Parallel IMC and Simple Load Balancing with the Data Server Model

Alex R. Long, along@lanl.gov

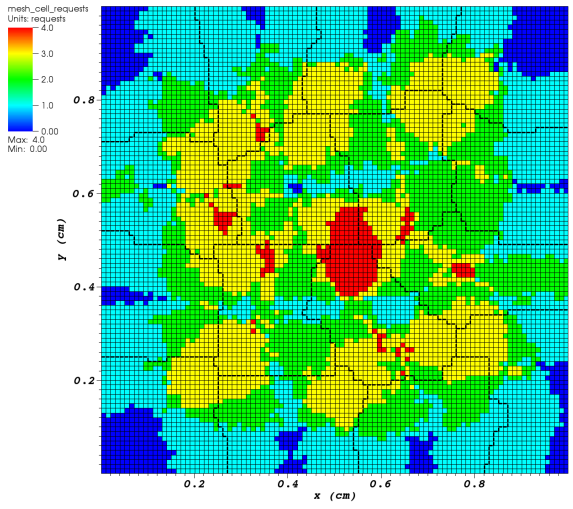
Implicit Monte Carlo [2] can dominate compute time in multiphysics simulations. This computational demand is made worse when the workload is not balanced between parallel processes. The standard method decomposes the spatial domain and passes particles between parallel processes. In the standard method, the load imbalance is proportional to the particle work imbalance, which can be large when an energy-based source strategy is used. A variant of the "data servers" model from neutronics is used to allow particle work to be decomposed independent of spatial and physical data. This method is called mesh passing. Mesh passing performs comparably to particle passing on load balanced problems and shows substantial improvement on problems with particle load imbalance. The amount of data held in a mesh cell does not significantly impact the scaling behavior, meaning that parallel IMC is likely bound by network latency and not network bandwidth.

Background and Motivation

In thermal radiative transfer (TRT) energy emitted from a material is is proportional to temperature to the fourth power. This means that small spatial gradients in temperature can lead to relatively large spatial gradients in emission energy. The Implicit Monte Carlo method solves the TRT equations by simulating photon particle histories with pseudorandom numbers. If particles are made to represent equal amounts of emission energy (energy-based source strategy), the large spatial gradients in emission energy translate directly into large spatial gradients in particle density. In parallel simulations where the mesh is decomposed, parallel processes may have a drastically different number of particles to simulate. As these simulated particles move off of the mesh owned by a parallel process some kind of parallel communication is necessary. In current IMC codes, the particle is buffered with other particles and then passed to the parallel process that owns the mesh needed by these particles. This method is effective if the particle count is balanced across ranks but requires advanced methods of selective replication if the particle load is not balanced.

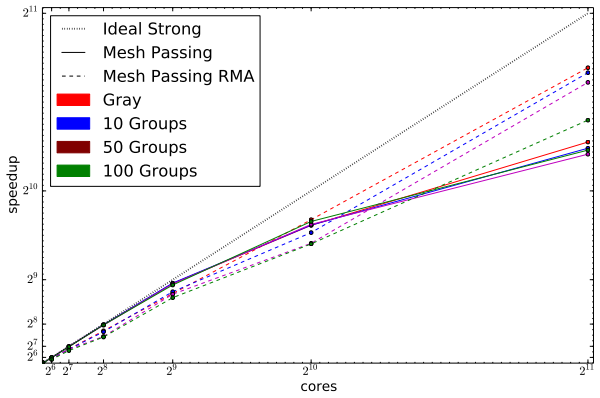
Description/Impact

Instead of passing particles that move out of a subdomain, the processor that owns the particle could request the mesh data needed by that particle. This is similar to the "data server" model in neutronics. This data server model has been used for spatial data in simple neutronics simulations [3] and to handle tallies in large nuclear reactor simulations [1]. In the current implementation of this model, which we call "mesh passing," mesh data is requested directly from the processor that owns the mesh data. This model allows a parallel process to simulate particle histories regardless of the spatial location of the particle. Particles can then be evenly distributed between parallel processes.



Heat map of data requested in parallel IMC with data servers.

Parallel IMC and Simple Load Balancing with the Data Server Model



Scaling of the data server model with various cell data sizes

Anticipated Impact

The mesh passing method is performs well when the medium is optically thick or when the timestep size is small. If either of those conditions are true, mesh passing should scale well regardless of particle load imbalance. The performance also relies on the effectively overlapping mesh communication with particle work. If these demands are met, the mesh passing method could drastically improve load imbalance problems in parallel IMC without the need for selective replication or work functions. Initial work shows that increasing the amount of data in a cell by a factor of one hundred (as in large multigroup simulations) only decreases scaling efficiency by about 10%. The promising results thus far do not include recent optimization in one-sided messaging in MPI 3.0 or in MPICH on Ares networks.

Path Forward

The mesh passing method has been implemented in a mini-app called *Branson* with two-sided MPI messaging and passive, one-sided MPI messaging. The method is currently being evaluated on a wide range of physical and algorithm parameters. We hope to use Branson to show the need and utility of fast one-sided RMA operations in future architecture procurements. Branson is open-source and available at https://github.com/lanl/branson.

Acknowledgements

Los Alamos Report LA-UR-17-29091. Funded by the Department of Energy at Los Alamos National Laboratory under contract DE-AC52-06NA25396.

References

[1] Nan Dun, Hajime Fujita, John R Tramm, Andrew A Chien, and Andrew R Siegel. Data decomposition in monte carlo neutron transport simulations using global view arrays. *The International Journal of High Performance Computing Applications*, 29(3):348–365, 2015.

[2] J. A. Fleck and J. D. Cummings. An Implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport. *Journal of Computational Physics*, 8:313–342, 1971.

[3] Paul K Romano, Benoit Forget, and Forrest Brown. Towards scalable parallelism in Monte Carlo particle transport codes using remote memory access. *Progress in Nuclear Science and Technology*, 2:670–675, 2011.