Framework for advanced plasma simulations on GPU and many-core HPC Clusters

Aeronautics and Astronautics
University *of* Washington

CPD
Computational Plasma Dynamics Lab

Noah Reddell, Uri Shumlak: University of Washington, Guoyong Fu: Princeton Plasma Physics Laboratory

Introduction:

We have created a fluid modeling code called WARPM to utilize modern many-core compute devices – namely GPUs.

Fusion plasma simulation has much in common with computational fluid dynamics (CFD), but adds the complexities of modeling the electromagnetic interactions of the charged plasma species, confinement fields, and heating fields. Modeling involves elaborate codes with computational demands that limit achievable approximation of real-world behavior.

WARPM is designed to both minimize data movement and maximize data-parallel computation. The code is a hybrid combination of OpenCL for parallel computation, MPI for communication between nodes, and threads for task-parallelism. WARPM uses Message Passing Interface (MPI) to distribute computation across multiple nodes of a cluster and for parallel file I/O. The OpenCL standard is central to the new code. GPU(s) and/or multi-core CPUs are utilized simultaneously to compute updates to the system of fluid equations.

WARPM

- •Washington Approximate Riemann Plasma code – many-core edition
- •C++ development: Relies heavily on object-oriented structures and templates
- •Uses coordinated combination of OpenCL, threads, and MPI for different levels of parallelism.
- •Primarily uses Finite Volume method for hyperbolic problems on structured grids.
- •WARPM is innovative in its flexibility to solve an arbitrarily sized system of equations using solvers specified at run-time, using GPUs and other many-core devices.
- •Targets heterogeneous computational resource and multi-level domain decomposition
- •A "typical" simulation might use ~128 nodes for ~1000 core hours.

Example System of two-fluid plasma hyperbolic equations:
- •electron and ion continuity
- •electron and ion momentum
- •electron and ion state
- •electron and ion pressure
- •Faraday's Law
- •Ampere's Law
- •current density definition

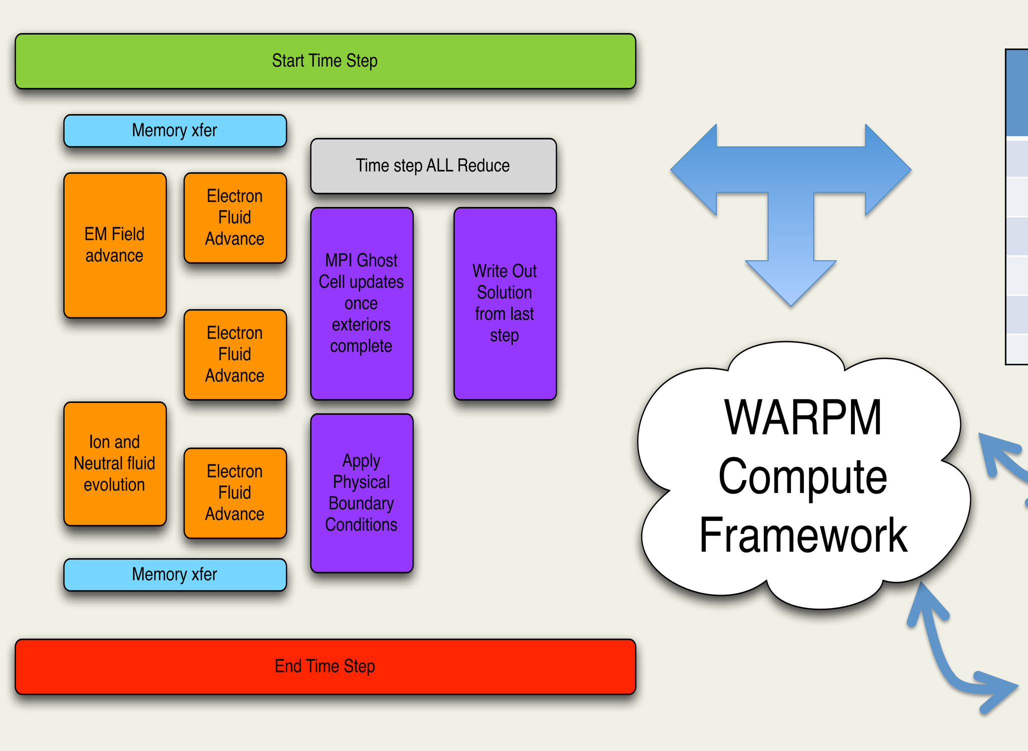
In total, 11 equations (incl. 5 vector equations)

