Grid implementation of the Weather Research and Forecasting model

Davor Davidović
Centre for informatics and computing
Ruđer Bošković Institute, RBI
Zagreb, Croatia
davor.davidovic@irb.hr

Danijel Belušić, Maja Telišman-Prtenjak
Department of Geophysics
Faculty of Science, University of Zagreb
Zagreb, Croatia
dbelusic@irb.hr, telisman@irb.hr

Abstract— **Atmospheric science is advancing towards very complex phenomena at ever smaller temporal and spatial scales. One of the principal tools utilized in atmospheric science are weather prediction models. These models usually demands large execution times and resource allocation like CPU time and storage space. The main goal of our research is porting of the Weather Research and Forecasting model to the grid infrastructure. Porting has been done through bash scripts that are using existing grid tools for job and data management, authentification mechanisms, and other application level services produced within the SEEGRID project. In this paper, through a few model runs on the Grid we describe certain benefits not only in overall execution time but also in ability of concurrent runs of the same model especially for scientific purposes. During the execution, we have also faced some drawbacks in data bandwidth, unreliability of some grid services and relatively hard control of the model execution flow. The final conclusion is that there is big need and justification for porting the WRF model to the grid, although it takes a lot of effort to be properly implemented.**

I. INTRODUCTION (HEADING 1)

Atmospheric science of today is advancing towards very complex phenomena at ever smaller temporal and spatial scales. These phenomena have large impacts on both scientific research and operational weather forecast. One of the principal tools utilized in these efforts are numerical weather prediction models. These numerical models usually demand large execution times and resource allocations; therefore, we need large parallel systems in order to execute them. One of such numerical prediction models is the WRF-ARW model (Weather Research and Forecasting model – Advanced Research WRF).

The WRF-ARW model is a next-generation numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs. It features a software architecture allowing for computational parallelism and system extensibility. WRF-ARW is suitable for a broad spectrum of applications across scales ranging from meters to thousands of kilometers.

Our research aims at reproduction/forecasting of the airflow over complex terrain (e.g. mountain ranges, coastal area) in Croatia. We are faced with the problem of the lack of computing resources due to the high-resolution model grids, necessary for more accurate and detailed results. Using high model resolution results in two natural consequences: an enlargement of the number of computational grid points, so to keep the total domain size sufficiently large; and reduction of the integration time step in order to avoid numerical instabilities. Both effects result in larger CPU and memory demands and a big storage space that cannot be provided. Even with enough processors, we still do not have the storage to save all the data that the model can generate per run (can be about a hundred GBs).

Till today, not much effort has been done in adopting WRF model to the grid infrastructure. The model has mostly been adopted, with relatively good success, for the execution on local clusters and, with great success, on supercomputers like in NCEP (National Centers for Environmental Predictions) or NCAR (National Center for Atmospheric Research). Due to the lack of financial resources and user access to such large supercomputers we have investigated other options for deploying our model and found that the grid can be a suitable choice. The grid infrastructure has already been utilized for other purposes and is ready to be exploited by numerous different science applications from different user communities.

Within the SEE-GRID-SCI project we have started deploying the WRF-ARW model on the grid infrastructure. We have adapted the WRF-ARW model for the grid usage in a way that hides the grid complexity from the end user as much as possible. Bash scripts that include different gLite and LCG tools are taking care of all the grid processes needed for the

WRF model execution and data flow, such as: storing the data on the grid storage elements (through the LFC file catalog), automatic downloading and executing the appropriate binaries (the model executables stored on different storage elements), retrieving the data after a job has completed, and checking the submitted job status. We have achieved that the model execution time is not only less than before, but even submitting a WRF job on the grid (preprocessing, main application run, post-processing) is fairly easier that on a local PC or a cluster. This is done by developing new execution scripts that take control of local model execution process.

From our previous work and results obtained from starting the WRF-ARW model on the grid, we will show that certain important benefits and improvements from using the grid infrastructure can be acquired. One can see that the main benefit is not only in the speedup of the model execution, but also in the ability to start multi-instances of the same model and also in resolving bigger problems (Gustafson's law). We will also report on the current status of the WRF model in Croatia as well as other partner's countries in this project: Bosnia and Herzegovina, Georgia and Armenia.

