Eurographics Symposium on Parallel Graphics and Visualization (2015)
C. Dachsbacher, P. Navrátil (Editors)

Visualization of High-Resolution Weather Model Data

S. Liu[1], G. Foss[1], G. Abram[1], and A. Bowen[1]

[1]University of Texas & Austin TX, USA

Abstract

Significant bottlenecks from I/O required novel techniques be used for a high-resolution severe weather simulation and visualization in the region centered around O'Hare International Airport. The Raytheon Company collaborated with the Texas Advanced Computing Center (TACC) and National Center for Atmospheric Research (NCAR) to utilize the Weather Research and Forecasting Model (WRF). WRF is a state-of-the-art parallel mesoscale numerical weather prediction system. In order to minimize the data read from disk, we develop a custom importer in Paraview to read the WRF output data directly from HDF5, de-stagger variables as required, convert the grid coordinates to first create accurate real-world (longitude, latitude, elevation) coordinates and then project them into a WGS84 coordinates. This projection enabled us to correctly co-locate the WRF data with an aviation map base layer that was texture mapped onto a refined grid to approximate the non-linearity of a (longitude, latitude) coordinate system.

Categories and Subject Descriptors (according to ACM CCS): Modeling and Simulation [Computing Methodologies]: Simulation types and techniques—Scientific Visualization

---

1. Introduction/Background

WRF [WRF15] is a parallel mesoscale numerical weather prediction system. It is suitable for traditional mesoscale meteorological simulations as well as high-resolution simulations. There are several previous studies and experiments implementing a variety of WRF simulations that are extreme in resolution or scale on different platforms. The most significant ones include the Hurricane Sandy landfall simulations performed on Blue Water at National Center for Supercomputing Applications [JSS*13], the nested high-resolution heavy rainfall simulations on IBM Blue Gene/P [MSG*12], the idealized rotating fluid simulations on a dry atmosphere on BlueGene/L at IBM Watson and Lawrence Livermore National Laboratory [MHL*07]. Significant bottlenecks from I/O required novel techniques be used for the simulation and visualization of this data-set and are detailed in this paper.

2. Target domain and Expected Resolution

The region of interest is a cylinder area centered around the O'Hare International Airport shown in Figure 1. The diameter of the area is about 224 kilometers; the height of the area is over 21 kilometers. The resolution of our simulation is about 167 meters in the horizontal direction and about 90 meters in the vertical direction. The data collection interval is set to be 3 seconds, which allows us to produce around 2400 frames in total for a simulation of two 1-hour times (each approximately 1200 frames) of pre-determined interesting weather phenomenon.

3. WRF Model

The WRF model supports four main modes: serial mode, shared-memory parallel mode, distributed-memory parallel mode, and hybrid mode. In this project, we build an optimized WRF model (version 3.5.1) with Intel-13 compiler and MVAPICH2 library, and the distributed-memory parallelism is utilized to complete most of the simulation runs. To cover the complete target area of our interest comprehensively with a satisfactory resolution, we employ a mesh with 1345x1345 grid cells in the horizontal direction and 234 levels in the vertical direction. This also defines the dimension of the output data for future processing and visualization: 1345x1345x234.

To achieve expected high resolution, we employ a one-way nested WRF model run with a three to one nesting ratio in the horizontal direction and a one to one nesting ratio in

DOI: 10.2312/pgv.20151156

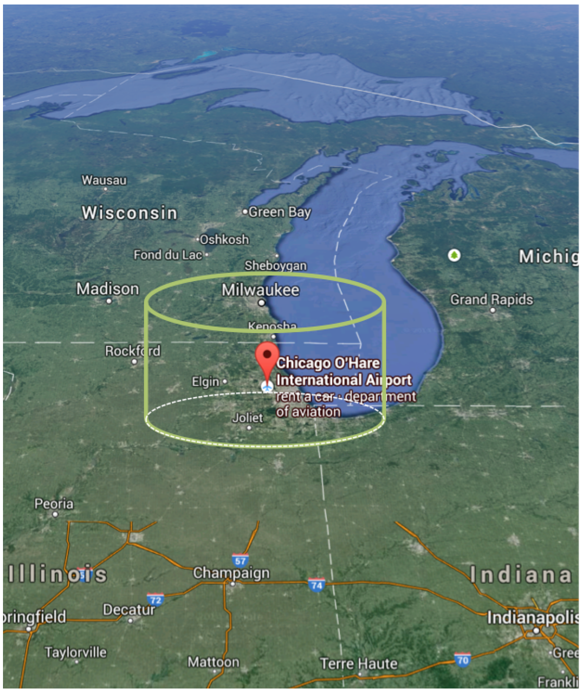

Figure 1: *The target area around O'Hare International Airport, Chicago, Illinois.*

