CS475 Research Topic:
Towards Implementation of the Colorado State University Global Cloud Resolving Model on the NSF Blue Waters System

Reviewed by: Laurette Hamlin, November 20, 2016

The paper I read was entitled, "Towards Implementation of the Colorado State University Global Cloud Resolving Model on the NSF Blue Waters System, by Dr. David Randall and Dr. Ross Heikes in the Dept. of Atmospheric Sciences at CSU. I was interested specifically in knowing how parallel computing was being applied in active research on our campus. Since I also have an interest in Atmospheric Dynamics, I searched for meteorological models being implemented that were highly suited to the HPC techniques we have been learning in class, and found this paper. It was perfect for me to read because it involved the hands-on implementation of several techniques we learned in class. I was quite excited to read how class topics had been successfully applied in this real-life scenario.

The CSU Global Cloud Resolving Model (GCRM) implements an important new technique in the study of convection and clouds: high resolution processing achieved through 2-D horizontal grid spacing. Since this model also involves both a vertical dimension (which must scale low enough to simulate individual clouds and high enough to simulate entire cloud systems) and a time dimension (to simulate entire cloud life cycles) on top of the curved 2-D horizontal grid, a highly efficient parallel design is critical for the system to be feasible. In fact, previous to approximately the year 2000, the general cloud resolving model was just too computationally inefficient to run in a timely manner.

Typically, the GCRM ran on two configurations at the National Energy Research Scientific Computing Center (NERSC): "Hopper", a Cray XE6 which has since been retired, and "Edison", a Cray XC30. In 2013, the model was adapted to run on one of the foremost supercomputers in the world, the Blue Waters system. It is a Cray XE/XK hybrid machine composed of AMD Interlagos processors and NVIDIA Kepler accelerators.

I found that the model's "icosahedral grid domain decomposition" was a cool spherical tiling technique that involved first breaking the earth's surface into 10 quadrilateral regions, each of which could then be recursively broken down further into blocks of four sub-domains to completely tile the earth's surface with as few as 40 tiles or as many as computationally feasible. The paper gave a visual example that showed an overall total of 642 cells covering a sphere, each to be assigned to MPI tasks, and then showed how the depth of recursion in generating the grid had to be carefully considered as it corresponded to the resolution of the model processing. Depths of

recursion in determining resolution ranging from 9-12 were measured, which corresponded to block sizes ranging from 64 to 4 cells. Since each level of depth sub-divided each block 4 times, the total number of cells grew roughly by a factor of four with each level of depth, and grid point spacing decreased roughly by half. The largest measured by the paper, at recursion depth 12, had a total of 167,772,162 cells in the horizontal grid and could calculate up to 256 layers in the vertical direction!

I also learned an interesting new technique for accomplishing ghosting of tiles: Morton-style numbering of the cells and the blocks. I discovered that the Morton code is a function that maps multi-dimensional data to one dimension while preserving locality of the data points. For the GCRM model, Morton numbering allowed "the physically contiguous tiles to also be logically close to each other." This was critical because the model's numerical finite-difference operators required data from neighboring tiles using MPI sends and receives. Although the paper did not explicitly say so, I pondered whether this also led to efficiency by coalescing the horizontal grid memory on a coarse-grained level.

As for measured speedup on different systems, the paper gave a good example of plotting number of tasks to time on a double logarithmic scale for four scenarios and four recursion depths, almost exactly like our lab7, and did an excellent job of interpreting the results. In this case, even though the Blue Waters system was twice as fast, the original Cray systems scaled better. This led to an excellent analysis of communication overhead for ghost cell updates over various numbers of cores. The authors were able to show that even though the fastest times stayed fairly consistent, the longest times dramatically increased as the number of cells increased. Furthermore, even though there were relatively few of these outliers, they caused all cores to wait for them to complete, causing up to a 3x slowdown, so increasing the number of cores could have a heavy impact on efficiency.

The paper also discussed migration to GPU processing. The largest consideration was the "relaxation operator", because it was a global variable which would require MPI communication overhead between blocks as well as nearest neighbor halo communication within each block. It was determined that the transfer from CPU to GPU was a relatively slow bottleneck, offset by the high computation performance improvements from the GPUs, but the efficiency depended on the size of the tile. The paper concluded with a note that data transposes should be explored next.

Overall, this was a very fun and exciting paper to read because it was at a perfect level for me to understand and explore. In fact, I contacted Dr. Randall about this paper and have been invited to see the model for myself. I cannot wait to meet my fellow HPC coders and see in person how their work implements what I have been taught in class.

Bibliography

Randall, David and Heikes, Ross. "Towards Implementation of the Colorado State University Global Cloud Resolving Model on the NSF Blue Waters System." *Blue Waters Sustained Petascale Computing*. Accessed November 15, 2016. https://bluewaters.ncsa.illinois.edu/documents/10157/5ae7eae1-146a-4c75-9851-653eb558b375

Duda, Jeffrey D. "A Review on the Uses of Cloud-(System-) Resolving Models." *Iowa State University, Department of Geological & Atmospheric Sciences*. Accessed November 15, 2016. http://www.meteor.iastate.edu/~jdduda/portfolio/542.pdf

Various authors. "Z-order curve." *Wikipedia*. Accessed November 18, 2016. https://en.wikipedia.org/wiki/Z-order_curve

Author unknown. "Elliptic Partial Differential Equation." *Wolfram MathWorld*. Accessed November 20, 2016. http://mathworld.wolfram.com/EllipticPartialDifferentialEquation.html