II. Grid computing basics

Grid computing is the combination of computer resources from multiple administrative domains applied to a common task, usually to a scientific, technical or business problem that requires a great number of computer processing power or the need to process large amount of data. A typical Grid infrastructure includes: knowledge management resources, data and computing resources, and the appropriate network to accommodate interactions.

The users of a Grid infrastructure are divided into Virtual Organizations (VOs), abstract entities grouping users, institutions and resources in the same administrative domain. A VO typically corresponds to a particular organization or group of people in the real world, like meteo VO for people working in the field of meteorology. Membership of a VO grants privileges to users like ability to use special resources or equipment. In the next paragraph we describe some most important grid resources and services that are used in this paper:

- User interface (UI) is the access point to any grid infrastructure, for example SEEGRID. This can be any machine where users have a personal account and user certificate installed. From a UI, user can be authenticated and authorized to user grid resources, and can access the functionalities offered by the information, workload and data management systems. From UI user can perform grid operations like: submit job for execution, cancel jobs, retrieve the output of finished jobs, show the status of the submitted jobs, copy, replicate and delete file form the Grid etc.

- A Computing Element (CE), in Grid terminology, is some set of computing resources localized at a site (i.e. cluster).

- A Storage Element (SE) provides uniform access to data storage resources. The Storage Element may control simple disk servers, large disk arrays or tape-based mass storage systems.

- The Information Service (IS) provides information about the Grid resources and their status. This information is essential for the operations of the whole Grid.

- The purpose of the Workload Management Systems is to accept user jobs, to assign them to the most appropriate Computing Element, to record their status and retrieve their output.

- Job to be submitted are described using the Job Description Language (JDL), which specifies, for example, which executable to run and its parameters, files to be moved to and from the CE on which the job is run, input Grid files needed, and any requirements on the CE.

III. Grid implementation of the WRF model

A. *Implementation idea*

The primary idea is to develop scripts that hide grid complexity from the end users. The WRF execution on the grid is divided into three different stages: pre-processing, core model execution and post-processing.

The first one incorporates downloading and preparing the model input data, which consist of the terrestrial data and the initial and boundary conditions data. The second one enables running of the model on a grid Computing Element, and the third one deals with managing of the model output data.

The basic scheme is to store all the WRF model data, including binaries and input and output model files, on the grid storage elements, while the execution scripts are installed in the user's home folder on the User Interface node. With the submission process, the execution scripts are sent to the grid CE, the necessary model input files and binaries are downloaded and the model execution is started. After the model execution is finished all the results are stored on the grid storage elements from where users can download them directly to the UI node or local computers using DM-Web application.

B. *Storage structure*

Storage scheme is, together with the model grid operating scripts, the most important part of the model grid implementation, so we will describe it in more details.

Users who want to use the WRF model on the grid must be members of the meteo VO. As a part of the meteo VO, WRF is using tools for data management provided by this VO. The tool for the data management and replica file system is the LFC catalog. The root folder for the meteo VO within the LFC catalog is:

/grid/meteo.see-grid-sci.eu/

In this folder all members of the VO have permission to write, read and modify all files and folders. We have decided to make a root folder for the WRF-ARW model within the meteo VO root folder called WRF-ARW, so that the full path to the model root folder is:

/grid/meteo.see-grid-sci.eu/WRF-ARW/

Hereinafter the "root" will mark the root folder for WRF-ARW in the LFC file catalog.

In the root folder there are three main folders: "bin", "input_data", and "output_data".

```
/WRF-ARW/
    /bin
    /input_data
    /output_data
```

As the project involves a number of different users of the WRF model, the users are divided into regions of interest. The users are usually separated by regional affiliation; therefore we have 6 different regions: Armenia, BiH (Bosnia and Herzegovina), Croatia, Georgia, and SEEurope as the default region for all users.