Uses OpenCL Language

Inspired by roadmaps to many-core CPUs, rapid gains in GPU compute capability, and the success of NVIDIA's CUDA language for general purpose GPU computing, a consortium of software and hardware companies formed within the Khronos Group. The group ratified an open language specification that allows applications to be developed for heterogeneous multi-core platforms using common source code. Specification ver. 1.0 implementations started emerging in the Fall of 2009.

- •Virtually every leading company in the HPC sector has an active role in the development of OpenCL. (NVIDIA, Intel, AMD/ATI, IBM) The language is likely to be enduring and does not tie your code to a single platform.
- •OpenCL is powerful in that it exposes the developer to architecture influences, without being too low-level. This is not to say it's easy, just an ok balance of complexity and performance.
- •OpenCL is a data parallel language well-formulated to support many scientific HPC applications.
- •OpenCL is both a C-based language for kernels and an API for managing those kernels and compute resources.

WARPM Framework orchestrates computational kernels, memory movement, and disk i/o based on user specified dependencies



task <id>	Steps to apply	List of Dependencies	List of buffer resources
<1>	Riemann.eq0()	<0>	[qold, apdq]
<2>	Riemann.eq1()	<1>	[qold, apdq]
<3>	RiemannT.eq0()	<1>,<2>	[apdq, bpapdq]
<4>	RiemannT.eq1()	<1>,<2>	[apdq, bmapdq]
<5>	Godunov Update	<3>,<4>	[qold, qnew]
⋮	⋮	⋮	⋮

Users supply the computational model and evaluation sequence.

Performance and Energy Impact of Data Movement

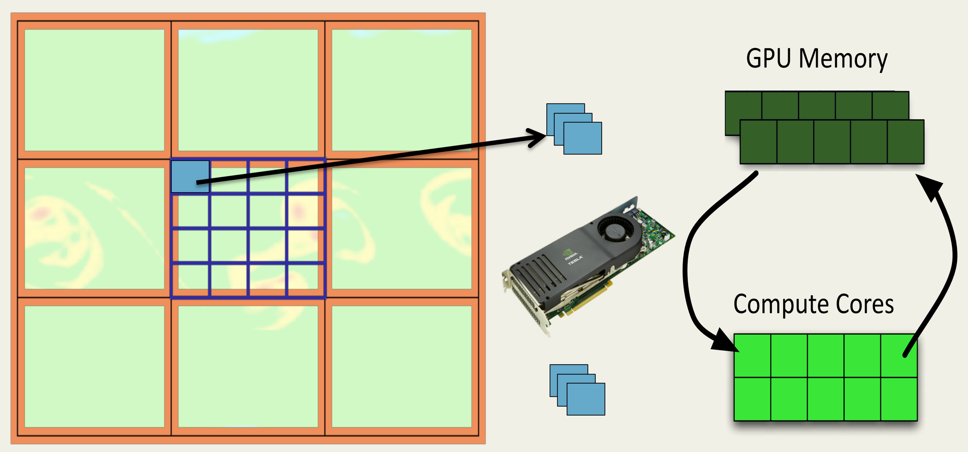
Current Hardware Performance Illustrates Need for Careful Consideration for Data Movement.

- PCIe x16 – 8 GB/s host to GPU
- Using NVIDIA Fermi M2050 specs:
- Peak performance: 515 Gflop/s DP
- 515 Gflop/s ÷ 8 GB/s × 8 bytes/double
  - 515 floating point operations needed per operand transferred between GPU and host to hide PCIe bottleneck
- Memory bandwidth: 148 GB/s to cores
  - 28 floating point operations needed per operand transferred between GPU GRAM and core to hide memory interface bottleneck
- Energy requirements for a FLOP around 50 pJ.
- But moving data and other overhead accounts for about 1000-10000 pJ per FLOP.

