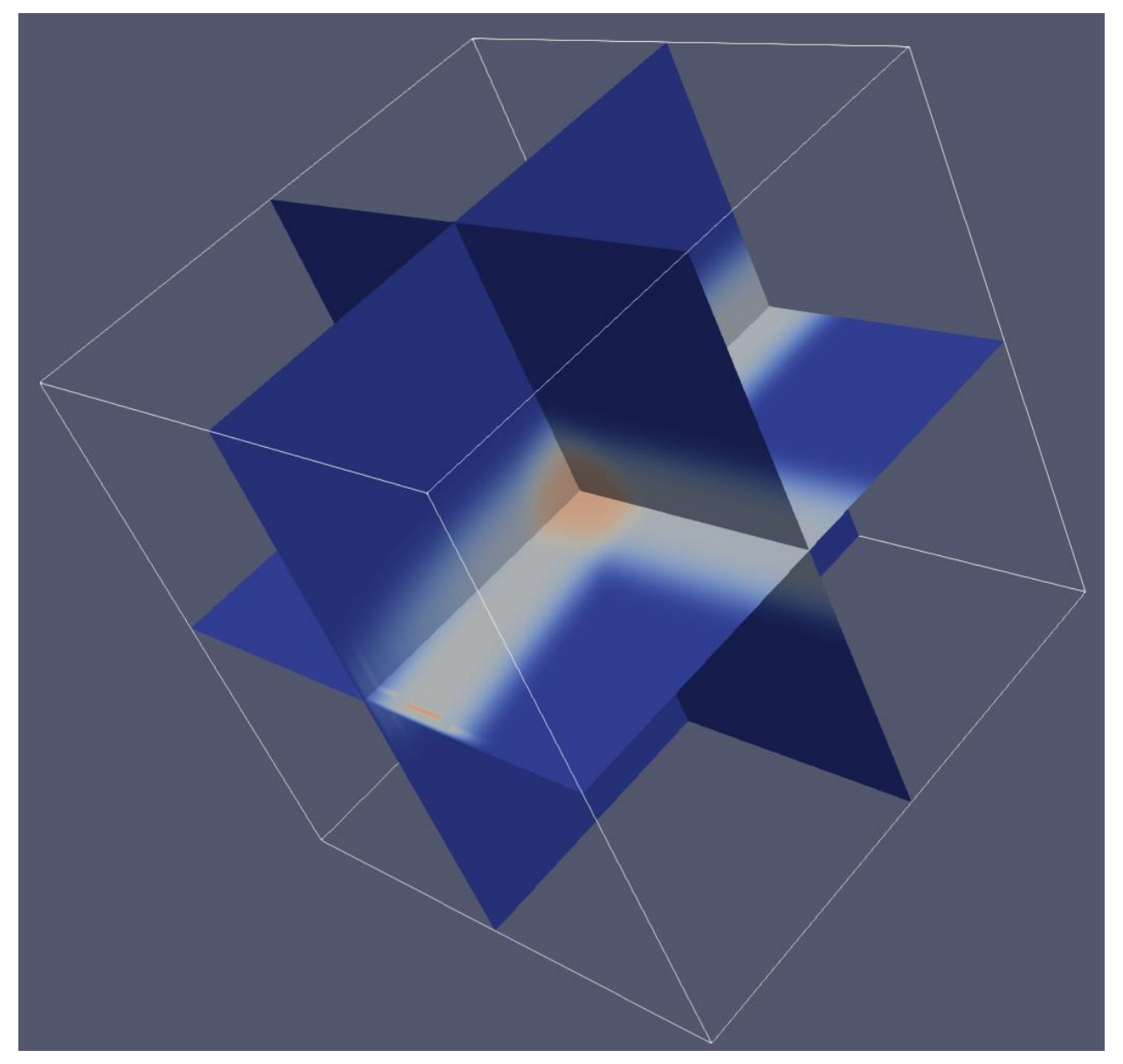
GPU Acceleration for a 3D Simulation of CBET