1) *Bin folder*

In the "bin" folder are all the binaries and auxiliary files for pre-processing ("WPS" folder), the main WRF binaries (the core application) ("WRF" folder), and the post-processing binaries and visualization tools ("postproc" folder). All binaries, depending on the version, are stored in this folder.

```
/bin
    /WPS
        /WPSV<version_number>_<arch>.tar.gz
    /WRF
        /WRFV<version_number>_<arch>.tar.gz
    /postproc
```

2) *Input_data folder*

In the "input_data" folder are the input data for WRF pre-processor (WPS). There are the static terrestrial data that are not changing between different runs, stored in the "terrain" folder, and the time-dependent initial and boundary conditions in the "boundary" folder (which are the input for the ungrib.exe application in the WPS pre-processing system). The initial and boundary conditions are divided between regions of interest because these data are region-specific and to prevent unnecessary mixing of data between different users and regions. The structure of the folder is:

```
/input_data
    /boundary/
            /<REGION>
    /terrain
```

3) *Output_data folder*

The structure of output data folder is:

```
/output_data
    /<REGION>
            /sci
            /oper
```

In the "output_data" folder are all the outputs after the model has finished its execution on the grid. Data are separated depending on the region of interest and depending on whether the model was run for operational or scientific purposes.

C. *Model workflow*

The model workflow is described with "Fig. 1". The main entrance gate to the grid is UI node where are all the scripts that help users to operate the WRF model on the grid are installed.

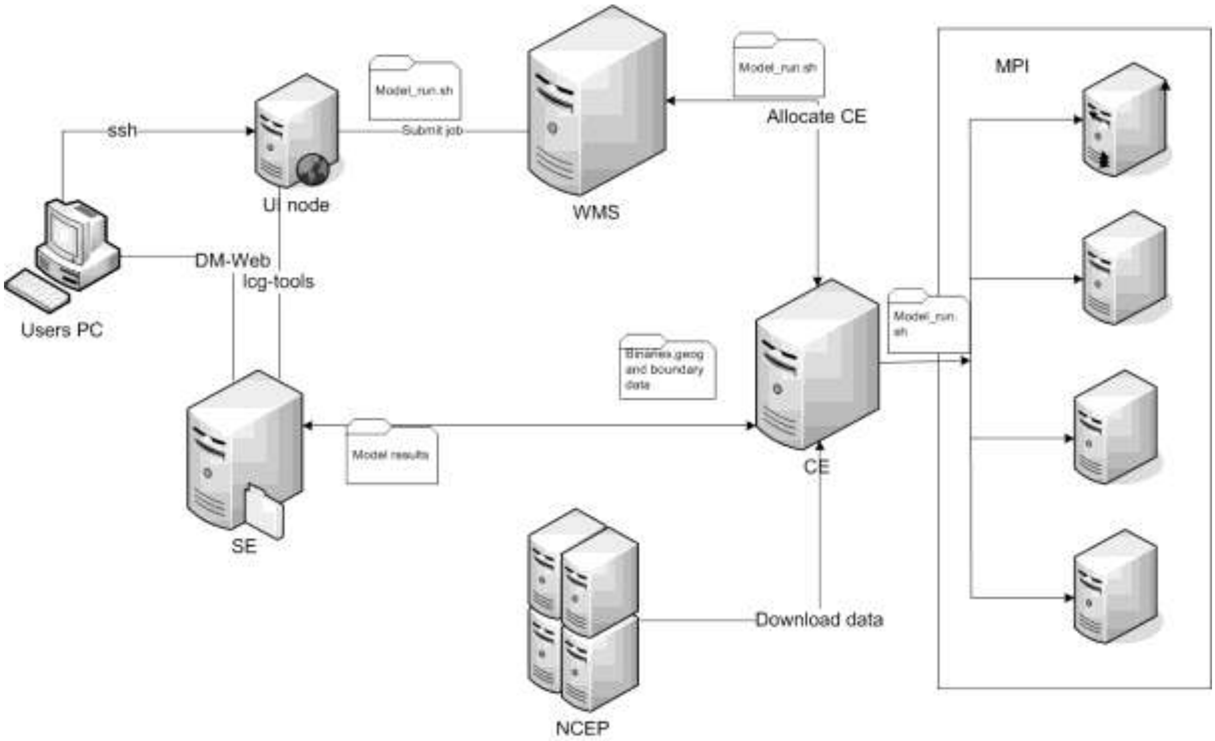

Figure 1. WRF grid workflow

The concept in a nutshell is that the user, working on UI node, submits model to the grid WMS (Workload Management System). WMS allocates the grid CE i.e. grid cluster, and before the model start to execute on the allocated CE, the input terrestrial and initial/boundary conditions data, as well as the WRF binaries, are downloaded on CE from the grid SE (Storage Element). When all input data and binaries are downloaded to CE, the model starts with the execution using the MPI (Message Passing Interface) [5] runtime environment.

