

Transparent Grid Enablement of Weather Research and Forecasting

S. Masoud Sadjadi¹, Liana Fong⁶, Rosa M. Badia², Javier Figueroa^{1,9}, Javier Delgado¹, Xabriel J. Collazo-Mojica⁸, Khalid Saleem¹, Raju Rangaswami¹, Shu Shimizu⁴, Hector A. Duran Limon⁵, Pat Welsh³, Sandeep Pattnaik¹⁰, Anthony Praino⁶, David Villegas¹, Selim Kalayci¹, Gargi Dasgupta⁷, Onyeka Ezenwoye¹, Juan Carlos Martinez¹, Ivan Rodero², Shuyi Chen⁹, Javier Muñoz¹, Diego Lopez¹, Julita Corbalan², Hugh Willoughby¹, Michael McFail¹, Christine Lisetti¹, and Malek Adjouadi¹

1: Florida International University (FIU), Miami, Florida, USA; **2:** Barcelona Supercomputing Center, Barcelona, Spain; **3:** University of North Florida, Jacksonville, Florida, USA; **4:** IBM Tokyo Research Laboratory, Tokyo, Japan; **5:** University of Guadalajara, CUCEA, Mexico; **6:** IBM T. J. Watson, NY, USA; **7:** IBM IRL, India; **8:** University of Puerto Rico, Mayaguez Campus, Puerto Rico; **9:** University of Miami, Coral Gables, Florida, USA; **10:** Florida State University, Tallahassee, Florida, USA

Contact: sadjadi@cs.fiu.edu

ABSTRACT

The impact of hurricanes is so devastating throughout different levels of society that there is a pressing need to provide a range of users with accurate and timely information that can enable effective planning for and response to potential hurricane landfalls. The Weather Research and Forecasting (WRF) code is the latest numerical model that has been adopted by meteorological services worldwide. The current version of WRF has not been designed to scale out of a single organization's local computing resources. However, the high resource requirements of WRF for fine-resolution and ensemble forecasting demand a large number of computing nodes, which typically cannot be found within one organization. Therefore, there is a pressing need for the Grid-enablement of the WRF code such that it can utilize resources available in partner organizations. In this paper, we present our research on Grid enablement of WRF by leveraging our work in transparent shaping, GRID superscalar, profiling, code inspection, code modeling, meta-scheduling, and job flow management.

Categories and Subject Descriptors

D.4.7 [Organization and Design]: Distributed Systems, Hierarchical Design; K.6.3. [Software Management]: Software Development; D.4.8 [Performance]: Measurements, Modeling and Prediction, Monitors;

General Terms

Measurement, Performance, Design, Experimentation.

Keywords

Grid Enablement, Scientific Applications, WRF, Portal, Meta-Scheduling, Job Flow Management, Modeling, and Profiling.

1. INTRODUCTION

Hurricanes are perilous natural hazards impacting different levels of society thus invoking a need to provide a range of users with accurate and timely information that can enable effective planning for and response to potential hurricane landfalls. The Weather Research and Forecasting (WRF) code [1] is the latest numerical model developed by the National Center for Atmospheric

Research (NCAR) for both operational forecasting and atmospheric research. A measure of its success is that WRF is being rapidly adopted by many meteorological services and researchers worldwide. The WRF was shown to significantly improve forecasts of hurricane structure and rainfall during the Florida Hurricanes of 2004 [2].

The current version of WRF has been designed to run either on a single machine (with one or multiple processors) or on a cluster of homogeneous nodes connected through a high-speed local area network. It has not been designed to scale out of a single organization's local computing resources. However, the high resource requirements of WRF for fine-resolution and ensemble forecasting demand a large number of computing nodes with substantial memory and disk storage, connected through a high-speed network. Currently, few organizations (even national agencies) have either the required computational power or bandwidth to produce high resolution ensembles and deliver them to emergency management, businesses, and the public. Therefore, there is a pressing need for the Grid-enablement of the WRF code so that it can utilize resources available in partner organizations. Florida universities are well positioned to conduct this research effort as they own the Florida Lambda Rail, a statewide high bandwidth optical network (see <http://www.flrnet.org>).

In this paper, we present our research on transparent Grid enablement [3] of WRF by leveraging our work in Transparent Shaping [4] and GRID superscalar [5] as well as other techniques for profiling, code inspection, code modeling [6], meta-scheduling [7,39] and Job Flow Management [8]. We present our preliminary work on the design and development of the *LA Grid WRF Portal* [9] that benefits from all the above mentioned technologies to provide a comprehensive, but customized, Web-based user interface for meteorologists to conduct their hurricane research and to forecast hurricanes in near realtime, as well as for end users (e.g., business owners, home owners, and emergency response teams) to easily access the forecast that is most relevant to them; thereby, enabling them to take more appropriate preventive and recovery actions. The design of this portal will allow the meteorologists to spend their time on the science part of weather research and forecasting, instead of on tedious and error-prone tasks of compiling, installing, and configuring the forecasting software as is currently the case. In addition, the

system administrators can easily add or remove computational resources (e.g., clusters, supercomputers, and virtual machines) that are allocated dynamically to the various simulation tasks based on preplanned high-level policies, or pressing emergent needs.

