4.6 EXTREMELY HIGH-RESOLUTION WEATHER MODEL SIMULATION, DATA PROCESSING, AND VISUALIZATION

Si Liu[1*], John Cazes[1], Greg Foss[1], Greg Abram[1], Donald Cook[2], and Craig Stair[2]
[1]Texas Advanced Computing Center, Austin, TX
[2]Raytheon Company, Dallas, TX

1. INTRODUCTION & BACKGROUND

To achieve extremely high-resolution severe weather simulation and visualization in the area of O'hare International Airport (ORD) as well as other domains of interest, Raytheon Company collaborates with Texas Advanced Computing Center (TACC) and National Center for Atmospheric Research (NCAR) to operate the simulation with Weather Research and Forecasting Model (WRF) (WRF Development Teams, 2015). WRF is a state-of-the-art parallel mesoscale numerical weather prediction system. It serves a large worldwide community of over 25, 000 users in over 130 countries for both atmospheric research and operational forecasting needs. It is traditionally suitable for mesoscale meteorological simulations, but is applied to high-resolution simulations as well in recent days.

There are several previous studies and experiments implementing a variety of extreme-resolution or extreme-scale WRF simulations on different platforms. The most significant ones include the Hurricane Sandy landfall simulations performed on Blue Water at National Center for Supercomputing Applications (Johnsen et al., 2013), the nested high-resolution heavy rainfall simulations on IBM Blue Gene/P (Malakar et al., 2012), the idealized rotating fluid simulations on a dry atmosphere on BlueGene/L at IBM Watson and Lawrence Livermore National Laboratory (Michalakes et al., 2007).

2. MODEL SETUP

2.1 *Target domain and expected resolution*

The target area mainly discussed and displayed in this paper is a cylinder area centered at ORD. The diameter of the area is about 224 kilometers (over 120 nautical miles); the height of the area is over 21 kilometers (about 70,000 feet). It is depicted on a screenshot of Google Map shown in Fig. 1.

The expected resolution in our simulation is about 167 meters in the horizontal direction and about 90 meters on average in the vertical direction. To create the dataset for high-resolution animation, the interval of data collection is set to be 3 seconds, which allows us to produce around 1,200 frames for one-hour simulation.

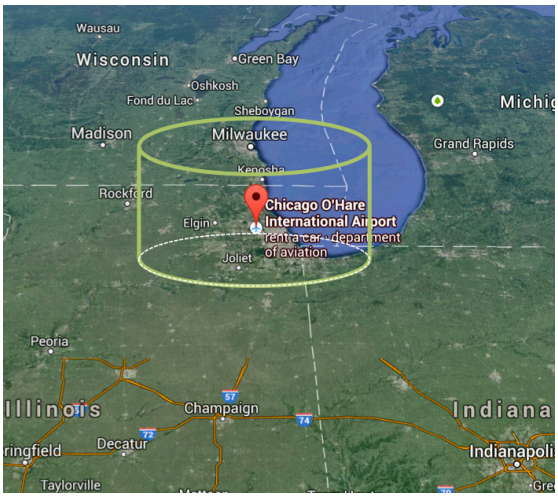

Figure 1: The target area of O'hare International Airport, Chicago, Illinois.

2.2 *WRF model setup*

The WRF model can be built and run in serial mode, shared-memory parallel mode, distributed-memory parallel mode, and hybrid mode. In our research, most of the simulations are realized by an optimized WRF model (version 3.5.1) with distributed memory parallelism. The model is built with Intel-13 compiler and MVAPICH2 library. To satisfy the requirement of the resolution in the target area and to cover the complete area of our interest, we create a mesh with 1345×1345 grid cells in the horizontal direction and 234 vertical levels, which also defines the dimension of the output data for visualization: 1345×1345×234.