WILLOW VEYTSMAN, SHUANG ZHAI, ADAM B. SEFKOW

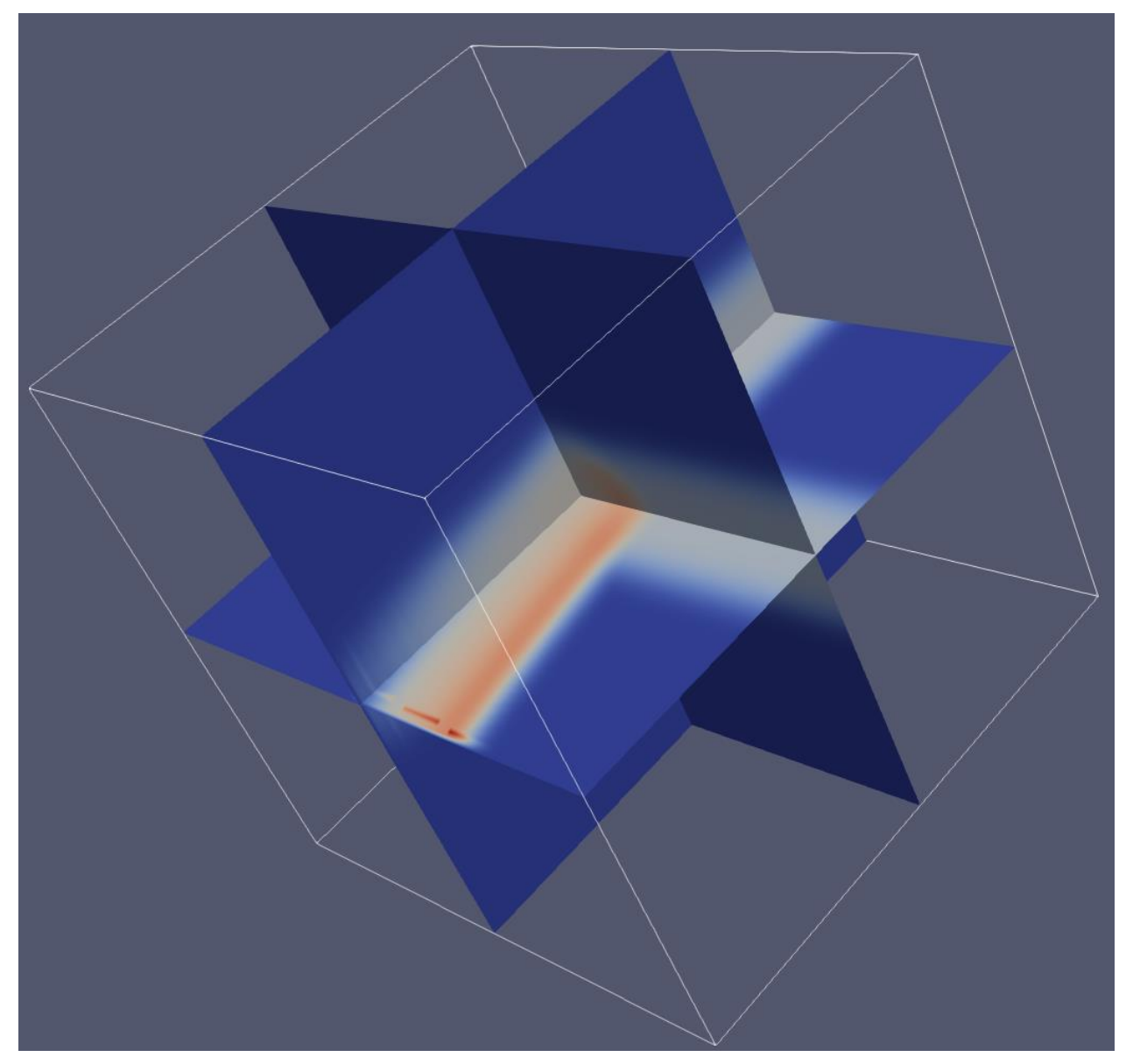
University of Rochester, Laboratory for Laser Energetics

Introduction and Background

- Cross Beam Energy Transfer (CBET) is a transfer of energy between lasers which intersect in a plasma.
- Computing CBET is a key component of a laser system simulation. We seek to speed up simulations to inform experiments, as conducting actual experiments is costly.



