

An Integrated Simulation and Visualization Framework for Tracking Cyclone Aila

Preeti Malakar^{1*}, Vijay Natarajan^{*†}, Sathish S. Vadhiyar[†], Ravi S. Nanjundiah[‡]

^{*}Department of Computer Science and Automation

[†]Supercomputer Education and Research Centre

[‡]Centre for Atmospheric & Oceanic Sciences

Indian Institute of Science, Bangalore, India

preeti@csa.iisc.ernet.in, vijayn@csa.iisc.ernet.in, vss@serc.iisc.ernet.in, ravi@caos.iisc.ernet.in

Abstract—Numerical simulation plays an increasingly important role in many scientific and engineering applications like climate modeling. One of the important problems in climate science is forecasting and tracking cyclones. Large-scale and long-range forecasting of cyclones involves high amount of parallel computations with large data sets, subsequent data analysis and high-resolution visualization for scientific discovery. In this work, we have developed an adaptive framework that performs simultaneous simulations and online visualization for forecasting and tracking cyclones. Our framework contains automatic mechanisms for identifying cyclones, adaptively changing the resolutions of forecasts and visualizations, the resource configurations for executions and the amount of data for visualization based on the cyclone characteristics. We have used our framework for tracking a tropical cyclone, *Aila*, in the Indian region.

I. INTRODUCTION

Cyclones, accompanied by torrential rains and powerful winds, cause extensive damage to lives and property. Hence, accurate predictions of cyclone characteristics and movement are imperative. Such accurate weather forecasts require high-fidelity simulations [1] with complex numerical models that involve petascale computations generating petabytes of data. Visualization is vital for subsequent data analysis and to help scientists comprehend the huge volume of data output. Visualization can be helpful in identifying important aspects of the modeled region. For example, the development of low pressure or the appearance of high vorticity can be easily detected.

¹Student author

Since high-fidelity simulations are computationally intensive and run for long hours, it is necessary to perform *in-situ* visualization [3]. This is even more important for cyclone simulations because the visualization output can provide advance information with respect to development of the cyclone. This allows adequate time to take preventive measures. Simultaneous execution of simulation and visualization steps not only reduces the turn-around time for visualization but also obviates the need for bulky storage. Hence, a framework that performs simultaneous simulations of cyclones and visualizations of pertinent forecast output is essential for climate scientists.

Cyclone simulations also involve various application dynamics. In order to obtain reasonably accurate results, the resolution with which a cyclone is simulated should depend on the intensity of the cyclone and should be varied at different times. Since different resolutions result in different amounts of computations, the number of processors used for parallel simulations will have to be dynamically changed during the simulations. The amount of data transferred from the simulations and visualized by the visualization engine should also be dynamically adjusted to the different intensity levels of the simulations. Thus, the framework should also adequately deal with the dynamics introduced by the cyclone forecasting system.

In this work, we have developed an integrated framework for simulation and visualization for tracking cyclones. Our framework automatically transfers data between the simulation and visualization components, invokes the visualization routines, and dynamically changes the resolution of simula-

tion, the number of processors used for simulations, and the number of time slices visualized. We have used the framework to simulate the cyclone *Aila* in the Indian region. *Aila* was the second tropical cyclone to form within the Northern Indian Ocean during 2009 [2]. The cyclone was formed on May 23, 2009 about 400 kms south of Kolkata, India and dissipated on May 26, 2009 in the Darjeeling hills. There were 330 fatalities, 8,208 reported missing and about \$40.7 million estimated damage. We simulated *Aila* at a resolution of upto 3.33 km using a mesoscale numerical weather forecast model WRF [6], using upto 54 AMD Opteron™ processor cores. We discuss the results of the experiments with our framework in this report.

II. RELATED WORK

The analysis and study of time-varying output data, obtained from numerical simulations, is integral to the scientific process once simulation has completed. Tiankai Tu et al. [4] proposed tightly-coupled execution of the simulation and visualization components in order to share data as shown in Figure 1. Hence the cost of communicating data from simulation component to the visualization component is minimized. The simulation component runs followed by the visualization component in the same set of processors. Thus there is alternation of simulation and visualization cycles using the same shared data. In their approach, the simulation component is stalled while the visualization component runs since both run on the same set of processors. The simulation component is generally more compute-intensive than the visualization component. Hence, stalling simulation while the visualization component runs would cause the subsequent output of simulation to be produced after some delay.

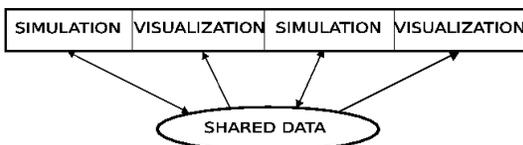


Fig. 1. Tightly-coupled simulation and visualization steps

Storm-forecasting is a relatively difficult problem in climate science. Efforts have been made by scientists at the Center for Analysis and Prediction

of Storms (CAPS) at the University of Oklahoma in this direction [1], [5]. They have conducted a real-time storm-scale ensemble forecasting experiment with the goal of testing multiple runs of a forecast model.