To optimize the computation of the high-resolution simulation, we apply a one-way nested WRF model simulation with 3:1 nesting ratio in the horizontal direction and 1:1 nesting ratio in the vertical direction. A coarse-grid (parent) run is implemented independently prior to a fine-grid (child) run to create more accurate initial and lateral boundary conditions for the fine-grid run. The fine-grid run, demanded by expected high resolution, is then executed and yields a huge amount of

* *Corresponding author address*: Si Liu, Texas Advanced Computing Center, Austin, TX, 78758; e-mail: siliu@tacc.utexas.edu

output data for analysis and visualization. The original input data for the coarse-grid runs is obtained from Global Forecast Systems (National Climatic Data Center, 2015) produced by the National Centers for Environmental Prediction. The plane graph of both the parent and child domains is displayed in Fig. 2.

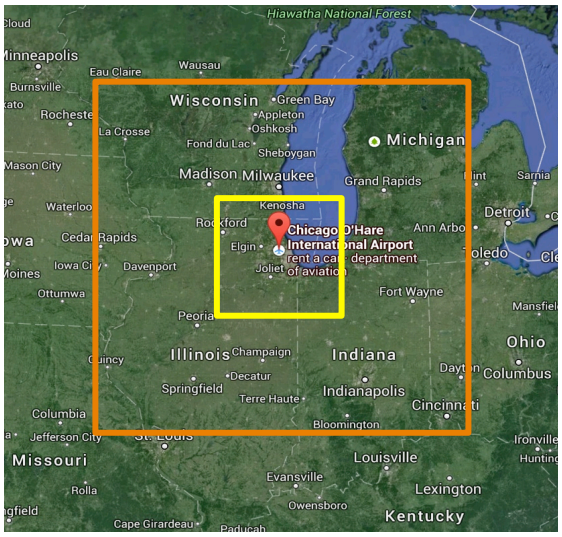

Figure 2: The plane graph of the parent and child domain.

WRF, like many other weather and climate models, nicely supports domain decomposition in the horizontal direction. Thus, it is 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. However, there is no vertical partition in the WRF model. High resolution in the vertical direction significantly increases the size and complexity of each single subdomain, which inevitably aggregates difficulties to the model simulation.

In this paper, we mainly 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 2010-06-23-23:00:00 to 2010-06-24-00:00:00. Due to the high resolution in both horizontal and vertical directions, the choice of the timestep is 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 obeys the Courant-Friedrichs-Lewy criteria and maintains the numerical stability.

2.3 *Computing and visualization resources*

In this project, high-resolution simulations, data processing, and visualization are carried out on TACC's Stampede Supercomputer (Texas Advanced Computing Center, 2015c). Stampede is a Dell Linux Cluster with 10 PFLOPS peak performance. It has over 6,400 compute nodes, each equipped with two Xeon E5-2680 processors and one or two Intel Xeon Phi coprocessors. There are additional 16 large-memory nodes with 1 TB of memory on the system. In additional to the Stampede system, a number of preliminary simulations are carried out on NCAR's Yellowstone supercomputer (Computational and Information Systems Laboratory, 2015) for test purposes.

3. HIGH-RESOLUTION SIMULATION

3.1 *Memory management*

The most common and critical problem in such a high-resolution simulation is memory limitation, though modern supercomputers offer much more memory resources than traditional computing machines. To understand the memory usage pattern and estimate the memory usage quantity, we use ***TACC Stats Tool*** (Texas Advanced Computing Center, 2015b) to collect system-wide performance data, especially CPU usage and memory usage information during the WRF simulations (Lu et al., 2013). Two significant programs in the one-way nested WRF simulation are ***ndown.exe*** and ***wrf.exe***. The former is the one-way nesting program that creates the initial and lateral boundary condition files for the fine-grid run. The latter is the numerical integration program that completes both coarse-grid and fine-grid simulations and generates all output data files. Fig. 3 shows the free memory and used memory quantities throughout typical runs of these two programs. From this figure, we can observe that each task needs a huge amount of memory during the computation. In addition, the first MPI task frequently requires extra memory than others.

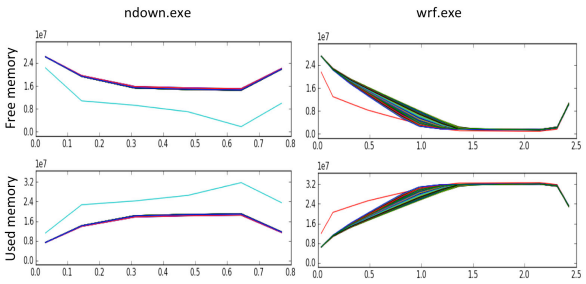

Figure 3: Monitoring free memory (top) and used memory (bottom) of ndown.exe (left) and wrf.exe (right) throughout typical simulation runs on Stampede by TACC Stats Tool. Each line represents the quantity of free or used memory on one compute node. The line that significantly deviates from the others indicates the additional memory cost of the first task.