We have established a state-wide consortium, the Florida Ensemble Forecast Enterprise, which aims to provide high-resolution, location-specific threats (i.e., hurricane landfall location / intensity, winds, gusts, quantitative rainfall and its spatial distribution, inland flooding, etc.) from land falling hurricanes over the state of Florida. The forecasting system will consist of high-resolution model ensemble predictions using vortex-following nested grids that can resolve the hurricane inner core (eye and eyewall) at ~1 km grid spacing, which is critical in forecasting hurricane intensity [10,11,12], and cover a relatively large area for 3-5 day forecasts. We will conduct numerical experiments to determine the number of ensemble members that are needed. The system also includes very high-resolution (<1 km) model domains over selected coastal regions of interest, which can be run at the time of hurricanes nearing landfall. These high-resolution model outputs can be used as input for further impact studies and products developed for the local urban planning and emergency management communities.

Currently, we are focusing on enabling business owners to make intelligent decisions based on the customized forecasts that they receive on the hurricane track/intensity, landfall location/timing, flooding and quantitative/spatial distribution of precipitation with a lead time up to 72 hours for a land-falling hurricane. The forecast results can also be utilized to provide both interactive and non-interactive weather data visualizations, enabling business owners to either navigate across their particular areas of interest over topographical maps (interactive) or employ the pre-generated videos (non-interactive MPEGs/GIFs) to effectively plan and prepare for impending weather hazards. Research on the design and development of a software artifact in the form of an Embodied Conversational Agent (ECA) or avatar will also be conducted to facilitate the human-computer interaction. The avatar agent will receive as input the data from weather simulation on zip-code scale hurricane forecasts and generates as an output a video clip of an avatar “weatherperson” describing the effect of the hurricane, customized to specific business owner’s properties, even if they are in different locations throughout the state of Florida.

The ongoing research presented in this paper is being carried out jointly by several participating universities/institutes spread throughout the state of Florida (i.e., Florida International University, Florida State University, University of North Florida, University of Miami, Embry-Riddle Aeronautical University and Florida Institute of Technology); we also continue our collaboration with IBM TJ Watson, National Oceanic and Atmospheric Administration, and the National Weather Service.

2. Grid Enablement

Grid-enabling is the practice of taking existing applications, which currently run on a single node or on a cluster of homogeneous nodes, and adapt them (either automatically or manually) so that they can be deployed over non-homogeneous computing resources connected through the Internet across multiple organizational boundaries (e.g., multiple clusters from

different organizations) without major modifications to the underlying source code. We consider a Grid-enablement process to be successful if the resulting Grid-enabled application “performs better” than the original application. The better performance may be interpreted differently (e.g., the speedup gained for the wall clock execution time or the efficiency of resource usage).

Grid-enabling involves distributing a set of tasks on a number of heterogeneous nodes located on different administrative domains, collecting the results produced by the nodes and integrating the results as if the original application was running on the original execution environment. There are a number of challenges in this area. Facilities for resource matching are needed whereby the requirements (e.g., hardware, software and time constraints) of a set of tasks are matched against a set of advertised resources. Capabilities for predicting the amount of execution time required by an application are also needed. Such predictions can determine if a time constraint requirement can be accomplished for a given resource match. Resource scheduling and load balancing facilities are needed to enable sharing a set of nodes among a number of running applications. Finding the optimal number of nodes is another important issue whereby it can optimize for both the total execution time of the applications and the resource usage.

In the context of Grid-enablement of WRF, we aim to address the following challenges that we have identified so far. First, the high latency of Internet compared to high-speed LANs is unsuitable for the real-time requirements of the current WRF implementation¹. For example, we cannot simply use domain decomposition for Grid enablement of WRF due to the significant communication overhead involved when the boundary Grid points assigned to run on the resources in one organization need to communicate with their neighboring grid points assigned to a partner organization potentially across the Internet. Second, the high overhead of the Grid middleware software, such as Globus Toolkit (GT4) and Commodity Grid (CoG) Toolkit coupled with the inefficiency of the meta-schedulers are inhibitors for Grid-enabling WRF as a real-time application. Third, manual and ad-hoc modifications to the WRF source code risk compatibility with future WRF versions. Hence, we need a systematic approach for Grid enablement of the WRF code, where modification is transparent to the original code.

3. Application & Scenario

We are currently in the process of Grid-enablement of the Weather Research and Forecast (WRF) model. Specifically, we are using the WRFARW v2.2 numerical modeling. This model incorporates 165,000 lines of source code in FORTRAN and C and another 40,000 lines are generated at compile time. So far, we have promising results with regard to ensemble forecast generated from a suite of randomly perturbed initial/boundary conditions and physics variations. The LA Grid WRF Portal (introduced in Section 7) is expected to hide all the details of the ensemble run from the end users, in this case the meteorologists and end users.

We plan to develop a three-layer nested ensemble forecast. The largest domain is a rectangle of 15 km resolution and includes

¹ WRF can tolerate network delays of up to 100 microseconds and the current Internet delay is around 1 millisecond.

Florida in its center, part of our neighboring states, Caribbean, and part of Atlantic Ocean. The second domain is a rectangle of 5 km resolution that surrounds Florida closely. Finally, the third (innermost) domain contains one or more rectangles of a finer resolution of 1 km (or even higher) over some selected zip-code areas in Florida.

Initially, the composition of ensemble member models will only include the random perturbation of initial conditions using the NCAR WRF ensemble software. We will start with 6 member models and run forecast up to day3 with a horizontal resolution of 15 Kilometers. Afterwards, we will include more variations in the composition of ensemble members by changing the physics, boundary conditions, resolution and dynamics of the models. We will validate the ensemble forecast against the observations through skill scores, error statistics and spatial distribution of various forecasted products such as rainfall, storm track, land fall location/time, and storm intensity in order to assess the deficiencies in the forecast and simultaneously initiate necessary steps to improve. We will disseminate a number of ensemble forecast products to the public, business communities and other user agencies with a lead time of up to 72 hours for further processing in a real-time mode. These include hurricane landfall location / time, storm intensity, rainfall distribution / quantity, inland flooding events, and spatial wind speed distribution.