In Future Computing Roadmaps, this problem only becomes more severe.

Designed to Minimize and Hide Data Movement

- Working with block structured grids, data space resident on one node is further subdivide into patches.
- Patches are small enough for multiple to fit in GPU memory so that GPU-Host memory transfer can overlap with GPU computation.
- Patch processing is sequenced so that exterior patches on a node are computed first. MPI ghost cell updates can then occur in parallel with interior patch computation – hiding the MPI communication.
- All computational steps are grouped into a single kernel so that data need only be transferred from GPU memory to the core once per time step.



Performance Improvement:

- Initial performance testing conducted comparing WARPM to single-threaded WARPX code performance for 2D wave propagation on a single node.
- WARPM achieves twice the computational performance of WARPX on the same machine (12-core CPU execution)
- Performance gains due to primarily to eliminating MPI overhead on same node and establishing predictable compile-time execution sequence.
- Performance improvement margin goes up with more complex fluid problems because these are more numerically intense. (i.e. more floating point operations per data element) This better utilizes the GPU capability.

Physics Study: Energetic-Particle-Induced Geodesic Acoustic Mode in Tokamaks

Energetic particles such as those produced by neutral beam injection can excite a mode similar to Geodesic Acoustic Mode via free energy associated with velocity space gradient in energetic particle distribution.

EGAM has been demonstrated by Particle-In-Cell simulations.[2] We develop a continuum kinetic model that is more suitable for our many-core computing framework and future architectures. Simulations aim to reproduce behavior demonstrated by PIC codes including non-linear 'bursting' in the radial electric field.

Physical Model

- •Drift-kinetic treatment with static magnetic field.
- •Delta-f method utilized for hot particles' deviation from an initial single streaming distribution.

$$f_h(R,Z,v_\parallel) = f_{h0} + \delta f$$

- •Four coupled PDE system as seen below

$$\frac{\partial \delta f}{\partial t} = -(mv_\parallel^2 + \frac{1}{2}mv_\perp^2)\frac{\partial f_0}{\partial E}G(r,\theta)E_r - v_\parallel \vec{b}\cdot\nabla\delta f - \frac{R}{B_0R_0}E_r\left(\sin\theta\frac{\partial\delta f}{\partial R} - \cos\theta\frac{\partial \delta f}{\partial Z}\right)$$
$$-\vec{v}_d\cdot\nabla\delta f - \left(\frac{v_\parallel}{B_0R_0}\sin\theta E_r + \frac{\mu B_0}{2mR_0^2 q}Z\right)\frac{\partial\delta f}{\partial v_\parallel}$$

$$(\delta P_\parallel + \delta P_\perp)(R,Z) = \int m(v_\parallel^2 + \frac{1}{2}v_\perp^2)\delta f d^3v$$

$$\frac{\partial E_r}{\partial t} = -\frac{\left\langle G(r,\theta)\left(\delta P_{\mathrm{th}} + \delta P_\parallel + \delta P_\perp\right)\right\rangle}{\left\langle \frac{\rho|\nabla r|^2}{B^2}\right\rangle}$$

$$\frac{\partial \delta P_{\mathrm{th}}}{\partial t} = 2\gamma G(r,\theta)P_{\mathrm{th}}^{eq}(r)E_r(r)$$

Computational Model

- •Finite Difference Formulation
- •RK2 Time Stepping
- •Two spatial dimensions and one velocity space, v_{par}, with magnetic moment μ constant.
- •Hot Particle delta-f represented on a Cartesian R-Z-v_{par} grid
- •Thermal Pressure represented on a polar r-θ grid.
- •Typical domain dimensions: 60 elements in each physical dimension, 100 elements in velocity space

Simulated Geometry

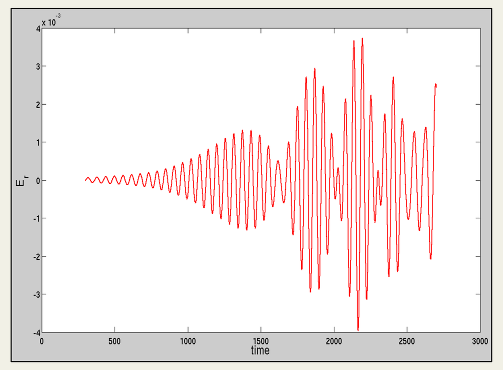
- •Circular tokamak cross-section (No aspect ratio effects)
- •Periodic in toroidal dimension.
- •Velocity space uniform and typically 100 elements