the vertical direction. The GFS input data [Nat15] is first obtained from the Global Forecast Systems produced by the National Centers for Environmental Predction. An independent coarse-grid (parent) run is then executed prior to a fine-grid (child) run to create more accurate initial and lateral boundary conditions for the fine-grid run. Afterwards, the fine-grid run is executed and yields a huge amount of output data for analysis and visualization. The boundaries of both the parent and child domains are displayed in Figure 2.

The WRF climate model nicely supports domain decomposition in the horizontal direction, making it straightforward to introduce more grid cells and subdomains and perform the simulation with more processes and MPI tasks when a larger area needs to be studied. There is no vertical partition supported in the WRF model, making high resolution in the vertical direction significantly increase the size and complexity of each single subdomain, which inevitably aggregates difficulties to the model simulation.

In this project, we focus on the high-resolution simulations covering the following time period of interest: from *2010-6-23-21:00:00* to *2010-06-23-22:00:00* and from *2010-06-23-23:00:00* to *2010-06-24-00:00:00*. The high resolution in both the horizontal and vertical directions makes the choice of the timestep very limited. In the demo simulation shown in this paper, the timestep is chosen to be 0.5, 0.75, or 1 second depending on the period of time we study to ensure that the simulation maintains the numerical stability.

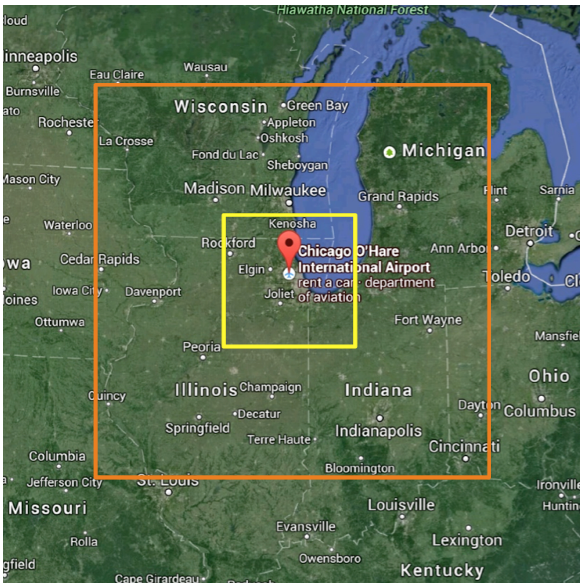

Figure 2: *The boundaries of the parent and child domains.*

4. Computing and Visualization Resources

Our main computing and visualization resource to complete high-resolution simulations, data processing, and visualization work is TACC's Stampede supercomputer [Tex15]. Stampede is a Dell Linux Cluster with over 6400 compute nodes, each equipped with two Xeon E5-2680 processors and one or two Intel Xeon Phi coprocessors. There are 16 extra large-memory nodes with 1 TB of memory and 32 cores/node for data-intense applications on the Stampede supercomputer. The peak performance of the machine is about 10 PFLOPS. Additionally, a number of preliminary tests are implemented on NCAR's Yellowstone supercomputer [Com15].

5. I/O Workflow

WRF supports several different I/O mechanisms, including "spokesman" sequential I/O, MPI collective parallel I/O with shared data files, and parallel I/O with independent data files. The sequential I/O method uses only one process to read and write data files, restart files, and output files throughout simulation. In our target simulation, over 10 GB of disk space is demanded per timestep to store output files for visualization. Consequently, about 30 TB of data are generated for the simulation of two 1-hour times. Therefore, the sequential I/O method is not feasible, as the time spent in I/O operations increases significantly in such a large-scale simulation. The MPI collective parallel I/O method designates a few processes to carry out I/O operations on the data files as parallel I/O aggregators. To achieve high efficient performance, the method requires extra support of the parallel versions of HDF5 [The15] or NetCDF [Uni15] and advanced tuning or optimization. Although some user-friendly tuning tools have been developed by Mclay et al. [MJL*14]), Zimmer et al. [ZKL13] and others, such as the T3PIO library, optimizing the parallel performance of this method is still very challenging in general. The parallel I/O method with independent data files utilizes all processes to perform I/O to independent data files. Although satisfactory performance can be obtained when the number of processes is small, excessive data file access creates contention to the Meta Data Server (MDS) and Object Storage Targets on the parallel Luster file system.