In our current ensemble modeling efforts, we have included a suite of global multi-models forecasts from a number of groups such as GFS (US Weather Services), NOGAPS (US Navy), UK-MET (England), GFDL (Princeton) and a number of in house models (i.e., CLIPER,BAM) developed by Hurricane Research Division (HRD) of Atlantic Oceanic and Meteorological Laboratory (AOML) and National Hurricane Center (NHC). We have noted a consistent superior performance in forecast skills from the construction of a multi model super-ensemble [13,14,15]. The proposed suite of high resolution mesoscale multi-models (e.g., WRF ARWs) derived from NCAR and other institutions are expected to provide a similar very high level of skill in hurricane forecasting (i.e., track, intensity, landfall time/location, and rainfall) obtained by invoking multi-model ensemble methodology. Thus we expect to provide a viable forecast to the state of Florida with a reasonable lead time.

Beyond Grid enablement of hurricane modeling across multiple geographic and temporal scales for implementation of real-time ensemble generation, there is the potential for creating a suite of products both distinct and integrated which couple multiple types and scales of model simulations (atmospheric, ocean wave, storm surge, hydrological, socio-economic models, etc.). This capability coupled with user- and application-centric visualizations gives rise to a new class of data analytics for decision support in a proactive sense that has heretofore been unavailable. Ocean wave and storm surge models may also be run on the Grid and are used to better predict impacts on coastal infrastructure, shipping and oil drilling interests in the eastern Gulf of Mexico and along the shoreline as the hurricane continues its eastward track. Mean and ensemble members of the atmospheric, ocean wave, storm surge and hydrological models can be coupled to create predictions on storm movement, intensity, near shore wave heights, storm surge, and flooding forecasts that include both optimistic and pessimistic bounds on the likely outcomes. The various model forecasts may be concurrently produced and visualized with Grid resources to

create customized decision support guidance for emergency management, utility, transportation, debris removal and other interests as the storm makes landfall.

4. Motivation and Impact

An improved hurricane forecasting development for tracks, landfall location/timing, intensity, heavy rains and flooding with a reasonable lead time can save lives and has great commercial value for industries, emergency management and the public. We perceive that major insurance industries, transportation, local government, petroleum exploration, energy utilities, agriculture, housing, retail corporate outlets and even fisheries can significantly benefit from use of such improved forecast products. An accurate forecast with a reasonable lead time will also provide emergency managers with sufficient time for preparation and the ability to make appropriate administration decisions about evacuation initiatives. Therefore, the benefits of this research effort are far reaching and affect all aspects of our society. Our goal is to produce superior forecasting products through the use of multi-member ensemble forecasts, multi-model forecast consensus and the coupling of different types of models (atmospheric, ocean, surge, hydrologic, business, etc.) to train a large number of exceptional students who after their graduation can work on related enterprise-specific applications where their knowledge of ensemble modeling, model coupling and applications would be in high demand. We perceive that the promise of such tailored industry products will elicit the involvement of many private firms as well as public, corporate, industry groups, and software firms. These will certainly involve the creation of high technology jobs in each of the involved participants, and a more robust and hurricane survivable Florida economy.

A higher resolution ensemble forecast requires considerable computational / storage resources. Sufficient local resources are not always available due to high costs associated with maintaining these high-end computing clusters. National authorities do not currently have the required computational power or bandwidth to produce such high resolution ensembles or deliver them to emergency management, businesses, and the public. Within the Grid environment, high resolution model predictions and visualizations of complex fields such as: rainfall, wind and waves can be generated for distinct localities in high detail. Mesoscale and cloudscale model ensemble forecasts run in real time utilizing Grid resources have the potential to enable timely generation of decision support products. Information, data and analytics produced in multiple forms, integrated and delivered in support of planning, response and remediation allow improved estimates of storm impacts and damage to infrastructure such as electrical grids and transportation systems.

Grid enablement for the generation of new and unique products, guidance, and analytics in support of mission critical decision made in a transparent manner, while utilizing a dynamic, heterogeneous and geographically disparate set of computing resources, is one of the strategic goals of our ongoing research in this area. It is a necessary and evolutionary step in the convergence of science and technology for societal benefit. Therefore, there is a pressing need to allow WRF as an example of a state of the art predictive modeling tool to run on Grid computing environments, where resources are shared among different organizations. The initiative of Grid-enablement of WRF would not only facilitate the demand of huge resources for

computation but also reduce the work load for meteorological modelers from indulging in the technical aspect of configuring, porting and running the models in versatile computing platforms. They would rather invest their time and energy in investigating the meteorological forecast and work on the model deficiencies.

5. Grid Middleware

Middleware aims at enabling communication among loosely coupled heterogeneous and distributed systems. Such systems are heterogeneous with regard to one or more of the following characteristics: hardware platform, operating system or programming language. Middleware is used in a diverse group of application domains such as remote database access systems, distributed transaction systems, cooperating systems, and distributed multimedia systems. However, the main idea in all cases is that middleware should provide support to address issues of heterogeneity and distribution. That is, middleware aims at providing facilities for distributed applications to interact in an environment with diverse hardware and operating systems. Hence, middleware is essentially “a layer between network operating systems and applications that aims to resolve heterogeneity and distribution” [16]. Common examples of middleware include CORBA, Java’s RMI and .NET. Significant efforts have been carried out in the development of Grid enabling middleware. A number of Grid middleware platforms such as Globus [17], Legion [18] and Condor [19] have emerged. These platforms have been successful in enabling the execution of applications across multiple administrative domains. Nevertheless, as mentioned in section 2, there are still a number of unsolved issues such as predicting the execution time of an application in a Grid environment. In the rest of this section, we explain our existing work on meta-scheduling and job-flow management and will introduce other tools and utilities used in our Grid-enablement approach in the following section.