The model execution starts with the pre-processing system WPS. WPS consists of the *geogrid* application that prepares the terrestrial data, the *ungrib* applications that extracts the input initial/boundary conditions data, and after these applications are finished, the *metgrid* application horizontally interpolates the intermediate-format meteorological data that are extracted by the *ungrib* program onto the simulation domains defined by the *geogrid* program.

When the pre-processing is finished the core application starts first with the *real* application which sets up vertical model levels for the model input and boundary files and finally the *wrf* application for numerical integration. Only the *wrf* application is run using MPI on multiple CPUs/cores. WRF workflow on a grid CE is illustrated in "Fig 2".

After successful completion, the output data are stored on a pre-defined grid SE. User can download model output data from SE using the LCG-tools, DM-Web application service developed within SEEGRID project or using developed scripts for automatic WRF model's data collecting.

Submission of a job is the process that sets all model parameters, describes the job using JDL (Job Description Language), updates the WRF model namelist files (model descriptions files for domains, physics, etc.) and submit model to the grid using gLite-tools. Regarding the job described, WMS allocates a grid CE where the user's job will be executed. The main script for submitting model on the grid is wrf-submit. The script is responsible for defining and setting almost all parameters needed for the model execution. Based on the user command line parameters the script sets model execution type (scientific or operational), number of processors the model will run on, model input files and some other model specific parameters.

The main model execution scripts on the grid CE is the model_run.sh script. The script is sent from the user UI node together with the job submission. This is the main script on CE responsible for all model operations. The first step is to download and extract model binaries, set file structure, download the static (terrestrial) data from the LFC catalogue and the time-dependent (initial/boundary conditions) data from the LFC catalogue or the NCEP server, depending on the execution mode (will be described in the paragraph "*D. Starting the model*").

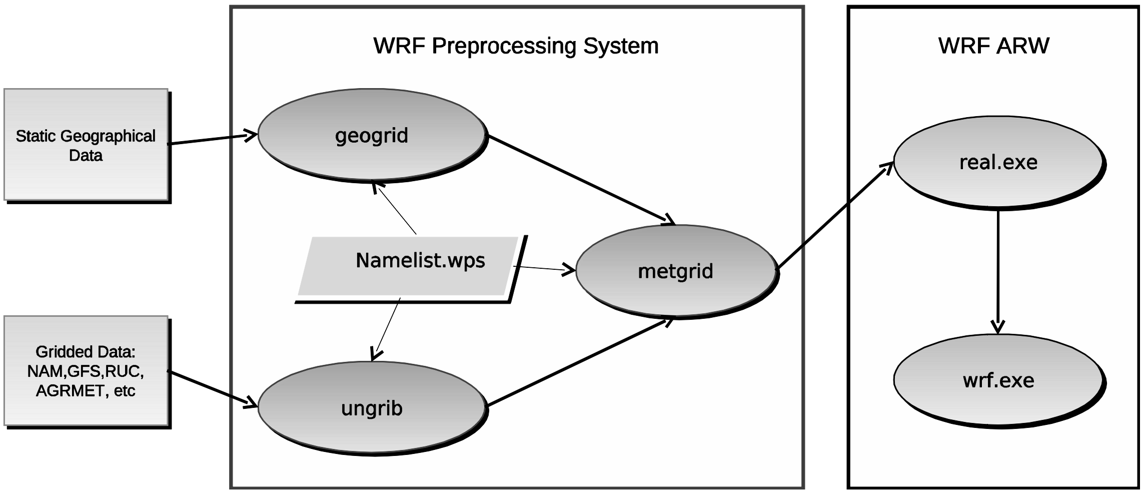

Figure 2. WRF model workflow on a grid CE

D. *Starting the model*

There are two different WRF model modes then users can use: scientific and operational mode. The scientific mode is intended to be used for research purposes, where the users have full control over all settings and relevant model parameters while the operational mode is intended to be used for daily operational forecast with most of the model settings already pre-defined.

1) *Scientific mode*

As said before, the scientific mode is used for the research purposes when the users do not need current boundary conditions data downloaded from NCEP, ECMWF (European Centre for Medium-Range Weather Forecasts) or any other large-scale forecasting center. Users are expected to manually upload and save their initial/boundary conditions data at the LFC catalog in the input data folder for their region. For example, a user from Croatia who wishes to run the model in the scientific mode must save all his initial/boundary conditions data in the folder:

/WRF-ARW/input_data/boundary/CROATIA/raw/,

where the "raw" folder is used for storing the data in the GRIB file format, typical for describing meteorological data.