Before CBET



After CBET

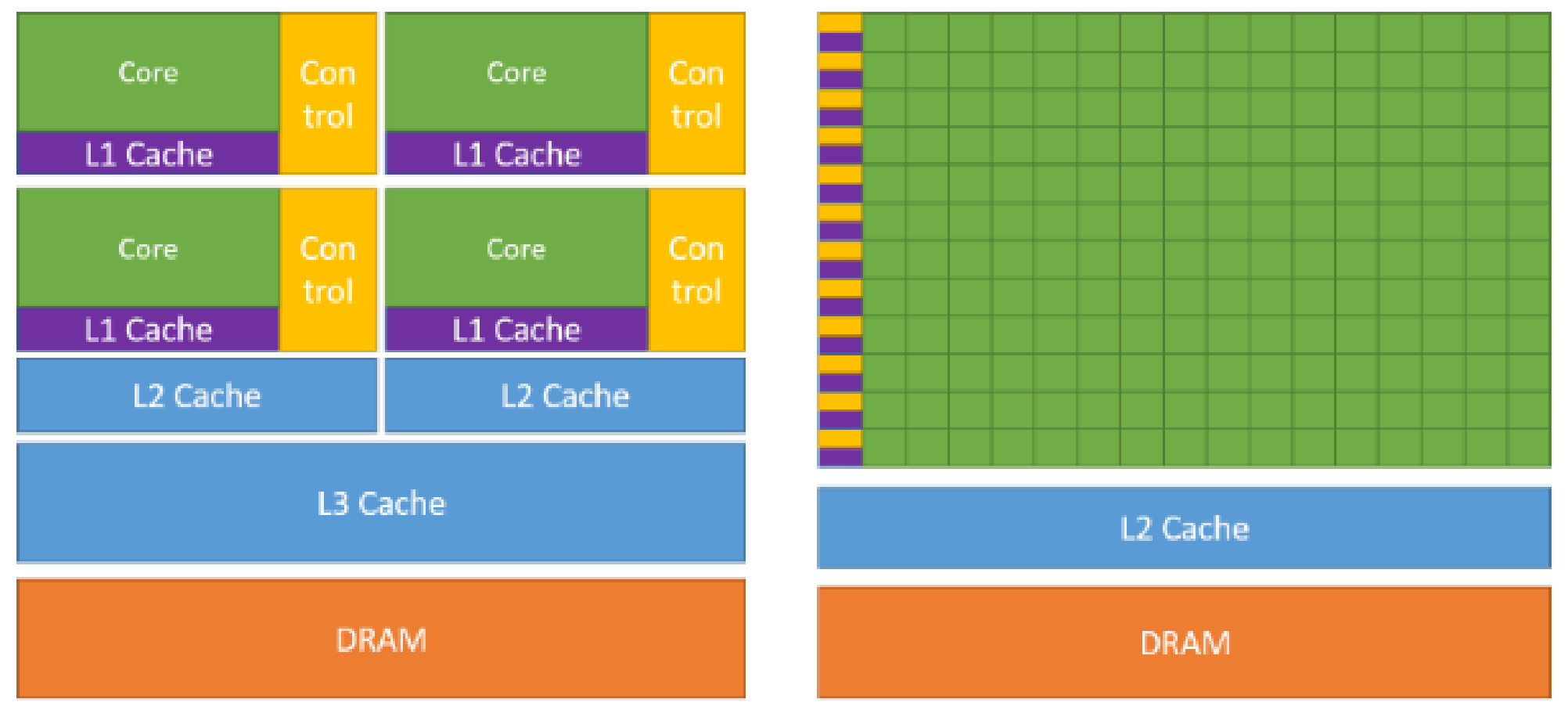
Our Simulation

Our code is based on the model described by Follett et al [1], which represents laser beams as a set of discrete rays, and the plasma as a discretized mesh. There are two main steps in the program's execution:

- **Raytracing**: As each ray's trajectory is traced across the mesh, information for CBET is calculated at each mesh zone crossing.
- **CBET:** For each pair of intersecting rays from different laser beams, an interaction multiplier is calculated, and intensities are updated. This step is repeated until the intensities converge.

GPU Parallelism

- The extent to which different tasks can be parallelized varies due to the constraints parallelism introduces
- Parallelization also presents an additional challenge to the programmer in adapting their code.
- GPUs have architectures specialized for massively parallel workloads.