5.1 Meta-Scheduling

We use the LA Grid meta-schedulers to utilize heterogeneous resources in different domains. Each domain is in charge of the resource publication and management in a peer-to-peer (P2P) fashion using the LA Grid MetaScheduler Protocol [39]. Considering the dynamic nature of resource availability, it is essential to make proper arrangements to schedule WRF Ensemble runs. Ideally, all of the members of an ensemble run will be executed in the local domain via the help of local schedulers (e.g., SGE, PBS, and Condor). In the case that not enough resources are available to run the model locally, the associated meta-scheduler will try to find a partner site that can accommodate to run the model within the specified time frame.

Our current meta-scheduler implementations at FIU, IBM and BSC are able to interoperate with one another and submit jobs using JSDL [20] to specify the execution environment, computational resources quality of service requirements, and transfer input/output files if necessary. In our current implementation, selecting the resources where the job will execute is done by the local schedulers. The meta-scheduler reports the set of resources that are available for remote job execution. However, there is no hard guarantee that these resources will be reserved for the remote job execution. In the future, we would like to be able to reserve resources in remote domains and get some guarantee that our job will be completed before the deadline.

The FIU and BSC implementations use GT4 [17] as the Grid middleware, and the IBM implementation uses ITDWB [21] as the Grid middleware. The FIU implementation also makes use of Gridway [22] meta-scheduler on top of GT4, to make transparent scheduling in different clusters within the FIU Grid. The BSC implementation is based on the new version of eNANOS [23], which is based on GT4 services. It uses a LA Grid service that extends the broker and a set of WS wrappers for the communication with other meta-schedulers. Several extensions and changes have been done in the internal implementation of eNANOS, especially in resource and job management. In particular, a new scheduling policy (called BestBrokerRANK policy) and an algorithm for aggregating the resource data were implemented. The IBM implementation extends a scheduling product ITDWB for intra-domain scheduling purposes. In the case of an inter-domain scheduling, we use the meta-scheduling protocol to negotiate and submit jobs between domains [39].

5.2 Job Flow Management

In our system, a job flow manager automates the process of WRF Ensemble runs. The flow manager captures the flow of operations that are involved in the Ensemble run, namely the data initialization tasks, WRF model execution tasks and data gathering and post-processing tasks. We use the WS-BPEL [24] standard as the workflow language, where job executions are modeled as graphs. The nodes of the graph represent tasks and the edges represent inter-task dependencies, data flow or flow control. The job flow manager manages the flow control of the tasks, while maintaining the sequencing and concurrency expressed between them. These tasks are always submitted from the flow engine to the local domain meta-scheduler, and monitored using a job submission and monitoring interface. Additionally, a notification interface can be used for sending back job state notifications to the flow engine. Based on resource information collected at the meta-scheduler, tasks are mapped and routed to an appropriate set of resources for execution. This resource mapping is handled entirely at the meta-scheduler, thereby cleanly separating its core functionalities from that of the flow engine.

We are using generic proxies [25] to make our job flow manager fault-tolerant. Fault-tolerant behavior is important in large and long-running workflows where a failure in a task could cause the workflow to fail because of the dependency relationship between tasks in the workflow. To prevent task level failure from propagating to the workflow, recovery actions are performed through a generic proxy, based on some predefined recovery policy. Also, for cases where task execution is slow, execution can be optimized by deploying delayed tasks on different resources using the functionality provided by the underlying meta-scheduler.

6. Tools and Utilities

We have divided the other tools and utilities that we are using for the Grid-enablement of WRF into the following two categories: Transparent Grid Enablement Tools and Profiling Tools.

6.1 Transparent Grid Enablement Tools

Transparent Grid Enablement (TGE) [3] is achieved by combining *GRID superscalar (GRIDSs)* [5] and *TRAP/J* [26]. Each tool provided us with the necessary features for transparent software adaptation from a sequential code to a Grid-enabled one. Inspired

by the *superscalar* processors, GRIDSs provides an easy programming paradigm for developing parallel programs. Similar to superscalar processors that provide out-of-order and parallel execution of machine instructions by bookkeeping their dependencies, GRIDSs provides parallelism to the functions of a program written in a high-level programming language such as Java. Using GRIDSs, a sequential scientific application developed by a scientist is dynamically parallelized in a computational Grid. GRIDSs hides the details such as resource mapping, staging input data files, cleaning temporary data files, task deployment, task scheduling, exploiting task-level parallelism, and exploiting data locality. We note that for many of its responsibilities, GRIDSs depends on other Grid computing toolkits such as GT4, Condor, and others.

TRAP/J is a tool that enables static and dynamic adaptation in Java programs at startup and runtime, respectively. It consists of two GUI-based interactive tools as follows: (1) *Generator*, which generates an adapt-ready version of an existing application by inserting generic hooks into a previously selected subset of classes in the application; and (2) *Composer*, which allows insertion of new code at the generic hooks both at startup or runtime. We note that only the pre-selected classes are capable of being adapted and they are called *adaptable* classes. Adaptable behavior is provided through alternative implementation of adaptable classes, which are called *delegate* classes. To replace alternative parallel algorithms developed using the GRIDSs codes, we use the Generator to make the classes with sequential code adaptable, and then we use the Composer to weave in the parallel code [3].

