Being used as a test platform for embedded sensitivity propagation research

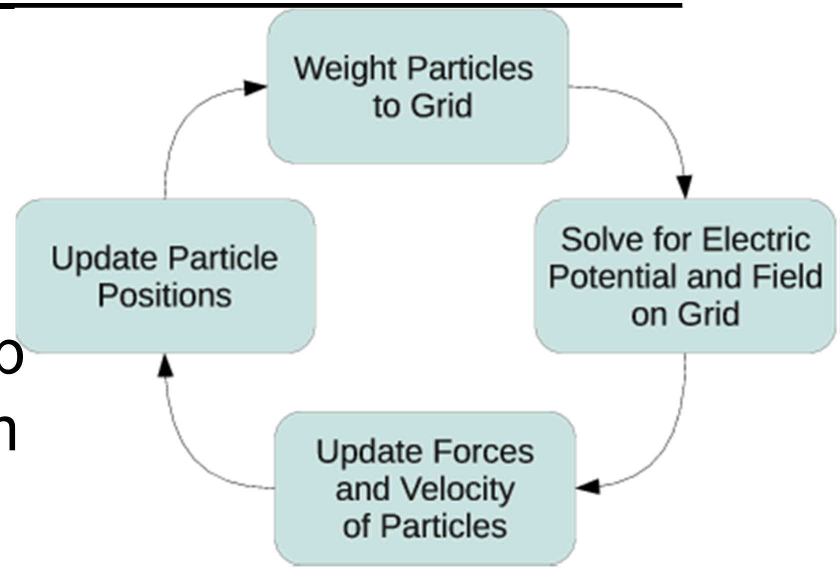
Kokkos: C++ Library / Programming Model for Manycore Performance Portability

- Portable to Advanced Manycore Architectures
  - Multicore CPU, NVidia GPU, Intel Xeon Phi
    - Backends – Cuda, OpenMP, pthreads and serial
  - Maximize amount of user (application/library) code that can be compiled without modification and run on these architectures
  - Minimize amount of architecture-specific knowledge that a user is required to have
  - Allow architecture-specific tuning to easily co-exist
  - Requires a C++11 compiler
- Performant
  - Portable user code performs as well as architecture-specific code
  - Thread scalable – not just thread safety (no locking!)
- Usable
  - Small, straight-forward application programmer interface (API)
  - Constraint: don't compromise portability and performance

Kokkos: Collection of Libraries

- Core – lowest level portability layer
  - Portable data-parallel dispatch: parallel_for, parallel_reduce, parallel_scan
  - Multidimensional arrays with device-polymorphic layout for transparent and device-optimal memory access patterns
  - Access to hardware atomic operations
- Containers – built on core arrays
  - UnorderedMap – fast find and thread scalable insertion
  - Vector – subset of std::vector functionality to ease porting
  - Compress Row Storage (CRS) graph
- Linear Algebra
  - Sparse matrices and linear algebra operations
  - Wrappers to vendors' libraries
  - Portability layer for Trilinos manycore solvers
  - Trilinos/Tpetra built on top of Kokkos for MPI+X linear algebra

Time Update



- Mini-PIC uses the standard operator split, leap-frog, time update scheme
- Elliptic solve uses a Bi-CGStab which extracts parallelism from Tpetra
- Particle position update requires element by element tracking
    - Needed for EM current deposition and thin feature detection
    - Iterative for hex elements, analytic for tets
- Weighting algorithms can cause conflicts

Particle Movement

- Particles are moved in “reference space”
  - Lower round-off, simpler conditionals
- On tet (simplex) mesh elements this requires a multiplication by the element Jacobian matrix
- On hex mesh elements, elements are non-affine and therefore one needs to iterate
  - Jacobian matrix is a function of position within the element
  - Even if velocity is constant within an element, reference velocity may not be, straight physical lines are curved in reference space
  - Iterate reference velocity until starting and ending points in physical and reference space coincide
    - Reference velocity represents transformed physical velocity at a point along the trajectory.
    - Reference velocity can be used to weight current