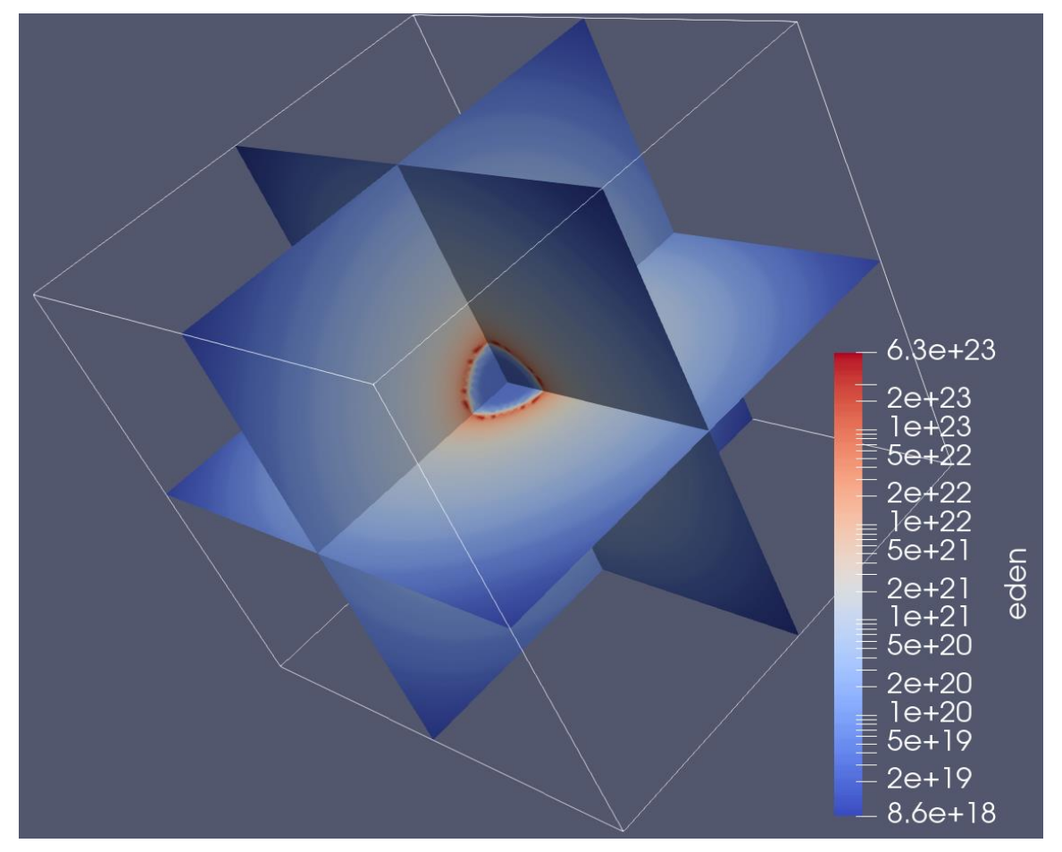


A comparison of CPU and GPU architectures (Nvidia)