6.2 Profiling Tools

For profiling we surveyed a myriad of tools that were considered viable for examining WRF code [27]. In order to choose one of them we created evaluation criteria composed of the following points: support for FORTRAN, support for OpenMP and MPI, support for available compilers (e.g., g95, PGI-CDK, Intel iFort), ease of use, documentation, technical support, and cost of application licenses. From over 40 profiling tools, we chose TAU [28], IBM HPC Toolkit [29], and BSC Paraver [30] as they excel in visualization, architecture and technical support, respectively.

Based on our continued need to profile WRF in a black box approach, we developed custom profiling and prediction tools that gather data from benchmarks to later make predictions of WRF runtime based on different computational parameters. These applications are called *amon* and *aprof*, correspondingly.

Aprof is based on a mathematical model of WRF execution behavior that can be used to predict the execution duration on an “unseen” execution platform with an arbitrary configuration of hardware resources. This can then be used by the meta-scheduler to distribute a given workload such that timeliness requirements for task completion can be met. Such a model can also be used to ensure quality-of-service to the user requesting the simulation (i.e., using the portal described in Section 7). To develop the model, a series of benchmark executions of WRF were performed on two separate clusters that are part of LA-Grid: *Mind* and *GCB* (see Section 8 for results). Since *aprof* bases its prediction on previous runs (and the system resources available during those runs), it was crucial to have as many combinations of at least two resources as possible. We chose *number-of-nodes* and *processor clock speed* as the two resources. Perl, Python, and Bash scripts

were used to automate the benchmarking by varying the number-of-nodes and limiting the effective clock speed at individual nodes, as well as to automate the process of executing the benchmark runs, collecting the results, and tabulating them. For limiting the available processing power, the open-source *CPUlimit* [37] application was used. For collecting process execution data, we used the in-house monitoring tool, called *amon*, to capture resource usage data of all processes and submit them to the central prediction system, called *aprof*.

Aprof is designed to (1) repeatedly receive observation data, each of which is monitored as a “criterion” variable and several “explanatory” variables, where the former variable corresponds to the observed execution time and each of the latter variables is composed of one or more resource parameters such as CPU clock, memory size, etc.; (2) apply regression analysis to estimate profile parameters for each target process; and (3) predict a value of criterion variables according to the estimated profile parameters when a set of values of explanatory variables is given. The observation data is incrementally accepted but maintained in a fixed size of program memory because of the linear nature of regression analysis. *Amon* not only captures the observation data but also makes prediction requests to *aprof* by providing the process name and detailed values of explanatory variables. When a new request is received, *aprof* estimates profile parameters based on the latest observation data and predicts an execution time for the specified process.

7. Portal and Visualization

In order to provide users a seamless and easily accessible interface for WRF simulation runs and visualization outputs, we designed the LA Grid WRF Portal. The portal utilizes Web 2.0 technology [31] to provide users with highly interactive Web based interface allowing for collapsing, expanding different sections of the portal along with the choice of defining new elements relevant to their interests. The portal utilizes rich, interactive and user-friendly interface based on Yahoo UI [32] and Google Maps [33] and makes use of Java Server Pages and PostgreSQL [34] for server side processing and storage. Yahoo UI is used for the design and implementation of the portal while Google Maps is used for generating relevant map projection and domain boundaries providing a base map for weather/hurricane data visualization purposes.

Figure 1 shows the architecture of the LA Grid WRF Portal. The portal is aimed at facilitating both the meteorologist and business owner communities to simulate WRF runs for their particular domains of interest and visualize the output related to different weather related parameters such as wind speed, precipitation and cloud water density among others, over the underlying Google Maps based projections. Google Maps allows for a highly interactive underlying map projection, which enables users to navigate as well as zoom-in/zoom-out of their particular domains of interest to better understand the overall weather forecast; hence, assisting them to undertake the necessary precautionary measures against any impending threats.

The portal provides separate interfaces for business owner, technical and meteorological communities based on their respective needs. For the meteorological community, since they would need to identify domains along with the ensemble members for the WRF ensemble runs; the LA Grid WRF Portal provides

them with an interface that allows them to define new ensemble members, combine different ensembles and/or change configuration parameters for these ensembles. The meteorologists define geographic domains and download atmospheric conditions for these domains for a certain instant of time via the widely used Global Forecast System (GFS) [35]. The GFS data is then used as the base for ensemble member generation and parameter configurations. Once defined, the meteorologists neither need to worry about scheduling or defining the clusters/Grid architecture, nor do they need to make manual changes in the WRF configuration scripts and code to generate and schedule a new ensemble run. The portal seamlessly schedules these across the LA Grid architecture via the meta-scheduler layers introduced previously. On the contrary, the business owners are mainly interested in the locations/domains where their assets are located and would like to simulate and visualize weather forecast results for the specific domains relevant to their asset locations. The portal interface for business owners thus provides an interface to visualize the weather forecast for the domains associated with the assets defined by these owners at a prior time. Similar to the meteorologists, the business owners can also add new assets to their profiles via the portal. Upon user request for weather visualization, the portal schedules a simulation run for the domain of interest via the meta-scheduler layer. Once the simulation has finished, it notifies the user of the job status and enables the visualization button for weather forecast visualization. In the event of results being available for a simulation run that had been scheduled at an earlier time, the portal simply notifies the user of the status, upon which the user can visualize the weather forecast data results immediately.

