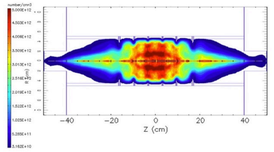
Heterogeneous Plasma Physics Codes in TriForce: Progress and Next Steps

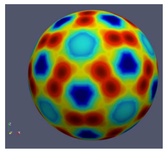
M. Burns, R. K. Follett, A. Bowman, S. Zhai, A. Poudel , S. Dwarkadas, S. Pai, A. B. Sefkow

The TriForce Center for Multiphysics Modeling, a collaboration between the Departments of Mechanical Engineering, Physics, Computer Science, and the Laboratory for Laser Energetics

Summary

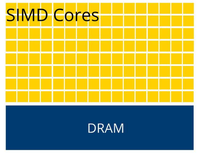
- Plasma code development targeted at high-performance heterogeneous platforms
- Focused on CUDA GPU paradigm
- Primary algorithms: Particle-in-Cell (PIC) and iterative Cross Beam Energy Transfer (CBET)
- Multi-GPU clusters supported, targeting multi-node systems
- Focus on large spatial-scale simulations

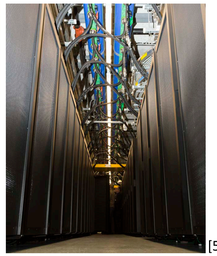


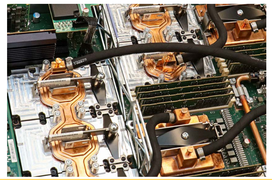


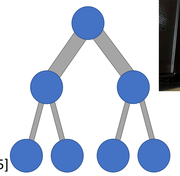
Motivation

- Algorithms applicable to wide range of plasma phenomena
  - Inertial Confinement Fusion
  - Magnetic Confinement Fusion
  - MHD
- Reduce simulation computation overhead, allow for more rapid simulation and experimentation
- Better leveraging available hardware in modern high-performance computing (HPC) centers
  - ORNL Summit (#2 on Top500): 6 GPUs per node [5]
  - ANL Aurora (Upcoming Exascale): 6 GPUs per node [6]
  - ORNL Frontier (Upcoming Exascale): 4 GPUs per node [7]
- Exploiting parallel aspects of algorithms to a larger extent
- Leverage multi-node computation for larger problem sizes (MPI)
- Goal: Understanding best computing practices for heterogeneous code development



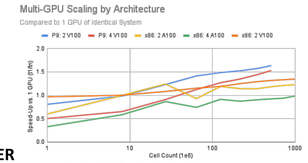

[5]

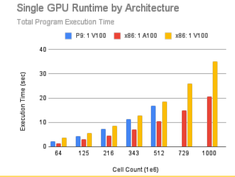

[5]



Heterogeneous Ray Tracing

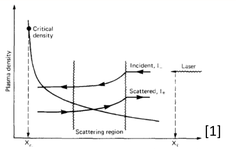
- Subroutine of CBET algorithm
- Traces ray paths across plasma
- Each thread assigned subset of rays
- Optimized for GPU execution
  - Data locality: Ray reindexing,
  - Minimal Global Memory Accesses
- Operational on multi-GPU nodes
  - Portable across architectures (POWER and x86)
  - Distributes problem by ray information
    - No cross-talk necessary
- Currently investigating:
  - Data movement optimizations
  - Unified Memory performance
  - Optimized pipeline utilization



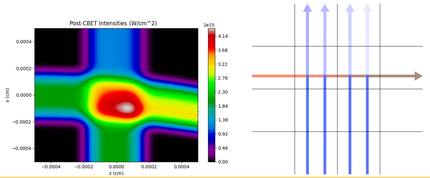


CBET Algorithm

- Calculating cross beam energy transfer across expanding plasma
  - Based upon stimulated Brillioun scattering [1]
  - Dependent on laser and plasma characteristics
  - Reduces ICF implosion efficiency
- Simulation
  - 1-step, iterative, explicit algorithm [2]
  - Ray-based modeling approach
  - Distribute rays amongst threads of computation
    - Map thread groups to warps
    - Minimize synchronization