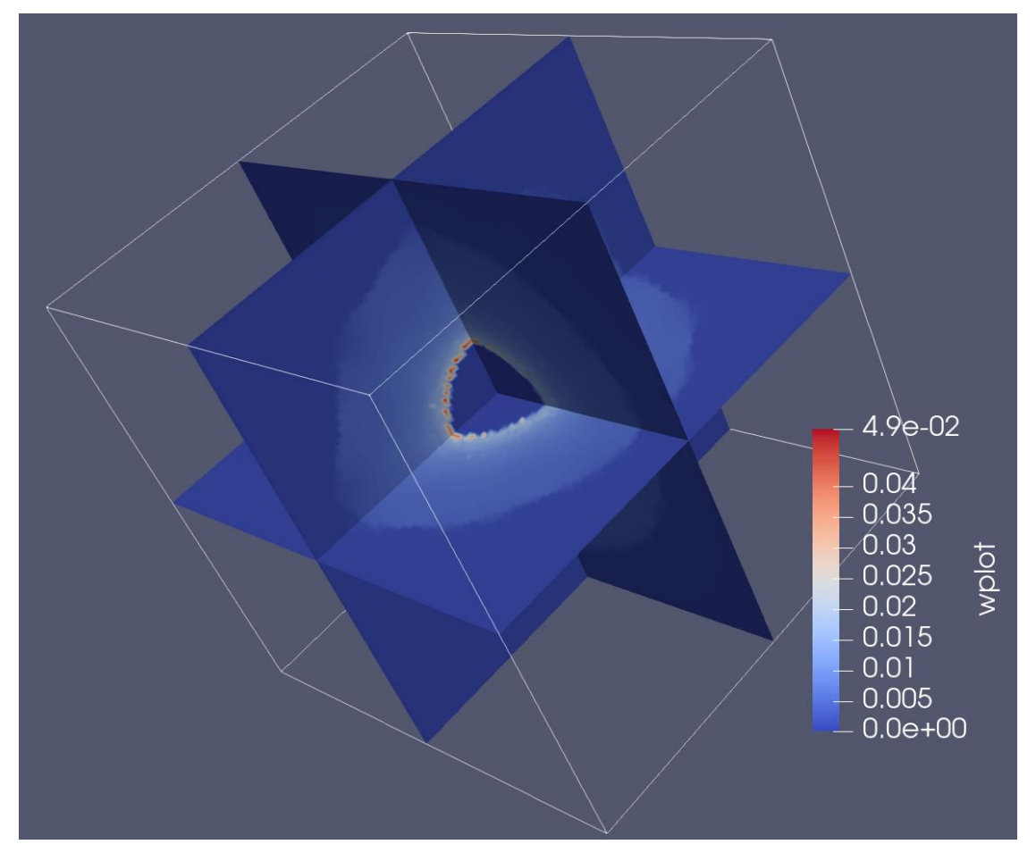
Parallelizing CBET

- Raytracing is considered embarrassingly parallel because none of the threads calculating ray trajectories need to communicate.
- The CBET step is more difficult to parallelize because all of the interaction multipliers must be calculated before intensities are updated, requiring multiple synchronizations at each iteration.
- The CBET step also requires all of the information to be available at once, which limits the resolution due to less memory on GPUs, and forces the program to re-compute values it could otherwise save due to the lack of memory..
- However, this model avoids many common issues with parallelizing code, such as data races and high overhead.