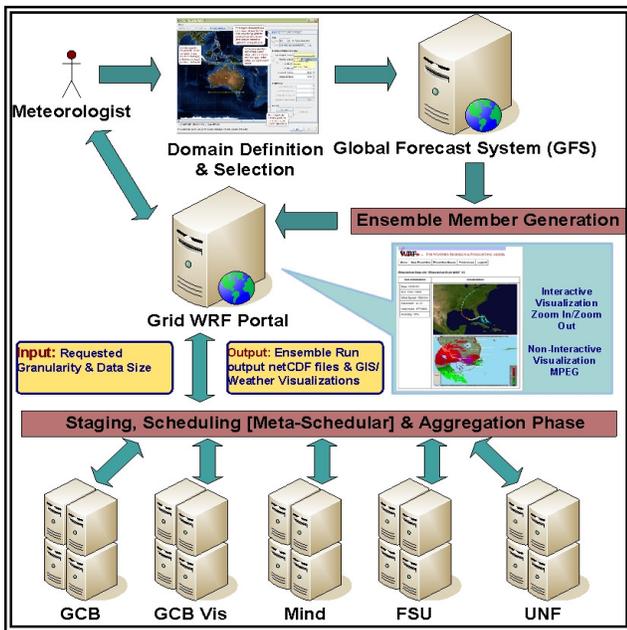


Figure 1. The LA Grid WRF Portal Architecture.

Visualizations for the WRF ensemble runs are achieved via VisAD [36] for interactive visualizations. We plan to incorporate non-interactive MPEG/GIF based visualizations for each of these ensemble runs that will provide the less technology savvy users with a higher level view of the weather conditions. For each of the ensemble runs, the simulation produces a netCDF file relevant to

the domains identified at the time of scheduling. VisAD is an open source Java based library that allows for reading and writing netCDF data files. Upon each successful ensemble run, the portal utilizes JSP based technology to read these netCDF files, incorporates the geo-coordinate information of the relevant data from these files and generates the visualizations corresponding to the meteorologist/business owner's domain.

8. Benchmarking

One of the major expectations from this research is to be able to run an ensemble forecast of hurricane track using the resources available in the Florida WRF Ensemble institutions. To approach the Grid enablement of WRF and to support the optimal utilization of WRF computational resources, for example, we are developing a mathematical model that will help predict the required computational resources based on different demands from the portal users and available Grid resources. In this way, the system will be able to identify the computational resources needed to run a specific forecast or simulation. The following results show how by using our modeling approach we can provide a prediction of the WRF execution time based on processor power and numbers of nodes used in the run. Such a model is also needed to ensure quality-of-service to the user requesting the simulation. To develop such a model, a series of benchmarks were performed on two separate clusters: *Mind* and *GCB*.

Mind is a 15-node cluster consisting of dual-Xeon 3.6 GHz processors, 2GB of main memory, interconnected through a gigabit switch. *GCB* is an 8-node cluster with dual 3 GHz Pentium 4 processors, 1GB of main memory, interconnected through a gigabit switch. The benchmarks consisted of running a WRF simulation with a square area of 75 x 75 km, with a 4km resolution. We focused on varying available processing power and the number of computing nodes for the benchmarks. To obtain as much statistics as possible from the data, the simulation was executed first with all of the nodes of the cluster (for each cluster separately), and successive simulations were run with one less node. The procedure was then repeated by decreasing the percentage of available compute power to 80, 60, 40, 30, 20, and 10 percent. We achieved this by using the CPUlimit tool [37]. Varying the number-of-nodes shows how well the unmodified WRF execution model scales as more compute nodes are added. Varying the available processing power allow us to observe how well the CPU is utilized.

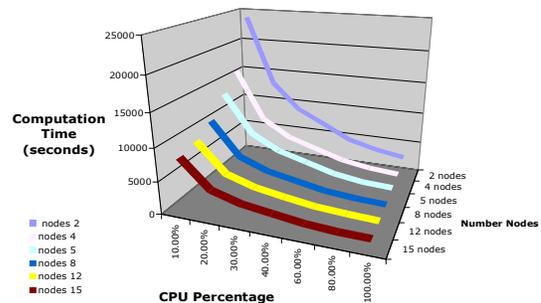


Figure 2. Execution time versus allocated CPU, for 2-15 nodes running on *Mind*. No apparent relationship can be seen.

After performing the previously mentioned benchmarks, we created a preliminary model. Simply looking at the number-of-

nodes and clock speed alone, it was not possible to find an obvious pattern that will help us model the application, as can be seen from Figure 2. This Figure illustrates how due to the increasing number-of-nodes in a WRF run, the performance curve of the application worsens. This worsening of performance is due to several computational variables (e.g., CPU power, number of compute nodes, memory, network latency, network bandwidth, and communication overhead). On the other hand, we found that when the product of the inverse of the CPU speed and the number-of-nodes was compared with the elapsed time, a linear relationship was obtained, as shown in Figure 3 (for brevity, we do not show the graph for the number-of-nodes). It is based on this observation that our profiling software, *amon*, was implemented.

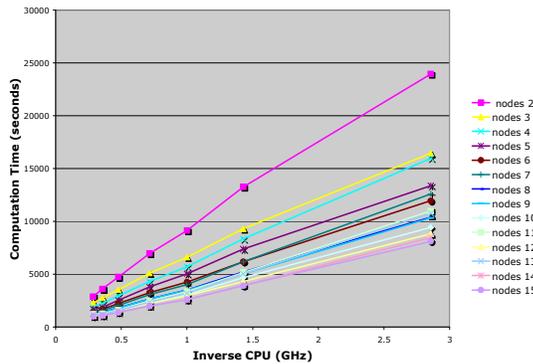


Figure 3. Computation time versus 1/(CPU speed) on *Mind*: A semi-linear relationship is seen.