Particle Migration

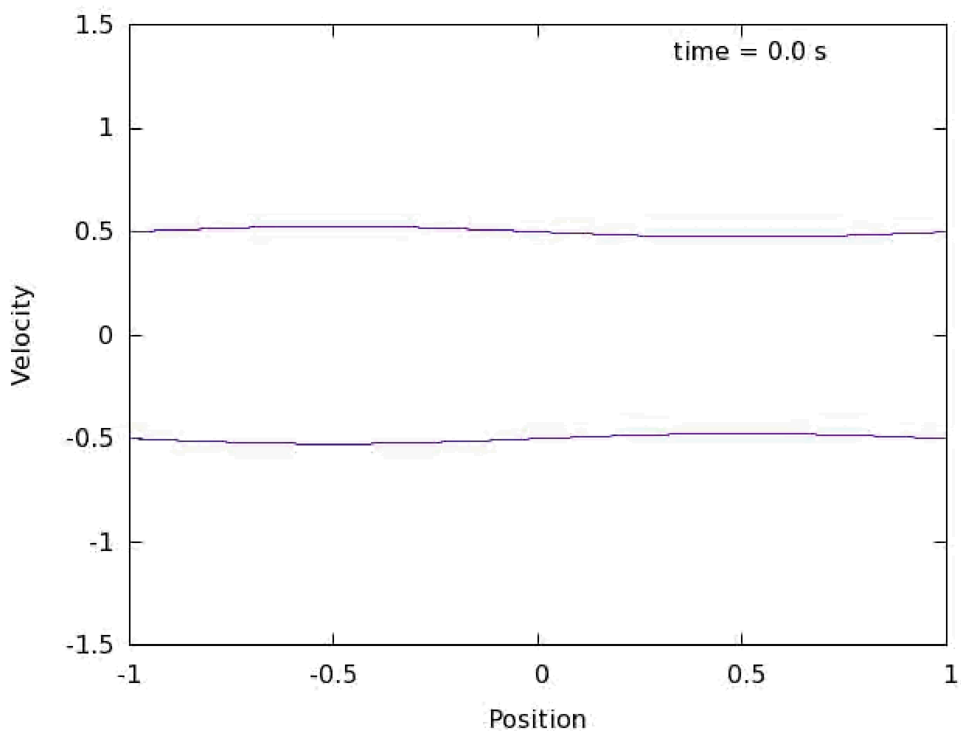
- At a high level particles are pushed to the end of their timestep or until they hit a processor boundary
  - Particles are packed into migrate buffers and migrated to neighboring processors
  - Particles are unpacked on destination processor and the remaining of the push is conducted
  - This is repeated until all the particles reach their final location
  - MPI calls are done in the host space, copied to/from device space

Charge Weighting

- Weighting charge in a threaded environment can result in data write conflicts
  - Atomic operations are typically used in summations
    - Lots of conflicts and poor performance on Cuda
  - Alternative – sort the particles by element and sum into local variables
    - Fewer atomic operations
    - Sort can be expensive
    - Not extendable to EM solve and currents due to element crossing
    - Only practical on Nvidia, sorting too slow on other platforms