As in a serial, non-grid execution of the model, user has to describe his model using the WRF files, namelist.wps and namelist.input. The user needs to define all relevant variables in these two files so that the model can execute normally.

The model is executing the by starting *wrf-submit* command. An example of how to the submit model in the scientific mode, with the region of interest "CROATIA" and using 16 CPUs would be:

```
$wrf-submit –m s –r CROATIA –p 16
```

This command automatically checks for the namelist files in the current folder, sets the environment variables, generates JDL job description files and submits the jobs using *glite-wms-job-submit* command from the gLite tools. The output of the *wrf-submit* command is a file that contains user's job ID required to check the job status and to retrieve the output when the job finishes. After the execution has finished, the model output data are stored using the LCG-tools in the LFC file scheme in the folder /WRF-ARW/output_data/<*REGION*>/sci/<*output_filename*>*.tar.gz.*

The user can retrieve the output data on the UI node using the *lcg-cp* tool (which is the standard LFC tool to copy grid files to UI) or the program *wrf-get-data* that collects the output of the finished jobs and download the model output results from the grid CE (LFC) to UI. The other option is to directly download the data from the grid CE to the local computer by using the DM-Web server that can be run from web browser

2) *Operational mode*

The operational mode is designed and adjusted to be as simple as possible and not to require much of user intervention for everyday automatic executing. Users have to adjust all model variables in the namelist files, while the start and end date of the simulation are set automatically (the start date is set to the current date). When executing in the operational mode, the input initial/boundary conditions data are downloaded automatically from the NCEP server on the grid CE and the user's voms proxy certificate is created automatically. An example of starting the operational forecast, using 8 processors, with the simulation time of 48 hours (2 days weather forecast) and "BIH" region would be:

```
$wrf-submit –m o –r BIH –p 8 –l 48
```

The output of the command is, again, a file containing the job ID. In the operational forecast, most of the parameters are set by default, e.g. simulation time period (in the example was it 48 h) or number of processors.

During the operational forecast we have faced problems that many jobs have stayed in the queue of the local batch system on CE for a longer time. As operational forecast has to be finished at certain times of the day we have arranged explicit resource reservation on two grid CE in Bulgaria (BG03-NGCC and BG04-ACAD, part of the SEEGRID grid infrastructure). After the resource reservation, the problem with the execution time was solved.

The users who need to run operational forecast on the earmarked resources have to be members of the group /meteo.see-grid-sci.eu/HR/App/WRF-ARW/ with the roles "*Developer*", because the reservation is made for this group within the meteo VO.

The Cron job scheduler (other job schedulers are also possible) is required on the UI to automatically submit the model on the Grid daily.

The tools for retrieving the model results and the output data are the same as for the scientific mode.

IV. MODEL CURRENT STATUS

The porting of the WRF model on the SEEGRID infrastructure has started in July 2008. At the beginning of the project, only two partner countries were involved: Croatia and BiH. Now 4 partners are participating in the porting, testing and running the WRF model on the Grid. Two new partners are Georgia and Armenia, but also some other countries show their interest in using WRF model, e.g. Moldova.

The model is in production phase for both the scientific and the operational mode. The currently active version installed on the grid is the WRF version 3.0.1.1, released on August 22, 2008. At the moment, we are updating to a newer version 3.1.1, released on July 31, 2009.

In Croatia, some daily forecasts in the operational mode are run, while most of the jobs are in the scientific mode. Most of the jobs using the older version of WRF have finished without error, while some execution problems have occurred with the newest version.

The current status of model usage in BiH is that the users from BiH mostly use the model for weather prediction in the operational mode. Two institutions are using the model for now: the Federal Hydro Meteorological Institute, Sarajevo, and the Republic Hydro Meteorological Institute, Banja Luka. The model is set for their needs to start daily at 4 am, running a 3-day forecast for BiH on 16 CPUs. Also, some manually submitted jobs are run during the day from both institutions, also in the operational mode.

Georgia and Armenia are currently in the phase of acquainting with the full model capabilities and the grid usage, although they have also started, particularly Georgia, with submitting jobs in the operational mode.