Each Stampede's normal compute node provides 32 GB of memory and supports up to 16 MPI tasks. In order to meet the memory requirements throughout the

simulation, a conventional method is to reduce the number of MPI tasks per node as shown in Fig. 4. Based on this method, we could carefully control the number of MPI tasks on every compute node to provide sufficient memory to these MPI tasks.

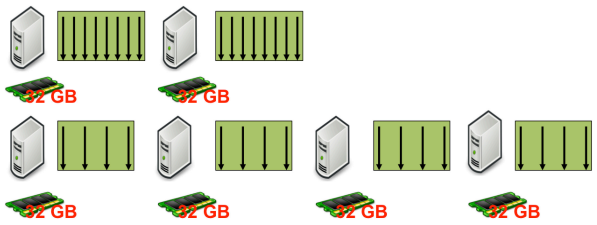

Figure 4: Reducing the MPI tasks per single compute node to offer more memory to each MPI task.

We notice that WRF, like many other parallel applications, requires more memory to serve the first MPI task. Therefore, we can further allocate one dedicated compute node for this MPI task when submitting WRF jobs to the Stampede system. If the first MPI task requires more memory than that provided by a normal compute node, it is necessary to reconfigure the SLURM scheduler (SchedMD LLC, 2015) on the Stampede supercomputer. Most modern supercomputers provide a number of large-memory nodes in addition to normal compute nodes, but almost all systems treat them as two different kinds of resources and manage them through different partitions or queues. By reconfiguring the Slurm scheduler, we are able to add one large-memory node with 1 TB of memory into the resource pool of normal compute nodes to host memory-intense task(s). A diagrammatic sketch of this reconfiguration is shown in Fig. 5.

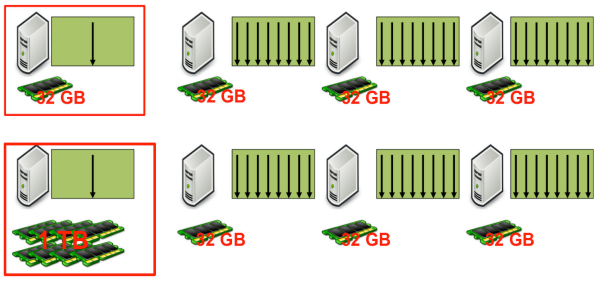

Figure 5: Allocating one dedicated compute node for the first MPI task (top) and adding one large-memory node into the resource pool of normal compute nodes through SLURM scheduler reconfiguration to host memory-intensive tasks (bottom).

These memory management techniques assist us to provide **ndown.exe**, **wrf.exe**, and all related programs with enough memory resources throughout the simulation.

3.2 *I/O workflow*

Basic I/O workflow WRF supports several different I/O mechanisms. The most traditional one is the "spokesman" sequential I/O method. In this method, the model only makes use of one single process to read or write initial and lateral boundary data files, restart files, and output files throughout the simulation. Though this method is easy to implement and manage, it is evidently not efficient nor scalable. The time spent in I/O operations increases dramatically as the problem size increases. It is a waste of computing resources, as all compute nodes have to wait for the completion of the I/O work. Consequently, this method is not feasible at all in our I/O-intensive simulation.

One alternative method is parallel I/O with independent data files, i.e. all processes perform I/O to individual data files on the parallel file system. The performance of this method is very satisfactory when the number of processes is limited. However, this method causes the bottleneck of metadata operations. On a parallel Lustre file system, a great deal of data file access produces contention to the Meta Data Server (MDS) and Object Storage Targets of the Lustre file system. Particularly, the heavy load to the MDS dramatically slows down the overall I/O performance and can even crash the whole Lustre file system.