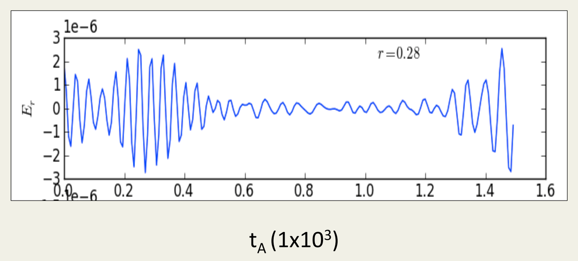
References

[1] U. Shumlak, et al. Advanced physics calculations using a multi-fluid plasma model. Computer Physics Communications, 182 (9). 2010.
[2] G.Y. Fu, Energetic-Particle-Induced Geodesic Acoustic Mode. Physical Review Letters, 101. 2008.
[3] G.Y. Fu, Nonlinear Simulation of Energetic Particle-induced Geodesic Acoustic Mode, presentation at The 11th IAEA Technical Meeting on Energetic Particles in Magnetic Confinement Systems, September 21-23, 2009, kiev, Ukraine

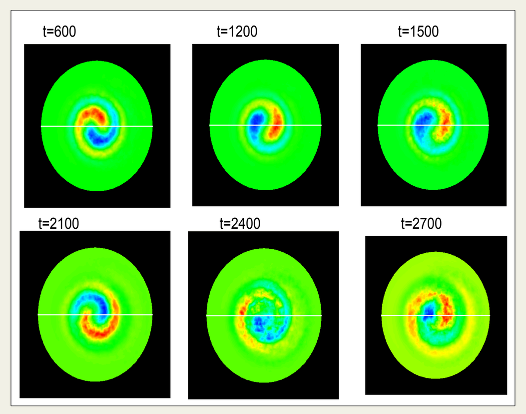
Original PIC Results[3]: WARPM Continuous Kinetic Results:

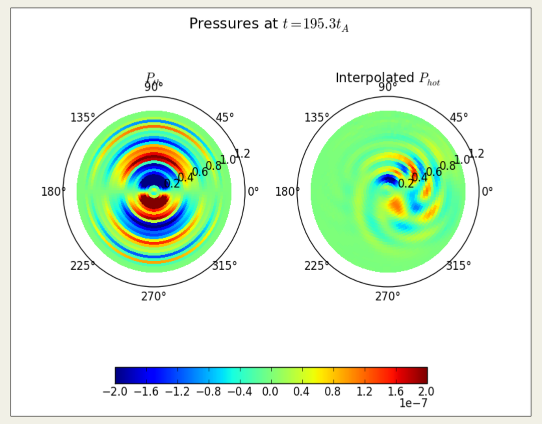
Electric field dynamics time-series





Nonlinear evolution of perturbed hot particle pressure





Summary and Upcoming Work

- •WARPM has proven to be an effective new scientific computing framework that is well situated for emerging computing architectures. Development will continue.
- •WARPM provides a framework that can be readily utilized by users to solve their own computational models.
- •Initial EGAM simulations in continuum kinetic formulation show some similar dynamics to PIC code results, but parameters to give non-linear bursting still need to be identified.
- •For physical investigation, we have a priority support general geometry promblems.
- •We are working to create kernels that are dynamically assembled at run-time based on the specific physics problem and numerical method chosen by the user.

Acknowledgements

I am very grateful for the advice, shared experience, and tutoring from the developers who have gone before me: Bhuvana Srinivason and Robert Lilly and my advisor Uri Shumlak, in which I could not have found a better mentor.
Much credit is also due to Dr. Fu who guided me through a three month internship at PPPL studying EGAM.
This work is possible through support of the DOE Computational Science Graduate Fellowship program.

DOE CSGF