V. RESULTS

Here we will presented model results for a few model runs in both the scientific and operational mode. We will show that there are certain problems in running the model on a larger number of CPUs (more that 16 CPUs) because of the communication between processes that is slowed down by the low LAN bandwidth.

In the operational mode we have made test for 72-hours weather forecast from 29th September 2009 to 2nd October 2009 over BiH, with a nested domain over Sarajevo. The model has two domains with the horizontal resolution between grid points of 18 km for the main domain and 6 km for the nested domain, 27 vertical grid levels and the time step of 2 min.

The test was made on the BG03-NGCC. The CE has the Scientific Linux 4.6 operating system with the gLite 3.1 and torque 2.1.9 batch system installed. Single worker node is 2 x Intel Xeon x86_64 processor with 8 cores total, 16384MB of RAM memory and 250GB storage space. Interconnection between worker nodes is 1Gbps network switch.

Tests were run on 2 to 12 CPUs in steps of 2, and from 12 CPUs to 24 in steps of 4. In "Fig 3" is the total time of the model execution. The total time includes the download and extraction of model binaries, download of the terrestrial and initial/boundary conditions data, pre-processing (WPS), model execution (main program) and storing the results on the LFC (grid storage elements).

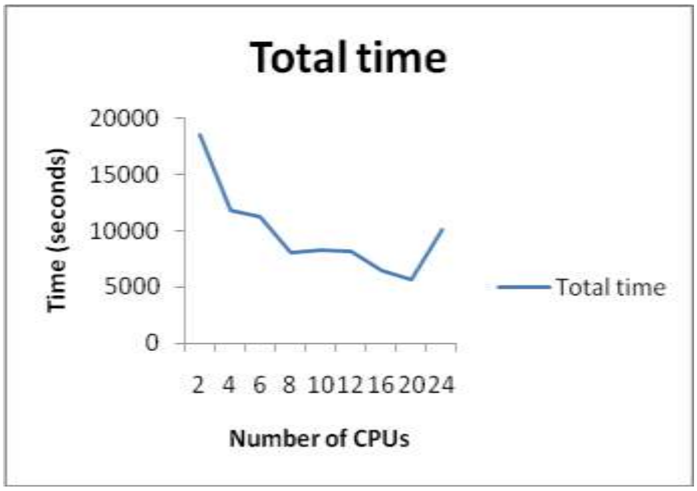

Figure 3. WRF job total time

One can see the fluctuation of the overall execution time. The fluctuation happens because a lot of copying, downloading and extracting of large data sets is made, which can easily interfere with current network occupancy, number of jobs already running, disk usage etc. The overall progress shows that we have gain speedup up to 16 CPUs, but afterward the speedup decreases.

On "Fig 4" we can analyze in more details the execution time of the main application (wrf.exe) which is the most time- and storage-consuming part of the model. We have measured three different times: real, user and system time. Real time is the elapsed real (wall clock) time used by the program, user time is the total number of CPU-seconds that the program used directly, and finally the system time is the total number of CPU-seconds used by the system on behalf of the process.

The most important and relevant time for testing is the real time, because it is the overall time that the application (wrf.exe) spends on the grid CE. As one can see on "Fig 4", the real time is decreasing till 16 CPUs and afterward starts to increase, while the user time continues to decrease. We can see that the system time is more or less static; it is slowly decreasing with the number of the CPUs. The gap between the real and user time is the communication or, better to say, the idle time of the program waiting for the communication to finish. With the larger number of CPUs the pressure on the local NFS and simultaneous write of all processors to the shared file system results in the slowdown of the entire local network and the overall computing time drastically increases.

By analyzing the user time we can see that there is a potential for further speedup, but it mostly depends on the bandwidth and the speed of the local network.

An example of the results in the scientific mode is shown in the following. We have simulated a bora event on the northern Adriatic.