Another possible I/O method is MPI collective parallel I/O with shared data files. A few processes are chosen to be parallel I/O aggregators, which carry out I/O operations on the data files. Similar to the previous parallel I/O method, all processes are involved in the I/O work. However, only one data file is generated per timestep and this data file will be accessed by the aggregators, rather than all processes. This method normally needs additional support of the parallel versions of HDF5 (The HDF Group, 2015) or NetCDF (Li et al., 2003; Unidata Program Center, 2015) and advanced tuning or optimization. Some parallel I/O architecture and query processing systems over HDF5 and NetCDF are also developed to improve the I/O performance. Please see reports (Su et al., 2013; Wang et al., 2014; Wang et al., 2013; Wauteleta and Kestenera, 2011; Yu et al., 2006) and references therein for more details. High efficient performance is expected if the whole program is properly optimized. However, optimizing the parallel performance of this method is generally not trivial, because there are so many parameters that affect the performance, and many optimal choices of these parameters are system-dependent or application-dependent. Some user friendly approaches for tuning parallel file operations have been developed by Lofstead et al. (2009), McLay et al. (2014), Zimmer et al. (2013) and others. Specifically, TACC has created the T3PIO library and obtained satisfactory optimized results with several WRF Benchmark runs (McLay et al., 2014). With the T3PIO

library, only limited source code modifications and optimization work are required, which will make this method more practical and convenient.

Advanced I/O workflow In our target simulation, each output file for visualization covers over 400 million grid points in three dimensions and contains dozens of different physical quantities. More than 10 GB of disk space is required per timestep to record these data indispensable for visualization. To generate the animation and observe the weather changes in a short period of time, we record the output data every 3 model seconds. As a result, as many as 1, 200 different states are recorded in one model hour, which yields more than 14 TB of data.

In our advanced I/O workflow, we modify the parallel I/O method with independent data files. On the Stampede system, each compute node has a local hard drive of around 80 GB. Around 64 GB of the space is available to applications running on the nodes. Instead of writing thousands of files directly to the Lustre parallel file system, we first write these data files to the local disk on compute nodes as shown in Fig. 6. A ***collection*** program developed by the first author runs along with the normal WRF simulation. This ***collection*** program
– packs all data files into one single tarball per compute node
– sends the tarball to the Lustre file system
– cleans the packed data files on the local disk.

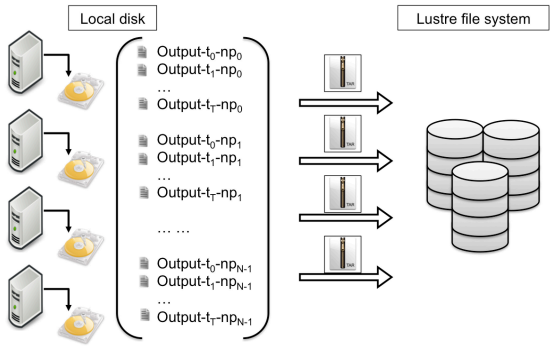

Figure 6: Advanced I/O workflow used in the high-resolution simulation for output data collection: output data files are written to the local disk on each compute node and then packed as one tarball file per compute node. Those tarball files are then sent to the parallel Lustre file system.

Since only one tarball file is written to the Lustre file system per compute node, and data transfer from the local disk to the parallel file system only happens limited times during the entire simulation, the total number of MDS requests is dramatically reduced. Furthermore, since the I/O work is implemented by all processes and the data transfer work is implemented by all compute nodes, the method is very efficient and scalable. In one of our typical runs with 1024 CPUs on 128 Stampede's compute nodes, the sequential I/O method takes about 10 minutes per timestep to write to the data file, i.e. more than 8 days for one-hour model simulation, whereas our advanced methods only need about half a second on average to complete the I/O work per timestep, i.e. about 10 minutes for one-hour model simulation.

Other I/O techniques in our simulation We are only interested in a number of target two-dimensional and three-dimensional variables including the radar reflectivity, cloud water content, cloud ice content, snow content, graupel content, etc. Therefore, we compile a specific version of WRF with a modified WRF registry file (EM COMMON), which reduces the output file size by 30-50%. In addition, we also routinely adopt the WRF checkpoint and restart mechanism in order to make all simulations complete within a reasonable wallclock limit, which reduces the risk of job failure and data loss.