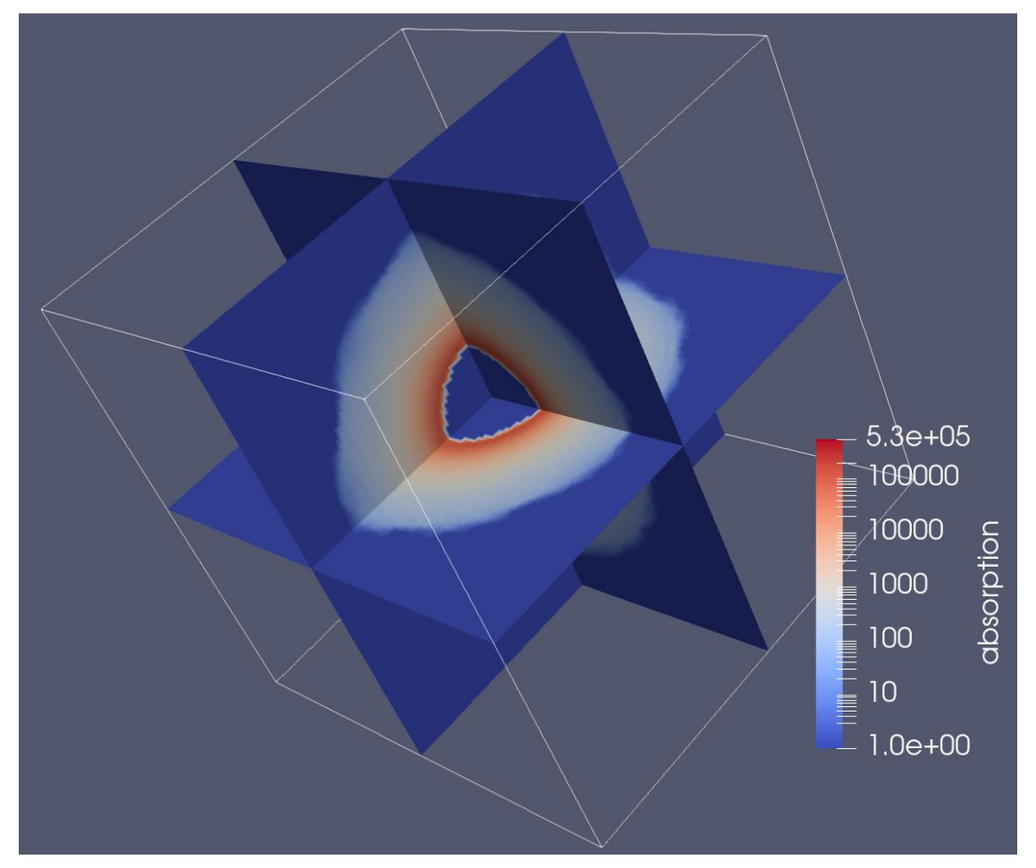
Results



Electron Density Profile

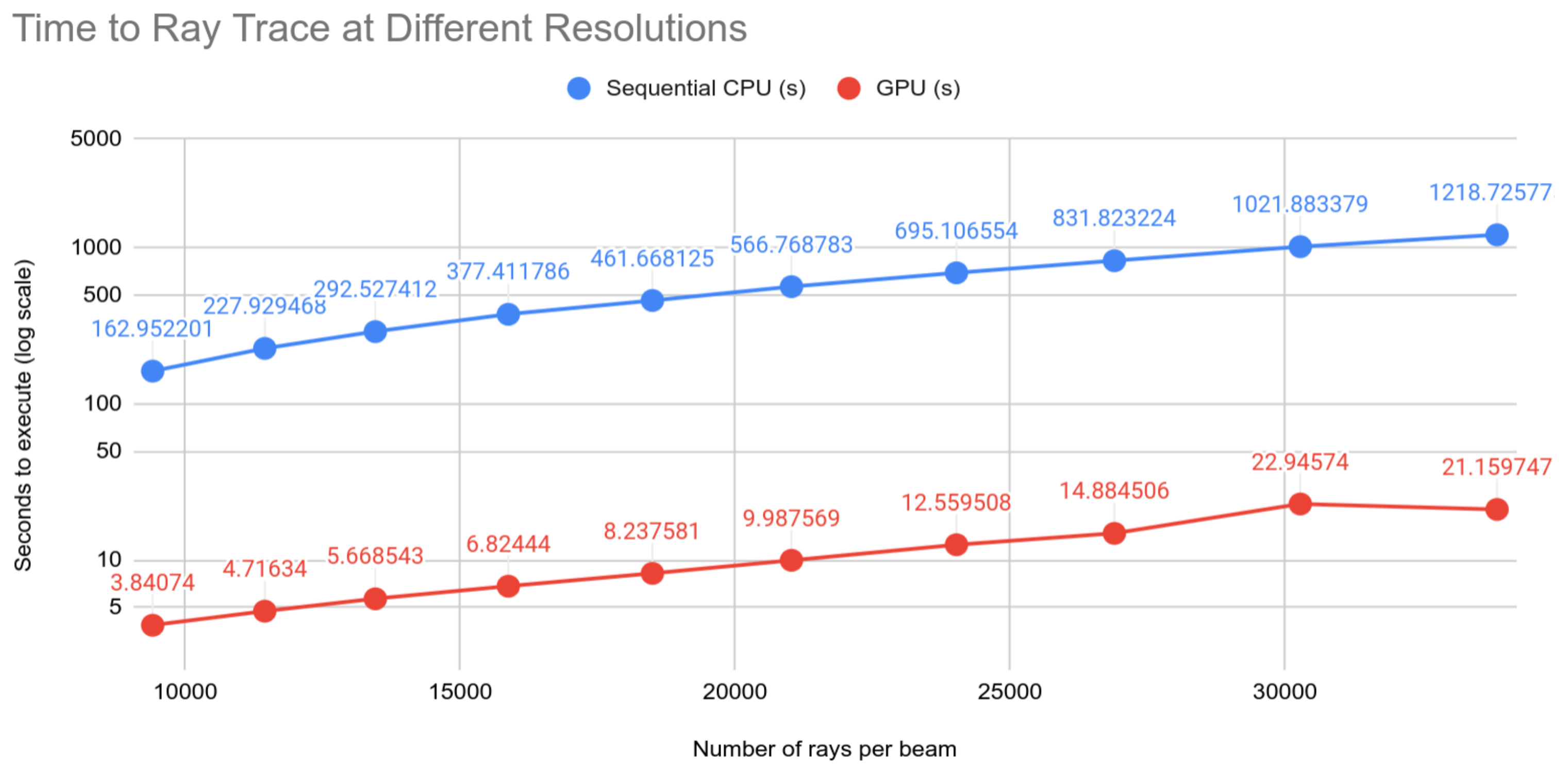


Laser intensity after CBET

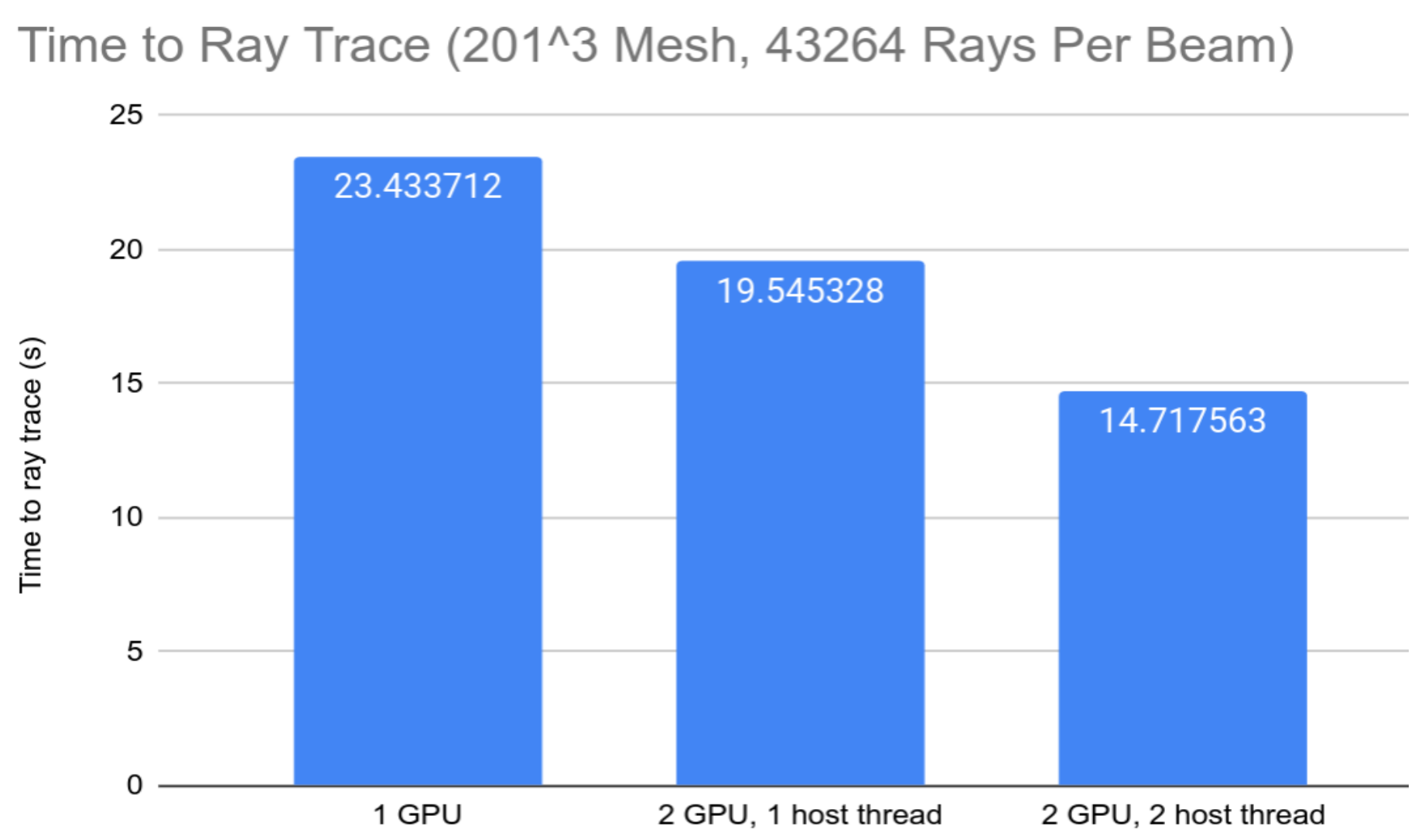


Absorption after CBET, TW/cm^3

Raytracing Performance

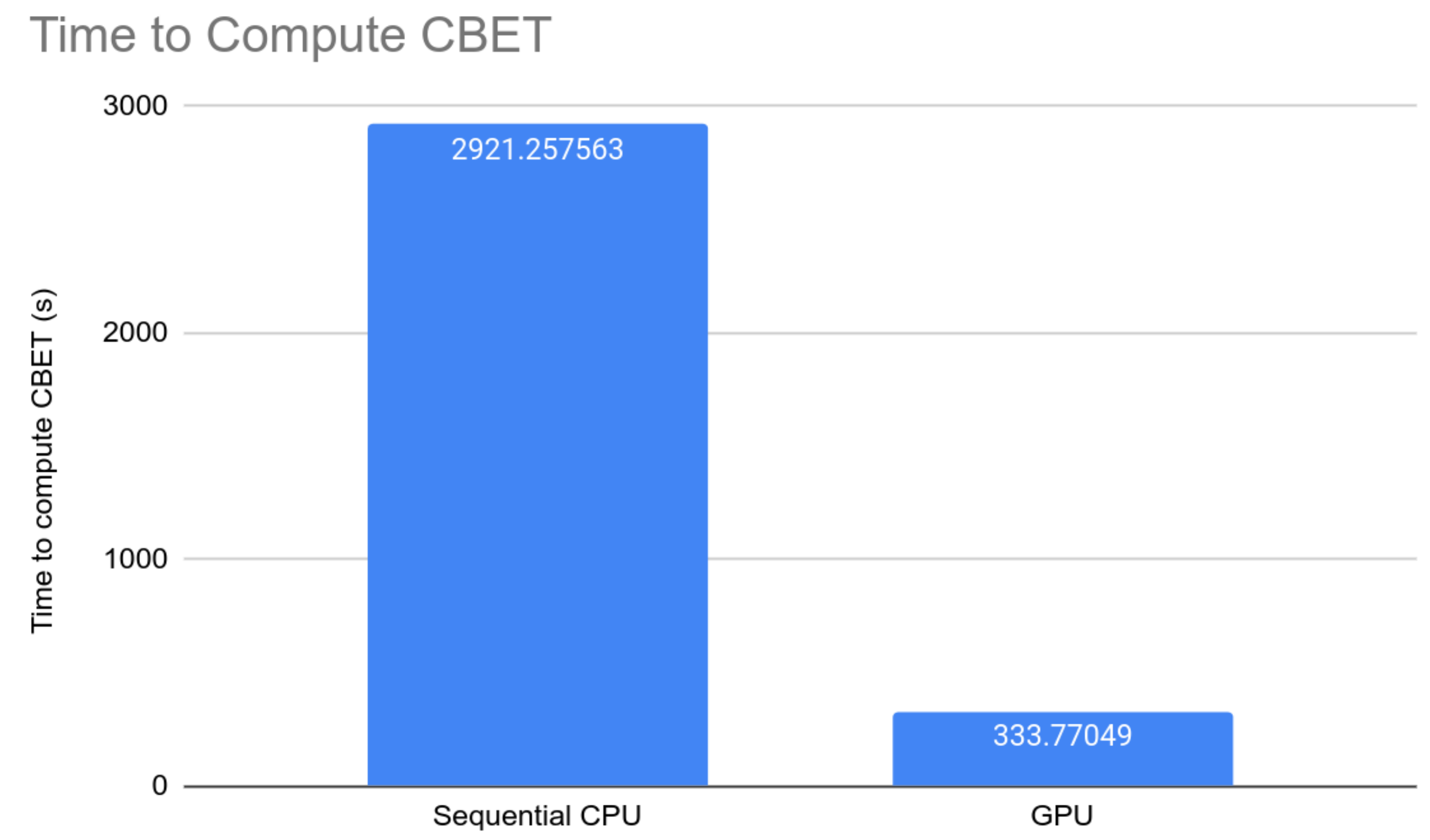


This graph shows the number of seconds it takes to trace the rays at different resolutions. At each point, the number of zones in the mesh is increased by 10 in each dimension, so more rays are needed to accurately represent each beam, and each ray will cross the mesh more times. The version of the code used here uses 2 GPUs.