III. METHODOLOGY

The modeled region of forecast is called a *domain* in WRF. There is one parent domain which can have child domains, called *nests*. Regional weather models like WRF model the climate using gridded data where the grid spacing determines the resolution. Decreasing the grid spacing increases resolution which leads to more accurate simulations.

The lowest pressure region is called the *eye* of the cyclone. Hence, to track the eye of *Aila*, we employ a finer resolution nest on the region of our interest inside the parent domain as shown in Figure 2. Our framework forms the nest dynamically based on the lowest pressure value in the domain and monitors the nest movement in the parent domain along the eye of the cyclone.

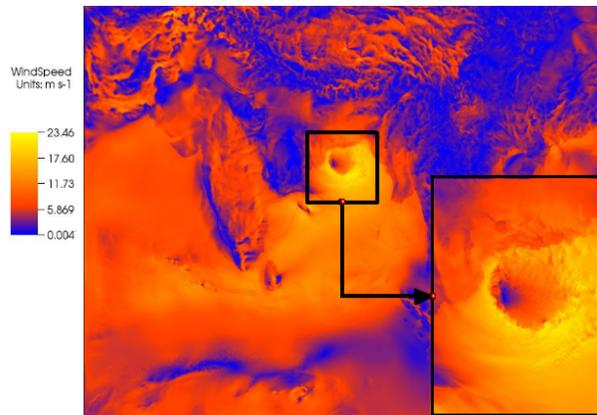


Fig. 2. Windspeed visualization in finer resolution nest inside parent domain

A. Nest refinement

Our framework spawns a nest when the pressure decreases to below 995 hPa. The nest is centered at the location of lowest pressure in the parent domain. As and when the cyclone intensifies i.e. the pressure decreases further, our framework changes the resolution of the nest multiple times to get a

better simulation result from the model. Finer resolution in turn implies more computation¹. So, our framework reschedules WRF on a higher number of processors when the resolution is refined.

B. Amount of Data for Visualization

WRF outputs data in the form of NetCDF [7] files. Each NetCDF file contains the output of a number of simulation timesteps. With finer resolution, the output data size per timestep as well as the time to compute each timestep increases. So that the visualization proceeds at the same rate as the simulations, our framework reduces the number of timesteps output per NetCDF file when the resolution is increased. This reduction in number of timesteps per file ensures that the rate of producing each NetCDF file does not increase significantly.

C. Online Visualization

Whenever an output is produced by the simulation, it is sent to the graphics workstation for visualization, thereby enabling runtime visualization and reducing storage space requirements. Visualization is performed directly on the NetCDF file without any intermediate processing step. We have visualized the output using volume rendering, vector plots employing oriented glyphs, pseudocolor and contour plots of the VisIt [8] visualization tool.

IV. INTEGRATION FRAMEWORK

The backbone of our integration framework is a co-ordinator process *wrf_monitor* which spawns and refines the nest. The simulation component passes information related to pressure threshold and nest location to *wrf_monitor* through intermediate files. As WRF is a regional model, with each level of refinement, it needs input data at a finer resolution. *wrf_monitor* configures WRF based on the current nest location and invokes the WRF Preprocessing System (WPS) to interpolate the meteorological data onto the changed domain. Based on the current resolution, *wrf_monitor* employs a higher number of processors to run WRF. This is illustrated in Figure 3.

¹For example, 10-km resolution gives better results than 15-km resolution but a 15-km resolution requires a 422×343 grid whereas a 10-km resolution requires a grid of size 643×517 for the same area.

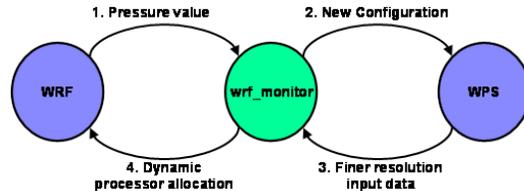


Fig. 3. *wrf_monitor* co-ordinator process

Our framework automatically transfers data between the simulation and visualization components as shown in Figure 4. This is facilitated by the *wrf_transfer* process that runs on the simulation engine host. As and when the simulation component finishes writing to a file, *wrf_transfer* transfers data from the simulation component to the visualization engine.