3.3 *Data processing and visualization*

Data processing During the simulations, we validate output data by examining the number of data files and their sizes. After the simulation finishes, we unpack these tarballs. To facilitate the data processing and visualization procedure, we also regroup these data files from process-based to timestep-based, i.e. all data files for the same timestep are grouped together after this data processing procedure.

For long-term storage convenience and possible visualization applicability, it may be worthwhile to merge split data files to one single file for each timestep, though this is totally optional for our visualization work. To merge the NetCDF data files generated in our simulation, we employ the Advanced Regional Prediction System (ARPS) (Center for Analysis and Prediction of Storms, 2015) developed at the Center for Analysis and Prediction of Storms at the University of Oklahoma. One sequential ARPS job is necessary for each timestep and this kind of jobs is memory-intensive due to the problem size. Therefore, we have to implement these ARPS data merging jobs on the large memory nodes of Stampede. Since there are only 16 large-memory nodes on Stampede with 1 TB of memory per node and these nodes are shared by all Stampede users, we need to carefully schedule our data merging jobs. ***TACC Parametric Job Launcher*** (Texas Advanced Computing Center, 2015a) helps us to submit multiple sequential applications simultaneously to the Slurm Scheduler. The only things that need to be pre-set are
– the total number of nodes required for applications
– the maximum number of applications to run per node
– a complete list of applications and parameters.

With the ***TACC Parametric Job Launcher***, we not only schedule thousands of memory-intense jobs conveniently, but also avoid overusing the resources on a shared system. The complete data processing

procedure is shown in Fig. 7.

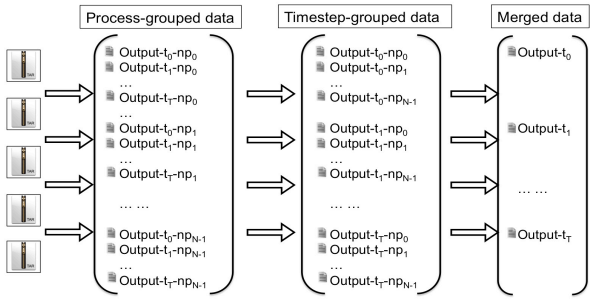

Figure 7: Data processing procedure after simulation runs: tarball files are unpacked and regrouped on the Lustre file system. Split data files will be merged to a single data file per timestep if necessary.

Visualization WRF output files use geopotential height values to identify altitude, whereas visualization software requires coordinate values in the height axis. Accordingly, we create Python programs to convert geopotential height values to height coordinates and convert the NetCDF data files to VTK files. Resulting VTK files are then read into ParaView (Kitware, Inc.) to create visualization images and animation. After rendering, resulting segments and variations are chosen for a video edited with script-controlled commands from the ImageMagick toolkit. Gimp is used for various other tasks, such as titles and more detailed compositing.

For a generalized aviation reference, an aviation map provided by Raytheon Company is included for background in the images and animation results. This aviation map is not for accurate orientation in the current version, but to aid in a viewer's interpretation that the visualization is about weather pertaining to aviation. A current, in-progress version of this project has incorporated accurate projection of texture-mapped aviation information into the visualization results.

We present three figures in this paper. Fig. 8 and Fig. 9 show scenes of rain water mixing ratio at early and late timesteps in the simulation. Rain content is modeled as an isosurface from a specific value chosen for visual interest. The surface is clipped at approximately 14,000 feet altitude to show interior values. The clouds outer surface represents a qrain value of 0.0001, while the sliced off top reveals qrain values from grey and white for the lowest values, shades of green for middle values, and the warmest colors to red for the highest.

In Fig. 10, wind velocity is modeled as streamlines, colored by magnitude (low values (blue) to high values (red)). Reflectivity is also incorporated as a slice close to ground level and colored with a simplified version of the weather maps conventional color key. The lowest values are colored black, later rendered transparent in a composite, so that the aviation-map-styled ground would show through.