In our simulation, we modify the parallel I/O with independent data files and break the bottleneck of metadata operation. We develop a collection program along with the normal WRF simulation, which writes the data files to the local disk on compute nodes, packs all data files into one single tarball per compute node, sends the tarball to the Lustre file system, and cleans the packed data files on the local disk. As a result, the total number of MDS requests is significantly reduced, and the performance is very efficient and scalable. This modified parallel I/O method takes only 10 minutes to complete the one-hour model simulation, in contrast to 8 days for the “spokesman” sequential I/O method in our typical simulation runs with 1024 processes on 128 Stampede’s compute nodes.

Since we are only interested in a number of target two-dimensional and three-dimensional variables, like the radar reflectivity, cloud water content, cloud ice content, snow content, graupel content, we also modify WRF registry file (EM_COMMON) and rebuild a specific version of WRF to limit the variables in the output files. This reduces the output file size by more than 30%. Meanwhile, a restart mechanism is used in our work in order to make all simulations complete within a reasonable wallclock limit and therefore reduces the risk of job failure and data loss.

6. Data processing and visualization

In the midst of the simulations, we validate all output data by examining the number of data files and their sizes. After the simulation finishes, we unpack these tarball files. Facilitating data processing and visualization, we also regroup the output files from process-based to timestep-based, i.e. all data files for a single timestep are grouped together.

In order to minimize the data read from disk, we develop a custom reader for ParaView [Kit15], our chosen visualization software. This reads the WRF output data directly from HDF5, de-staggers variables as required, and converts the grid coordinates to first create accurate real-world (longitude, latitude, elevation) coordinates then project them into a WGS84 coordinates. This projection enables us to correctly co-locate the WRF data with an aviation map base layer texture mapped onto a refined grid to approximate the non-linearity of a (longitude, latitude) coordinate system. We include this aviation map to guide the viewer’s experience, supporting the project’s theme of weather as pertaining to piloting aircraft.

We then use the tools in ParaView’s graphical user interface to model the data with isosurfaces, clipping, slicing, and streamlines, creating scenes for images and animations. The actual rendering process is via python scripts that load the scenes saved from GUI (state files). After final rendering, resulting segments and variations are chosen for the video and edited together with a bash script running commands from the ImageMagick toolkit to create the transitions. Gimp is used for various other tasks such as titles. Because the rendered animation frame files don’t include alpha information, a simple python script serves as compositor for layering images showing reflectivity over the aviation map texture. The lowest reflectivity values are mapped to black and this script replaces the black with corresponding pixels from a second rendering with everything removed except the aviation map. We present three figures in this paper. Figure 3 and Figure 4 show scenes of rain water mixing ratio (qrain) at early and late timesteps in the simulation. Rain water mixing ratio (qrain) is modeled as a volume, and its top is clipped at approximately 6500 feet altitude. The clouds outer surface represents a qrain value of 0.0001 kg kg^{-1}, while the sliced off top reveals a range (0.0001 - 0.001 kg kg^{-1}) from grey and white for the lowest values, shades of green for middle values, and the warmest colors to red for the highest. In Figure 5, wind velocity at 2000 feet altitude is modeled as streamlines, colored by magnitude (low values (blue) to high values (red)). Reflectivity is included as a slice close to ground level and colored with a simplified version of the weather maps conventional color key.

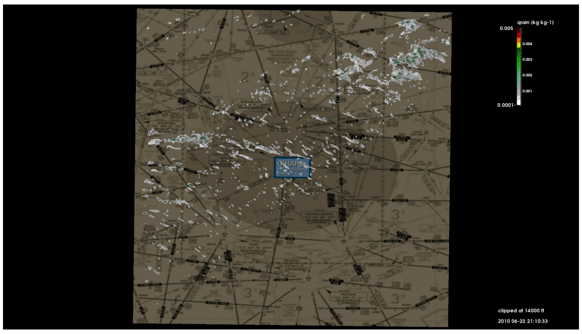

Figure 3: *A scene of rain water mixing ratio around the target area, clipped at 14,000 feet.*