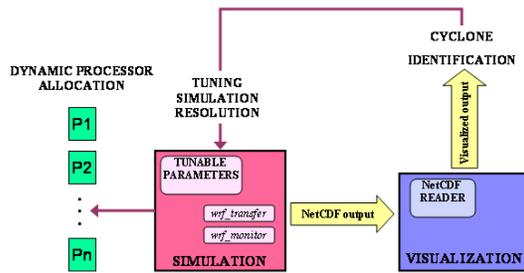


Fig. 4. Integrated Framework for Tracking Cyclones

V. EXPERIMENTS AND RESULTS

We have performed the simulations for an area of approximately 32×10^6 sq. km. from 60°E - 120°E and 10°S - 40°N , comprising of the region of formation and dissipation of Aila over a period of 4 days. The nesting ratio i.e. the ratio of the resolution of the nest to that of the parent domain, was set to 1:3. The 6-hourly 1-degree FNL analysis GRIB meteorological input data for our model domain was obtained from CISL Research Data Archive [9]. We used MPICH2-1.0.8 for running WRF. We utilized upto 54 AMD OpteronTM processor cores operating at 2.6 GHz connected by Gigabit Ethernet.

We started the simulations on 20 processors and increased upto 54 processors with each level of refinement of nest as shown in Table I. It can be observed from the table that the number of timesteps/file sent to the visualization engine is decreased with each level of finer resolution in an attempt to maintain constant output file size and

constant time for visualizing data from a file. This also keeps the rate at which output is produced constant because the amount of computation increases with finer resolution, thereby increasing the time to produce one timestep data.

TABLE I
RESCHEDULING SNAPSHOT DURING THE EXECUTION OF WRF

Pressure value (hPa)	995	994	992	990	988	986
Resolution (km)	24	21	18	15	12	10
No. of Processor cores	24	30	36	42	48	54
Number of timesteps/file	8	7	6	5	4	3
Output file size (GB)	0.79	0.92	1.1	1.3	1.7	1.8

The complete simulation for the period of 22 May 2009, 12:00 hours to 26 May 2009, 12:00 hours executed for a total of 21.5 hours of wall-clock time. The frequency of output of NetCDF files from WRF determined the frequency of visualization. Our framework ensured continuous data transfer from the simulation component to the visualization component and dynamically allocated a required number of processors based on the resolution of the simulation. Change in resolution triggers a reallocation of processors as shown in Figure 5.

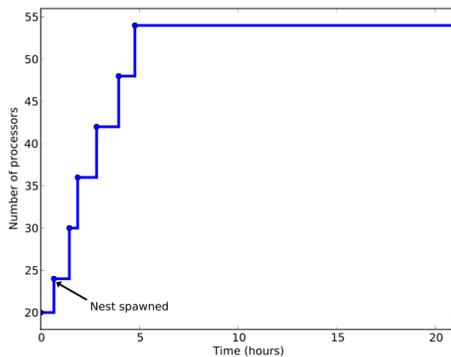


Fig. 5. Change in processor configuration with change in resolution over the duration of simulation

We have developed a plug-in for VisIt to *directly read* NetCDF output files, eliminating the cost of post-processing before data analysis. Visualization was performed on a graphics workstation with

a Intel(R) Pentium(R) 4 CPU 3.40 GHz and an NVIDIA graphics card GeForce 7800 GTX. We used hardware acceleration feature of VisIt. The timings for visualization of different-sized output files is shown in Figure 6. We observed that with hardware acceleration, there is an average decrease of about 72% in the time taken for visualization.

The variables visualized were pressure, wind-speed, rainfall and water vapor mixing ratio. The track of Aila as produced by simulation can be seen in Figure 7. It can be observed from the figure that the depression was formed in the central Bay of Bengal region (around latitude of 14° N) and traversed north-east upto Darjeeling (27° N). The visualization for water vapor mixing ratio along with wind speed vectors is shown in Figure 8. Figure 9 shows the accumulated total cumulus precipitation on 24th May, 2009 as output by WRF.

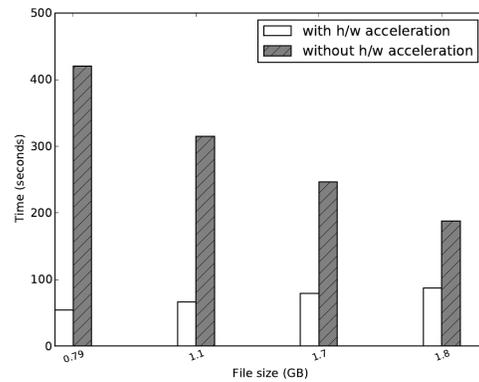


Fig. 6. Comparison of running time of visualizing different-sized output NetCDF files with and without hardware acceleration