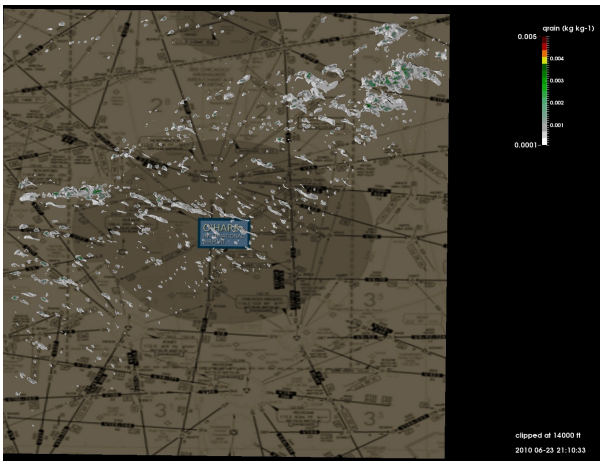

Figure 8: A scene of rain water mixing ratio around the target area at 21:10:33, on July 23, 2010, clipped at 14,000 feet.

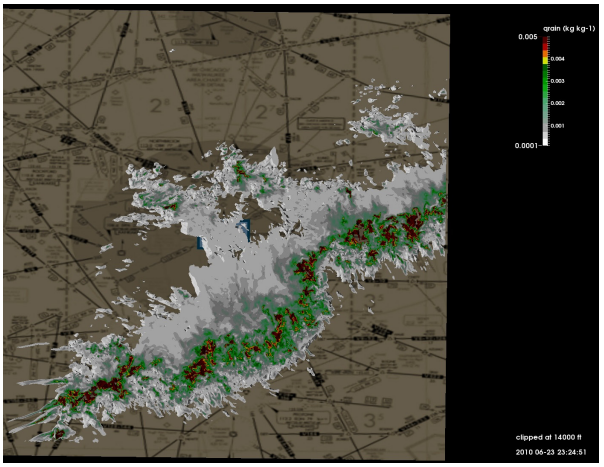

Figure 9: A scene of rain water mixing ratio around the target area at 23:24:51, on July 23, 2010, clipped at 14,000 feet.

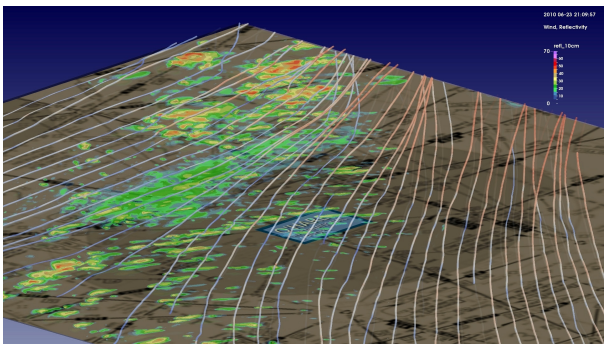

Figure 10: A scene of wind and reflectivity around the target area at 23:09:57, on July 23, 2010.

4. CONCLUSION AND FUTURE WORK

In our research, we have modeled a specific region and time frame to produce meteorological data with extremely high resolution. The resolution of our simulation in both time and space is beyond almost all similar weather simulations as we are aware of. Benefiting from the crucial resolution, meteorologists are able to observe subtle weather changes at local areas. Furthermore, all these techniques are applicable to a great deal of memory-intensive and/or I/O-intensive applications, high-resolution simulations, and other supercomputer platforms.

Our Raytheon Company and NCAR collaborators are comparing the simulation results with other observational and computational results for further validation. Some follow-on studies will be performed over other domains of our interest. We will also investigate WRF's performance benefits from Intel Xeon Phi coprocessors as well as advanced parallel I/O applications in the near future.

5. ACKNOWLEDGEMENT

This is a cooperative project of Raytheon Company, Texas Advanced Computing Center, and National Center for Atmospheric Research. The authors gratefully acknowledge the support from the Raytheon Company.

We would also like to express our special thanks to Ming Chen, David Gill, and Jordan Powers from NCAR for their support of the WRF simulation; Bill Barth, Doug James, Tommy Minyard, and Todd Evans from TACC for their support of the TACC resources and TACC Stats; and Yunheng Wang from Center for Analysis and Prediction of Storms for the support of ARPS.

