Percy Link
CS267 Assignment 0
Feb 7 2013

Bio and Research Interests

I am a 4th year Ph.D. student in Earth and Planetary Science, studying atmospheric science. I am particularly interested the water cycle, and my dissertation focuses on land evaporation and precipitation in California.

In my research, I use numerical models of the atmosphere and land surface, and I analyze large datasets (model output, satellite observations, and ground-based observations.) In general, the numerical models are community codes that are already programmed to run in parallel, but I write my own code to analyze data. I am taking this class largely because I'd like to be able to write code to analyze large volumes of data quickly. I would also like to learn about other ways to use parallel programming to enhance my research.

Application: Identifying Extreme Events in Climate Model Output

Prabhat from LBNL developed an "embarrassingly parallel" algorithm to search 13 TB of high-resolution climate model output for extreme weather events (hurricanes, extratropical cyclones, atmospheric rivers, heat waves and droughts, and cold snaps.) The scientific purpose was to evaluate how well the model represents these extreme events in space and time, by comparing modeled extreme event statistics with observed statistics.

The problem involves analyzing a very large dataset - multiple fields at high spatial (25 km over the whole globe) and temporal (3 to 6 hour for ~30 yr) resolution. Prabhat's application takes advantage of parallelism by breaking the problem into two steps. First, all points in space and time are checked against criteria for hurricanes (or other extreme weather phenomena). This step can be made "embarrassingly parallel" because each point in space and time can be analyzed independently. The second step is to stitch the qualifying points into trajectories; this step is run sequentially, but it is computationally cheap.

The application was written in C++ and used MPI. It was run on Hopper at NERSC, which ranked 19th on the Top 500 list in November 2012. For hurricanes, the detection step was run on 80,000 Hopper cores and completed in ~1 hour. If run in serial, the job would have taken ~9 years. The serial stitching step was completed in ~10 seconds.

The resulting hurricane detection statistics compared favorably to observations, although much of the credit goes to the climate model rather than to the detection algorithm. This analysis identified regions where model simulates hurricanes well (North Atlantic) and regions where it simulates hurricanes poorly (North Pacific). A remaining question is whether the algorithm correctly identified model hurricanes, and what its rate of false positives and missed detections is.

Because this application is embarrassingly parallel, it can scale to many processors. A practical bottleneck is the long waiting period in the queue for time on 80,000 cores at once. In practice, it might be faster to run on an order of magnitude fewer cores to reduce the queue wait time, despite the increase in computing time.

References:

Prabhat, "13 TB, 80,000 cores and TECA: The search for extreme events in climate datasets." Oral presentation at the American Geophysical Union Fall Meeting 2012.

Prabhat, O. Rubel, S. Byna, K. Wu, F. Li, M. Wehner, and W. Bethel. TECA: A parallel toolkit for extreme climate analysis. *Procedia Computer Science* 00 (2012) 1–10