We are still in the phase of finding which of the computational parameters affect the performance of WRF the most. Using our prediction tool, *aprof*, and data from previous benchmarks, we were able to observe the accuracy of our model. This was done using the CPU power and number-of-nodes parameters, and ignoring all others. The predicted values were very similar to the actual ones (plotted above). The error of the predictions was under 10% in most cases. Figure 4 graphically illustrates expected speedup (mesh), based on the *aprof* model, compared to the actual compute times (+'s), on the GCB cluster.

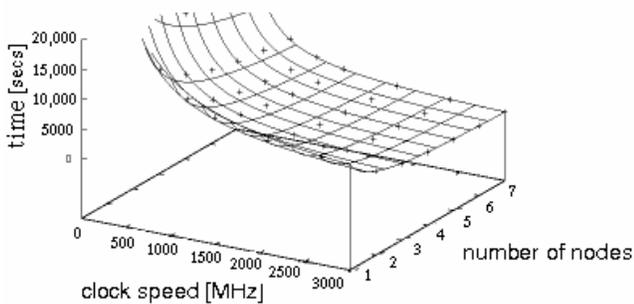


Figure 4. Actual and estimated computation times for GCB's inter-cluster predictions.

9. Conclusion: Lessons Learned!

First, we have learned that Grid-enablement is not a trivial task and there is no well-established or commonly accepted methodology that can walk us through the phases of Grid

enablement. It is almost impossible to provide one solution that fits all. However, we are hoping to be able to achieve a common and general methodology that prescribes a high level guideline for Grid enablement of scientific applications.

Second, we have learned that Grid-enabling a program that has been originally developed for a local cluster has many repercussions. The impact of moving from local to wide area network can be counterproductive even if the number-of-nodes and the processing power increases. Special care on the type and quality of connection between nodes and cluster should be taken. Through profiling, we have been able to measure the speedup gained and sometimes the results were very discouraging.

Third, we have learned that WRF is heavily versed in meteorological concepts. Therefore, even though it was written to be run on a distributed architecture, it is necessary to understand these underlying concepts to avoid assumptions that can lead to error-prone results. In order to better understand the internal intricacies and workings of WRF, we had to dig into the source code. As many scientific applications, WRF is based on FORTRAN and uses programming techniques which are seldom used in modern programming. This was quite challenging as the code base is huge and lacks proper documentation. To deal with this, we made our own documentation that includes the sum of all our understating of WRF, including source code and related papers [38].

Forth, even though our mathematical model for predicting the execution time of WRF is simple, we have learned that it demonstrates a step forward towards optimizing the speedup of Grid-enabling an application. Our results and simulations indicate very little error with some of the cross-cluster predictions, which proves its validity. Still, there are some points in which the error is higher than it should be. By leveraging the data from more clusters that are part of LA-Grid, we will be able to come up with a more assertive model that will provide even more consistent results.

Finally, we note that we are still in the early stages of defining a concrete scenario of ensemble forecast and it is yet premature to claim major success or accomplishments. However, the preliminary results of the portal and ensemble member generation and execution have been very promising.

10. ACKNOWLEDGMENTS

We are thankful to the following individuals for their contributions to some of the ideas presented in this paper: Yanbin Liu, Norman Bobroff, Balaji Viswanathan, Steve Luis, Shu-Ching Chen, Lloyd Trinish, Jason Liu, Alex Orta, T. N. Krishnamurti, Eric Johnson, and Donald Llopis. This work was supported in part by IBM (SUR and Student Support awards), the National Science Foundation (grants OISE-0730065, OCI-0636031, REU-0552555, and HRD-0317692). This work is part of the Latin American Grid (LA Grid) project [39].

11. REFERENCES

[1] The Weather Research & Forecasting Model, <http://www.wrf-model.org/index.php>.