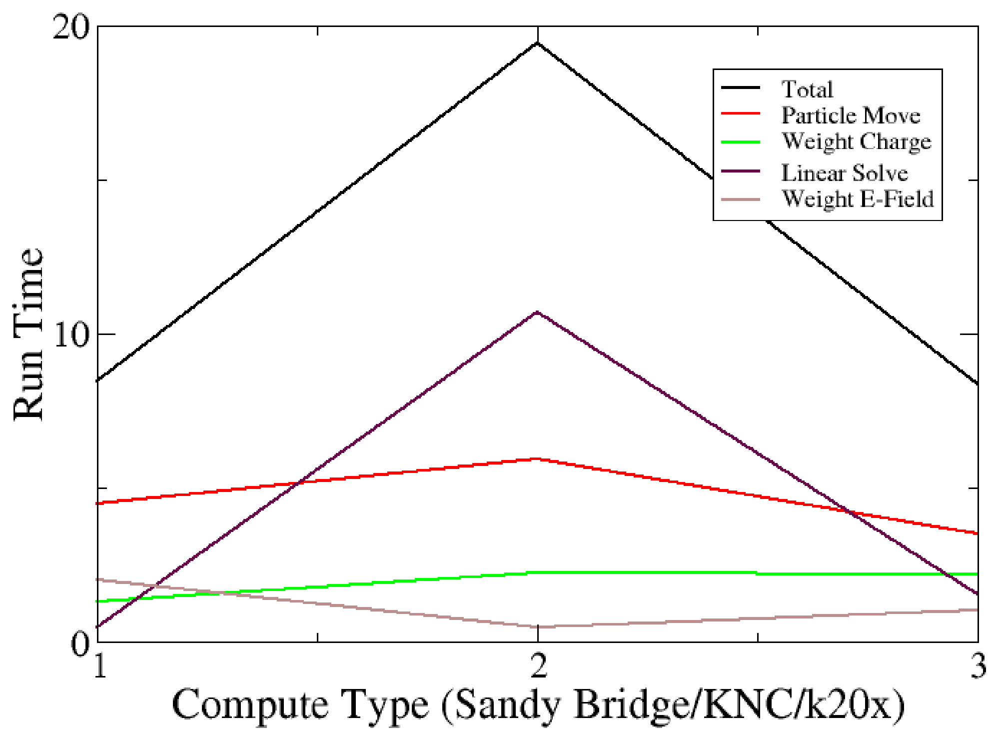
Results

- Two-stream instability
  - In a 3 dimensional singly periodic box with Dirichlet conditions on remaining four walls
  - 655k (dx ~ .03) DOFs and 262M particles (~400/element)



Portable Performance

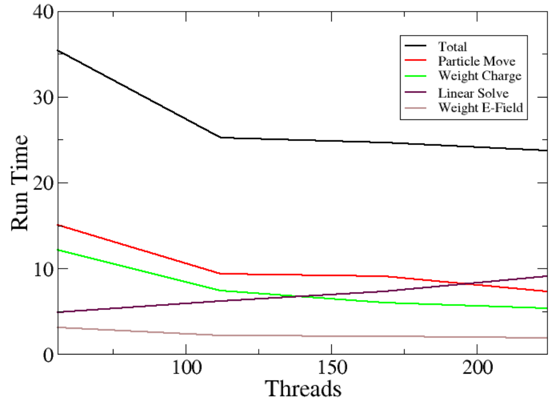
- Computed time per node on several machines
  - 18k unknowns and 11M particles, 20 timesteps CFL = 1.5
  - 16 cores of Sandy Bridge – 2 Knights Corner – 1NVidia k20x



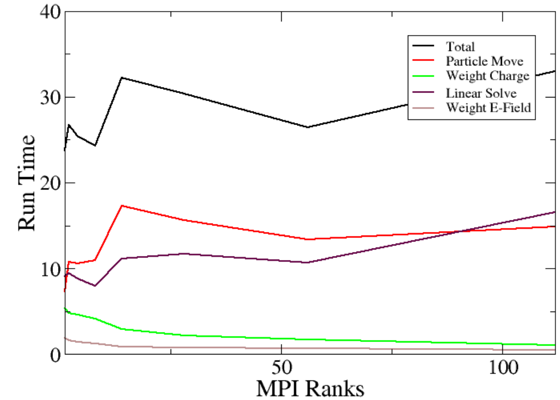
MPI vs Threads

- Comparisons of distributions between threads and MPI ranks

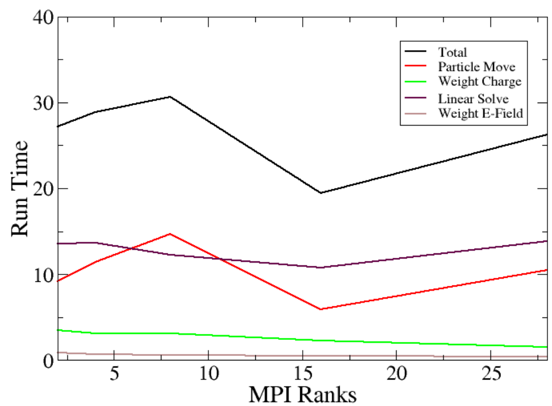
KNC 1 - 4 Hyperthreads



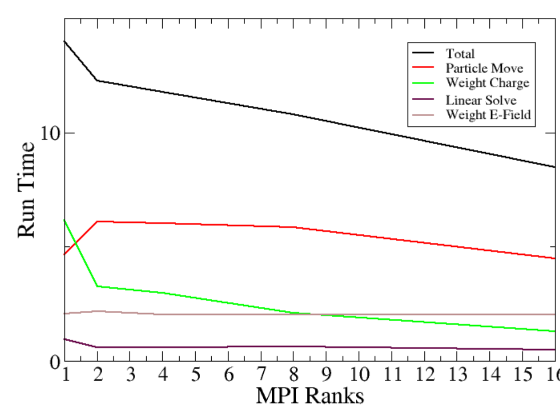
KNC 224 threads 1 -112 MPI ranks



2 KNC 456 threads 2 - 28 MPI ranks



Sandy Bridge 1-16 MPI ranks



Summary

- Open source MPI+X ES-PIC code available for collaboration
- Initial step towards full featured open source EM-Plasma tool
- Single code base – multiple execution spaces allows for cross platform computing
  - Algorithmic tuning still required for specific back ends
  - MPI only not sufficient for next generation platforms