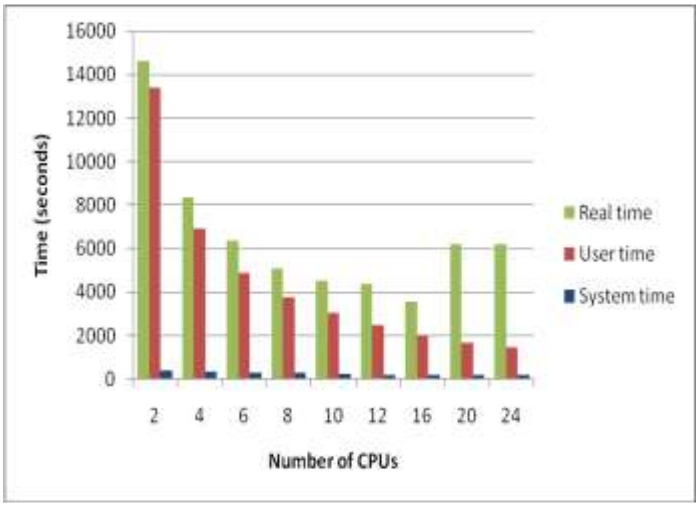

Figure 4. wrf.exe execution time

"Fig 5" shows very well formed bora jets at 10 m above ground on 29 June 2004 at 15 UTC: the primary bora jet through the Vratnik Pass with a width of 25 km, and the second, weaker bora jet through Gornje Jelenje with a width of 15 km. Both narrow jets join together over the Kvarner Bay, forming one broad bora jet about 50 km wide near the surface, several kilometers downstream of the coast. This merged bora jet stretches from the middle of the peninsula to the island of Lošinj, with its centre above the tip of Istria. Except for the bora jets, several simultaneous enlarged areas of bora minima are also seen. They are the western Istrian coast, the sheltered areas in Rijeka Bay (the western sides of Rijeka Bay and the island of Krk) and southern part of the island of Rab. At the western Istrian coast, the sea breeze develops in the narrow area despite the fact that the bora brings cold and dry continental air and suppresses the daytime temperature rise. In Rijeka Bay near Opatija, the low bora speed allows the formation of a weak thermally-induced perturbation.

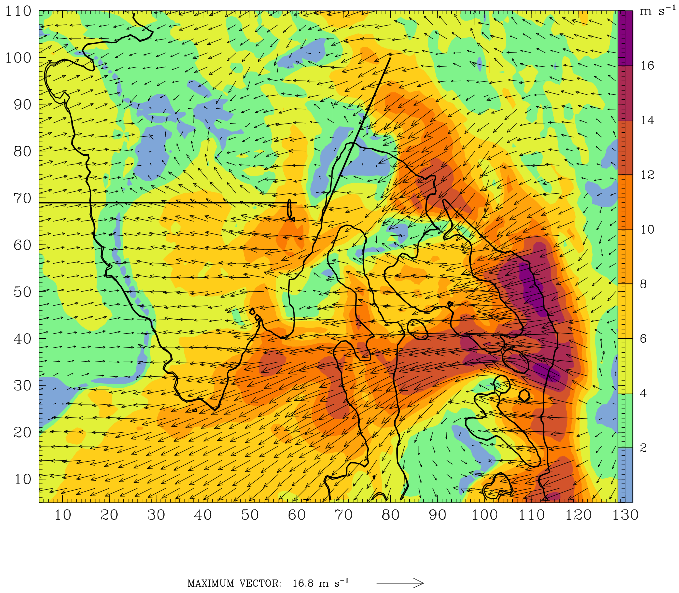

Figure 5. Bora wind simulation - North Adriatic

VI. REFERENCES

[1] W.Wang, C.Bruyere, M. Duda, J. Dudhia, D. Gill, H.-C. Lin, J. Michalakes, S. Rizvi, and X. Zhang, "ARW Version 3 Modeling Systems User's Guide", Mesoscale&Microscale Meteorology Division and National Center for Atmospheric Research,January 2009

[2] S. Burke, S. Campana, E. Lanciotti, P. Mendez Lorenzo, V. Miccio, C. Nater, R. Santinelli, and A. Sciaba, "gLite 3.1 User Guide", Manuals Series, CERN-LCG-GDEIS-722398, April 2009

[3] "SEE-GRID Wiki ", SEE-GRID Wiki – EGEE-see Wiki, [http://wiki.egee-see.org/index.php/SEE-GRID_Wiki]

[4] Belgrade University Computing Centre, "Data Management Web Portal – User Guide", Data Management Web Portal – User Guide – EGEE-see Wiki, [http://wiki.egee-see.org/index.php/Data_Management_Web_Portal_-_User_Guide]

[5] W. Gropp and E. Lusk, “Installation and User’s Guide to MPICH a Portable Implementation of MPI Version 1.2.7”, Argone National Laboratory, University of Chicago,