The graph on the left compares execution time for multi-GPU programs. In this case, the problem is too big to fit on one GPU and requires multiple rounds of moving the data off of the GPU. Here, we see that there is a performance improvement when using two GPUs, although with overhead. Some of this overhead can be mitigated by spawning multiple threads on the CPU to manage each GPU separately.

CBET Performance



This graph compares the performance of a sequential implementation of the CBET computation on the CPU and a parallel one on the GPU.

Discussion and Future Work

Our results suggest that the programming cost of adapting code to take advantage of GPUs is worth it in instances where a problem can be identified as embarrassingly parallel. For code with more complicated memory access patterns, the returns for using GPUs, due to their unique architecture, are diminished.

Future work on the CBET simulation may focus on utilizing multiple GPUs in the CBET step, further combining GPU and CPU parallelization, and using multiple computers in a cluster with message passing

References and Acknowledgements

[1] R. K. Follett, J. G. Shaw, J. F. Myatt, V. N. Goncharov, D. H. Edgell, D. H. Froula, and J. P. Palastro, “Ray-based modeling of cross-beam energy transfer at caustics,” Phys. Rev. E, vol. 98, p. 043202, Oct 2018.

This material is based upon work supported by the Department of Energy [National Nuclear Security Administration] University of Rochester “National Inertial Confinement Fusion Program” under Award Number(s) DE-NA0004144.