- [2] Patrick T. Welsh and Peter Bogenschütz. Weather Research and Forecast (WRF) Model: Precipitation Prognostics from the WRF Model during Recent Tropical Cyclones, 59th Interdepartmental Hurricane Conference, March 7-11, 2005 Jacksonville, FL.
- [3] S. M. Sadjadi *et al.* Improving separation of concerns in the development of scientific applications. In Proceedings of The Nineteenth International Conference on Software Engineering and Knowledge Engineering, pages 456-461, Boston, USA, 2007.
- [4] S. M. Sadjadi, P. K. McKinley, and B. Cheng. Transparent shaping of existing software to support pervasive and autonomic computing. In *Proceedings of the first Workshop on the Design and Evolution of Autonomic Application Software 2005, in conjunction with ICSE 2005*, St. Louis, Missouri, May 2005.
- [5] R. Badia, R. Sirvent, J. Labarta, and J. Perez. Programming the GRID: An Imperative Language Based Approach. Book chapter in Engineering the Grid, Section 4, Chapter 12, Jan. 2006.
- [6] S. Shimizu, R. Rangaswami, and H. Duran-Limon. "Platform-independent Modeling and Prediction of Application Resource Usage Characteristics", Florida International University Technical Report FIU-SCIS-TR-2007-07-05, July 2007.
- [7] L. Fong *et al.* The LA Grid meta-scheduling project. In the poster presentation session of the 5th Latin American Grid (LA Grid) Summit, The IBM T.J. Watson Research Center, NY, U.S.A., September 2007.
- [8] G. Dasgupta *et al.* Fault-tolerant job-flow management in grid environment. In the poster presentation session of the 5th Latin American Grid (LA Grid) Summit, The IBM T.J. Watson Research Center, NY, U.S.A., September 2007.
- [9] S. M. Sadjadi *et al.* The Latin American Grid (LA Grid) Weather Research and Forecast (WRF) Portal. In the poster presentation session of the 5th Latin American Grid Summit, The IBM T.J. Watson Research Center, NY, U.S.A., September 2007.
- [10] Chen, S. S., J. F. Price, W. Zhao, M. A. Donelan, and E. J. Walsh, 2007: The CBLAST-Hurricane Program and the next-generation fully coupled atmosphere-wave-ocean models for hurricane research and prediction. *Bull. Amer. Meteor. Soc.*, 88, 311-317.
- [11] Houze, R. A., S. S. Chen, B. Smull, W.-C. Lee, M. Bell, 2007: Hurricane intensity and eyewall replacement. *Science*, 315, 1235-1239.
- [12] Davis, C., W. Wang, S. S. Chen, Y. Chen, K. Corbosiero, M. DeMaria, J. Dudhia, G. Holland, J. Klemp, J. Michalakes, H. Reeves, R. Rotunno, and Q. Xiao, 2007: Prediction of landfalling hurricanes with the Advanced Hurricane WRF Model. *Mon. Wea. Rev.*, in press.
- [13] T. N. Krishnamurti *et al.* Improved Weather and Seasonal Climate Forecasts from Multimodel Superensemble, *Science* 3, Vol. 285, no. 5433, 1548 – 1550pp, 1999.
- [14] V. Kumar *et al.* Multimodel Superensemble Forecasting of Tropical Cyclones in the Pacific, *Mon Wea Rev.*, 131, 574–583, 2003.
- [15] W. C. Eric *et al.* Real-Time Multimodel Superensemble Forecasts of Atlantic Tropical Systems of 1999, *Mon Wea Rev.*, 131, 1878–1894, 2003.
- [16] Wolfgang Emmerich. Engineering Distributed Objects. John Wiley & Sons, Apr. 2000.
- [17] The Globus Project (Globus Alliance)", <http://www.globus.org>
- [18] Legion Web Site. <http://legion.virginia.edu>
- [19] M. Litzkow, M. Livny, and M. Mutka. "Condor – A Hunter of Idle Workstations", Proc. of the 8th International Conference of Distributed Computing Systems (ICDCS 1988), January 1988, San Jose, CA, IEEE CS Press, USA, 1998.
- [20] A. Anjomshoaa, M. Drescher, et. al. "Job Submission Description Language (JSDL) Specification"; Version 1.0, 2005.
- [21] V. Gucer, J. Biggs-Finstad, et. al. "Getting Started with Tivoli Dynamic Workload Broker 1.1". www.redbooks.ibm.com, SG24-7442-00.
- [22] GridWay Web Site. <http://www.gridway.org>
- [23] I. Rodero, R.M. Badia, J. Corbalan, J. Labarta, "eNANOS Grid Resource Broker", European Grid Conference 2005, LNCS 3470, Amsterdam, Netherlands, 14-16 Feb., 2005, pp. 111-121.
- [24] O. Ezenwoye and S. M. Sadjadi. Composing aggregate Web services in BPEL. In *Proceedings of the 44th ACM Southeast Conference (ACMSE 2006)*, Melbourne, Florida, March 2006.
- [25] O. Ezenwoye and S. M. Sadjadi. TRAP/BPEL: A framework for dynamic adaptation of composite services. In *Proceedings of the International Conference on Web Information Systems and Technologies (WEBIST 2007)*, Barcelona, Spain, March 2007.
- [26] S. M. Sadjadi, P. K. McKinley, B. Cheng, and R. Stirewalt. TRAP/J: Transparent generation of adaptable Java programs. In *Proceedings of the International Symposium on Distributed Objects and Applications*, Agia Napa, Cyprus, October 2004.
- [27] S. M. Sadjadi *et al.* Finding an appropriate profiler for the weather research and forecasting code. Technical Report FIU-SCIS-2007-09-03, School of Computing and Information Sciences, Florida International University, 11200 SW 8th St., Miami, FL 33199, August 2007.
- [28] Sameer S. Shende and Allen D. Malony. "The Tau Parallel Performance System," *International Journal of High Performance Computing Applications*. DOI: 10.1177/1094342006064482 2006; 20; 287.
- [29] IBM HPC Toolkit. https://domino.research.ibm.com/comm/research_projects.nsf/pages/actc.index.html.
- [30] Vincent Pillet, Jesus Labarta, Toni Cortes and Sergi Girona. "PARAVER: A Tool to Visualize and Analyze Parallel Code," CEPBA/UPC Report No. RR-95/03 February 1995.
- [31] O'Reilly, T. "What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software," <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-2.0.html>, 09/30/2005.
- [32] Yahoo UI, <http://developer.yahoo.com/yui/>
- [33] Google Maps, <http://www.google.com/apis/maps/>
- [34] PostgreSQL, <http://www.postgresql.org/>
- [35] Global Forecast System, http://en.wikipedia.org/wiki/Global_Forecast_System
- [36] VisAD, <http://www.ssec.wisc.edu/~billh/visad.html>
- [37] CPUlimit - CPU Usage Limiter for Linux: <http://CPUlimit.sourceforge.net>. Sept. 15 2007.
- [38] S. M. Sadjadi *et al.* Weather research and forecasting model 2.2 documentation: A step-by-step guide of a model run. Technical Report FIU-SCIS-2007-09-02, Florida International University, 11200 SW 8th St., Miami, FL 33199, August 2007.
- [39] R. Badia *et al.* *HPC and Grids in Action*, chapter Innovative Grid Technologies Applied to Bioinformatics and Hurricane Mitigation. IOS Press, Amsterdam, 2007.