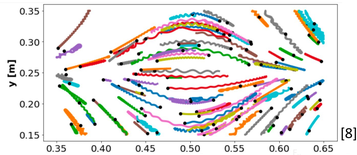


[1]

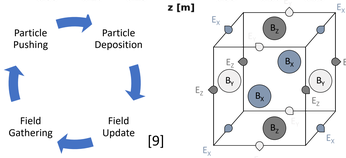


PIC Algorithm

- Kinetic model of plasma
  - Coalesce particles = "Macroparticles"
  - Break computation into discrete stages:
    - Field gathering, particle push, current deposition, field solution
  - Applicable to wide variety of systems (Poisson, Maxwell)
  - Execution time dominated by particle push
- Challenge
  - Optimize for heterogeneous architectures [3]
    - Maximize data locality
    - Mixed precision computation
    - Minimal branching
  - Minimize MPI overhead in multi-cluster systems [4]



[8]

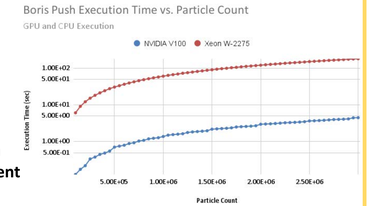


[9]

[9]

Heterogeneous Particle Push

- Subroutine of kinetic algorithms
  - Developed for PIC
  - Focuses on EM interactions
- Utilizes Boris push algorithm
  - Splits $\vec{E}$ and $\vec{B}$ momentum contributions
  - "Half time steps"
- Requires non-local access to field data
- Data migration necessary for subsequent PIC phases
  - Motivates fully heterogeneous code, maximize concurrency between host and device



Current State

- CBET
  - 2D Code: Verified and functional
  - 3D Code: Functional, undergoing verification
  - Built as a standalone program
- PIC
  - Particle pusher on GPU
  - FDTD on CPU
  - Currently optimizing on GPU

Next Steps

- CBET:
  - Add field deposition
  - Continued optimization
  - Integrate into wider TriForce code
- PIC:
  - Quantify TP of various push algorithms
  - Identify host- and device-suited phases
  - Implement as multi-GPU, multi-cluster code

Related Work

- [1] SBS Theory
- [2] CBET Algorithm: Governing equations and comparison to preexisting models
- [3] PIC: nGPU-CUDA Implementation: Single node, multi-accelerator
- [4] PIC: MPI w/ AMR: Multi-cluster, adds adaptive mesh refinement capabilities

References

- [1] C. Randall, et. al, The Physics of Fluids, vol. 24,no. 8, pp. 1474–1484, 1981.
- [2] R. Follett, et. al, Physical Review E, vol. 98, no. 4,p. 043202, 2018
- [3] S. W. D. Chien, et. al, *2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2020, pp. 149-156, doi: 10.1109/SBAC-PAD49847.2020.00030.
- [4] T. Matsui, H. Usui, T. Moritaka and M. Nunami, *2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing*, 2011, pp. 277-281, doi: 10.1109/PDP.2011.96.
- [5] "Summit". *Oak Ridge Leadership Computing Facility,* https://www.olcf.ornl.gov/summit/. Accessed 10/22/2021.
- [6] "Aurora", *Argonne Leadership Computing Facility.* https://www.alcf.anl.gov/aurora. Accessed 10/22/2021.
- [7] "Frontier", *Oak Ridge Leadership Computing Facility,* https://www.olcf.ornl.gov/frontier. Accessed 10/22/2021
- [8] Mike Lavell, private communication
- [9] Ayden Kish, private communication

Acknowledgments

This material is based upon work supported by the United States Department of Energy OFES under Award No. DE-SC0017951