6. REFERENCES

Center for Analysis and Prediction of Storms, cited 2015: Advanced regional prediction system. [Available online at http://www.caps.ou.edu/ARPS]

Computational and Information Systems Laboratory, cited 2014: Yellowstone: IBM iDataPlex system. [Available online at http://n2t.net/ark:/85065/d7wd3xhc]

Johnsen, P., Straka, M., Shapiro, M., Norton, A., and Galarneau, T., 2013: Petascale WRF simulation of hurricane Sandy deployment of NCSA's cray XE6 blue waters. *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis*, 63.

Li, J. and Coauthors, 2003: Parallel netCDF: A high-performance scientific I/O interface. *Proceedings of the 2003 ACM/IEEE conference on Supercomputing,* 39.

Lofstead, J., Zheng, F., Klasky, S., and Schwan, K., 2009: Adaptable, metadata rich IO methods for portable high performance IO. *IEEE International Parallel & Distributed Processing Symposium (IPDPS 2009)*,1-10.

Lu, C. D. and Coauthors, 2013: Comprehensive job level resource usage measurement and analysis for XSEDE HPC systems. *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*, 50.

Malakar, P., Saxena, V., George, T., Mittal, R., Kumar, S., Naim, A. G., and bin Hj Husain, S. A., 2012: Performance evaluation and optimization of nested high resolution weather simulations. *Euro-Par'12 Parallel Processing*, *Lecture Notes in Computer Science,* **7484**, 805–817.

McLay, R., James, D., Liu, S., Cazes, J., and Barth, W., 2014: A user-friendly approach for tuning parallel file operations. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 229–236.

Michalakes, J. and Coauthors, 2007: WRF nature run. *Journal of Physics: Conference Series*, **125**, 012022.

National Climatic Data Center, cited 2015: Global forecast system. [Available online at http://www.ncdc.noaa.gov/data-access/ model-data/model-datasets/global-forcast-system-gfs]

SchedMD LLC, cited 2015: Slurm workload manager. [Available online at http://slurm.schedmd.com]

Su, Y., Wang, Y., Agrawal, G., and Kettimuthu, R., 2013: SDQuery DSI: Integrating data management support with a wide area data transfer protocol.

Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 47.

Texas Advanced Computing Center, cited 2015a: Parametric job launcher. [Available online at https://github.com/TACC/launcher]

Texas Advanced Computing Center, cited 2015b: TACC Stats. [Available online at https://github.com/TACC/tacc_stats]

Texas Advanced Computing Center, cited 2015c: Stampede user guide. [Available online at https://portal.tacc.utexas.edu/user-guides/stampede]

The HDF Group, cited 2015: HDF5 home page. [Available online at http://www.hdfgroup.org/HDF5]

Unidata Program Center, cited 2015: NetCDF downloads. [Available online at http://www.unidata.ucar.edu/downloads/netcdf/index.jsp]

Wang, Y., Nandi, A., and Agrawal, G., 2014: SAGA: Array storage as a DB with support for structural aggregations. *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, 9.

Wang, Y., Su, Y., and Gagan, G.A., 2013: Supporting a light-weight data management layer over HDF5. *Cluster, Cloud and Grid Computing (CCGrid), 13th IEEE/ACM International Symposium on IEEE*, 335–342.

Wauteleta, P. and Kestenera, P., 2011: Parallel IO performance and scalability study on the PRACE CURIE supercomputer*. White paper, Prace*. [Available online at http://www.prace-ri.eu/IMG/pdf/Parallel_IO_performance_and_scalability_study_on_the_PRACE_CURIE_supercomputer-2.pdf]

WRF Development Teams, cited 2015: The weather research and forecasting model. [Available online at http://www.wrf-model.org/index.php]

Yu, H and Coauthors 2006: High performance file I/O for the Blue Gene/L supercomputer. *High-Performance Computer Architecture, The Twelfth International Symposium on IEEE*, 187–196.

Zimmer, M., Kunkel, J.M., and Ludwig, T., 2013: Towards self-optimization in HPC I/O. *Supercomputing, Lecture Notes in Computer Science*, **7905**, 422–434.