Adaptive Execution of Particle Advection Workloads

Kristi Belcher

I. Abstract

Particle Advection is a fundamental flow visualiation calculation with widely varying workloads. Determining the optimal architecture to run these workloads on is a critical consideration. In this study, we investigate the important tradeoffs that must be factored in when deciding how to best schedule the Particle Advection workloads on appropriate resources. Given this insight, our main contribution is a new algorithm which adapts execution using the GPU to run relevant parts of the problem and then switch the targeted device according to how the workload evolves. This algorithm is motivated by the observation that CPUs are sometimes able to better perform part of the overall computation since (1) this practice avoid latency times to the GPU and (2) CPUs operate at a faster rate when the workload can't take advantage of the threads on a GPU. We evaluate our algorithm by running workloads that vary over data set, number of particles, number of steps taken, and number of GPU nodes. We then compare our algorithm to traditional GPU-only and CPU-only approaches. Our findings show X, Y, and Z. The results of this study will help inform the Scientific Visualiaiton community about those architectures that give optimal performance at certain stages of the simulation run.

II. Intro

Particles. They Advect.

III. Related Work

We read lots of related work that built the basis for our study.

IV. Experiments

We had many experiments.

Perhaps eventually put a diagram here.

V. Conclusion

We studied the affects of adaptive device selection during a vtk-m Particle Advection run. We have shown that by analyzing the current workload and selecting an appropriate device, we get better performance over the traditional single-device alternatives. Our findings suggest that adaptively selecting an appropriate device can also lead to better energy efficiency, in addition to enhanced performance. We hope to get this paper published and solve world hunger while we're at it.