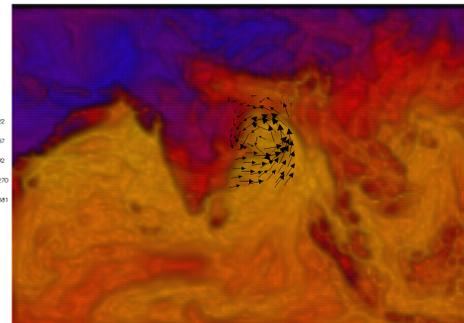


Fig. 8. Visualization of Water Vapor Mixing Ratio with wind vectors shown by using glyphs on the Aila region at 07:30 hours on 24th May, 2009

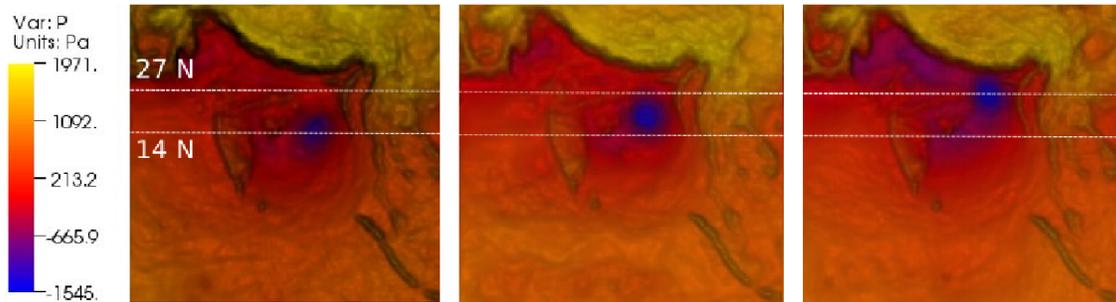


Fig. 7. Visualization of Perturbation Pressure at 18:00 hours on 23rd, 24th and 25th May, 2009

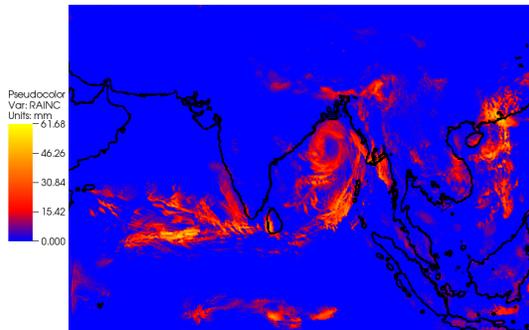


Fig. 9. Visualization of Accumulated Total Cumulus Precipitation on the Aila region at 07:30 hours on 24th May, 2009

VI. CONCLUSIONS AND FUTURE WORK

In this work, we have developed a framework for integrated simulation and visualization of cyclone Aila. In our work, we simultaneously monitored the output without having to wait for the 21.5 hour-simulation to complete. Our framework automatically places a finer resolution nest when the pressure dips beyond a threshold value. We also refined the nest automatically with time as the cyclone intensified, i.e, as the pressure value decreased. In the process, we scheduled WRF on a higher number of processors to cope with the increased computation due to increased resolution. Hence we are able to use multiple processors efficiently depending on computational requirements.

In future work we intend to parallelize the visualization process. We plan to investigate algorithms for adaptive resource sharing between the simulation and visualization components. We also intend to investigate interactive simulation/visualization, so that user input based on the visualization can steer the simulation.

REFERENCES

[1] M. Xue et. al., "CAPS Real Time Storm-Scale Ensemble and High-Resolution Forecasts as Part of the NOAA

Hazardous Weather Testbed 2007 Spring Experiment", *22nd Conference on Weather Analysis and Forecasting and 18th Conference on Numerical Weather Prediction*, 2007

- [2] Aila, http://en.wikipedia.org/wiki/Cyclone_Aila
- [3] Kwan-Liu Ma, Chaoli Wang, Hongfeng Yu, and Anna Tikhonova, "In Situ Processing and Visualization for Ultrascale Simulations", *Journal of Physics (Proceedings of SciDAC 2007 Conference) Volume 78*, 2007
- [4] Tiankai Tu, Hongfeng Yu, Leonardo Ramirez-Guzman, Jacobo Bielak, Omar Ghattas, Kwan-Liu Ma, and David R. O'Hallaron, "From Mesh Generation to Scientific Visualization: An End-to-End Approach to Parallel Supercomputing", *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2006
- [5] Kong et. al., "Real-Time Storm-Scale Ensemble Forecast 2008 Spring Experiment", *24th Conference on Severe Local Storms*, 2008
- [6] Michalakes, J. et. al., "The Weather Research and Forecast Model: Software Architecture and Performance", *Proceedings of 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology*, 2004
- [7] Rew, R. K. and G. P. Davis, "The Unidata netCDF: Software for Scientific Data Access", *6th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, California, American Meteorology Society*, 1990.
- [8] VisIt, <http://www.llnl.gov/visit>
- [9] UCAR CISL Research Data Archive, <http://dss.ucar.edu/>