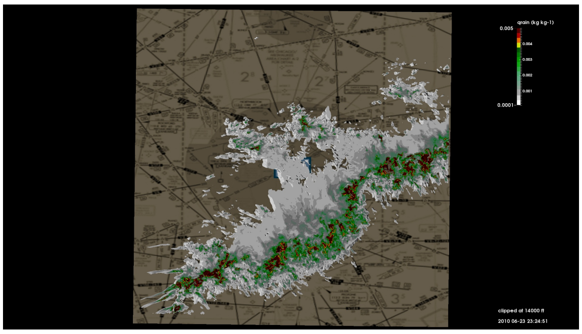

Figure 4: *A scene of rain water mixing ratio around the target area, clipped at 14,000 feet.*

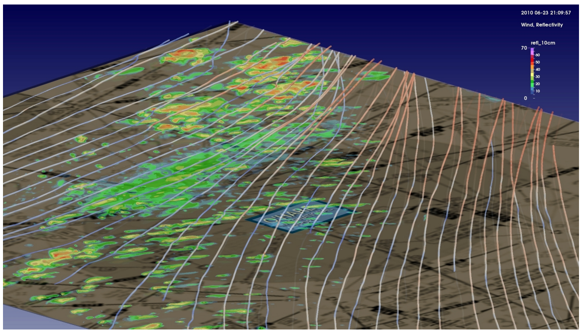

Figure 5: *A scene of wind and reflectivity around the target area*

7. Conclusion and Future Work

We have modeled a specific region and time frame and produced meteorological data of extremely high resolution, both in time and space beyond most similar weather simulations. With this level of resolution, meteorologists are able to observe subtle weather changes at local areas. Moreover, all these techniques are applicable to a great deal of high-resolution simulations, memory-intensive and/or I/O-intensive applications, and other supercomputer platforms.

References

[Com15] COMPUTATIONAL AND INFORMATION SYSTEMS LABORATORY, CITED 2014: Yellowstone: IBM iDataPlex system, 2015. `http://n2t.net/ark:/85065/d7wd3xhc`. 2

[JSS*13] JOHNSEN P., STRAKA M., SHAPIRO M., NORTON A., GALARNEAU T.: Petascale WRF Simulation of Hurricane Sandy Deployment of NCSA's Cray XE6 Blue Waters. *The International Conference on High Performance Computing, Networking, Storage and Analysis (SC13)* (2013). 1

[Kit15] KITWARE: Paraview home page, 2015. `http://www.paraview.org`. 3

[MHL*07] MICHALAKES J., HACKER J., LOFT R., MCCRACKEN M., SNAVELY A., WRIGHT N., SPELCE T., GORDA B., WALKUP R.: WRF Nature Run. *Proceedings SC'07/Gordon Bell Prize Finalist* (2007). 1

[MJL*14] MCLAY R., JAMES D., LIU S., CAZES J., BARTH W.: A user-friendly approach for tuning parallel file operations. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Piscataway, NJ, USA, 2014), SC '14, IEEE Press, pp. 229–236. URL: `http://dx.doi.org/10.1109/SC.2014.24`. 3

[MSG*12] MALAKAR P., SAXENA V., GEORGE T., MITTAL R., KUMAR S., NAIM A. G., HUSAIN S. A. B. H.: Performance evaluation and optimization of nested high resolution weather simulations. *Euro-Par'12 Proceedings of the 18th international conference on Parallel Processing* (2012), 805–817. 1

[Nat15] NATIONAL CLIMATIC DATA CENTER: Global Forecast System, 2015. `http:///www.ncdc.noaa.gov/data-access/modeldata/model-datasets/global-forcast-system-gfs`. 2

[Tex15] TEXAS ADVANCED COMPUTING CENTER: Stampede User Guide, 2015. `http://portal.tacc.utexas.edu/userguides/stampede`. 2

[The15] THE HDF GROUP: HDF5 Home Page, 2015. `http://www.hdfgroup.org/HDF5`. 3

[Uni15] UNIDATA PROGRAM CENTER: NetCDF downloads, 2015. `http://www.unidata.ucar.edu/downloads/netcdf/index.jsp`. 3

[WRF15] WRF DEVELOPMENT TEAMS: The Weather Research and Forecasting Model, 2015. `http://www.wrf-model.org/index.php`. 1

[ZKL13] ZIMMER M., KUNKEL J. M., LUDWIG T.: Towards Self-optimization in HPC I/O. *Lecture Notes in Computer Science* (2013), 422–434. 3