Adding GPU Support to the Markov Chain Monte Carlo Code CATMIP

In geophysics, we are confronted with many under-determined inverse problems. For example, all of our observations of earthquakes are made at the Earth's surface. So, when we try to infer how slip during an earthquake evolves in space and time, we find that there are many potential slip histories that are consistent with our limited observations and our understanding of earthquake physics. One way to approach these problems is with Bayesian analysis which allows us to infer the ensemble of all potential slip models that satisfy the observations and our prior knowledge of earthquake physics. In Bayesian analysis, our prior knowledge is known as the prior probability density function or prior PDF, the fit to the data is known as the data likelihood, and the target PDF that satisfies both the prior PDF and data likelihood is known as the posterior PDF. However, simulating the posterior PDF typically requires using Markov Chain Monte Carlo (MCMC) to draw tens of billions of random realizations of earthquake slip models, which may not be computationally feasible. To make this and similar geophysical inversions computationally tractable, we developed the **C**ascading **A**daptive **T**ransitional **M**etropolis **I**n **P**arallel (CATMIP) algorithm. CATMIP is an efficient parallel Markov Chain Monte Carlo (MCMC) sampler that is used for model fitting and uncertainty quantification in geophysics. Example use cases are earthquake rupture modeling, determining mineral composition on Mars, reconstructing the history of ocean salinity, and historical earthquake relocation. CATMIP employs many parallel instances of the Metropolis algorithm for sampling in a transitioning framework. Transitioning is a process in which a set of random samples at equilibrium with a known probability density function (PDF) are used as seeds for the Markov chains to sample successive target PDFs that incrementally move the distribution from the starting seeds to the final desired PDF that describes the relative plausibility of potential values for the model parameters. The algorithm is implemented as a Master-Worker model employing MPI for communication. The worker processes are loosely coupled with global parameters periodically optimized by the master process. This provides a very high amount of parallelism with little communication between updates. During the presentation we will discuss the history of the algorithm and elaborate the earthquake rupture modeling use case for the CATMIP package.

Our first step toward GPU optimization was to optimize the code for the CPU. CPU profiling revealed that most of the compute time is spent in calls to level 2 BLAS routines and calls to GSL random number generators. We revised the algorithm to employ level 3 BLAS routines instead. In our presentation we will describe how this was accomplished. Adding GPU support to CATMIP consisted mostly of replacing the calls to GSL with calls to GPU vendor-provided library routines. A small number of loops were directly implemented in CUDA. In the presentation will provide implementation details. Finally, we will discuss methods for profiling and opportunities for further optimizing GPU execution. By creating a code with the flexibility to run on either a CPU or GPU architecture, CATMIP can be used on systems ranging from large CPU-based HPC environments to single servers with GPU acceleration and everything in between.