WRF workflow on the Grid with WRF4G

V. Fernández-Quiruelas[1], **L. Fita**[1], **J. Fernández**[1] **and A. S. Cofiño**[1]
1. Grupo de Meteorología de Santander, Universidad de Cantabria, Spain *

1. Introduction

The Weather Research and Forecasting (WRF) modelling system is composed of several components which need to be executed sequentially. The manual execution of this workflow is a time-consuming and error-prone task: geographical data, simulation dates and other information needs to be copied into different configuration files; the activation of some options requires additional configuration options; etc. Thus, it is customary to automate the process to some degree. However, the automation depends on the experiment to be carried out since the workflow depends on the experiment. This leads to the development of specific WRF workflow automation scripts for each experiment. But this, again, is a time consuming task. When the experiment requires running more than a single model run, the complexity increases and the workflow of the different runs needs to be taken into account. This, again, can be done manually but, as the number of runs increase, the control of the different runs needs to be automated as well. At this point the problems multiply: the large number of simulations now require a monitoring system to check their successful completion; failed runs need to be re-run. If failures are common, the re-running process needs to be automated. This work presents a flexible framework (WRF4G) to manage the WRF workflow covering a wide range of simulation experiments composed of multiple runs with different degrees of dependence. The framework is layered to separate the experiment design from the execution environment. WRF4G includes a monitoring system and easily restarts broken simulations until the experiment is completed.

As the number of simulations of an experiment increase, local computer resources of small institutions may not be enough for running the experiment. Grid technologies emerged in the 90's to agglutinate geographically distributed resources and present them in a uniform and transparent way (Foster and Kesselman 1999). The Grid is specially well-suited for experiments consisting of a large number of simulations. However, Grid environments are subject to intermittent availability and failure rates are higher than in a local resource. Running a numerical model on the Grid is relatively easy, but it is also likely that the simulation fails, does not perform as expected or takes too long for the local queue. Climate models pose specific challenges for the Grid (Fernández-Quiruelas et al. 2010) and, in order to take advantage of Grid technologies, a specific workflow to recover from failures needs to be designed. WRF4G not only ports the WRF model to the Grid but also defines a workflow to manage a large amount of simulations, to recover from failures and to restart broken simulations with minimal data loss. These features are also convenient when running in local resources.

Current Grid infrastructures agglutinate a large amount of computer resources. As an example, the EU-funded project EGEE (Enabling Grids for E-sciencE) connects more than 150000 CPUs and 40 PB of storage.

2. WRF4G framework

The *WRF for Grid* (WRF4G[1]) framework encapsulates the standard WRF workflow allowing the design of experiments composed of a large number of simulations

* *Corresponding author address:* Valvanuz Fernández Quiruelas, Universidad de Cantabria, Avda. de los Castros S/N, 39005 Santander, Spain
E-mail: valvanuz.fernandez@gestion.unican.es

[1] Further and updated information is available at `http://www.meteo.unican.es/software/wrf4g`

with dependencies among them and varying parameters. The design is modular, and the user is faced with separate configuration files to select the execution environment and to define the experiment to be performed using WRF. Data and execution can be physically separated and are hidden behind data, execution and monitoring abstraction layers. This modular design enables, for instance, the separation of the execution from the data. The experiment can run in a machine and the output be saved to a different machine.

a. *Execution and experiment configuration*

The user enters the configuration into 2 files:

i. wrf4g.conf This file contains the resources to be used to run the experiment: the location of the input data, information on the computer system where the experiment will run, the number of processors to use, the location where the output data will be stored, etc.

ii. wrf.input This file contains the details of the experiment to be carried out: the kind of experiment (see the examples below), the input data to be used, the dates, the post-processing to apply before saving the data, etc. Additionally, all `namelist.input` options can be modified from this file. There is no need to provide a namelist.input file, the default will be used and only the modifications to the default file need to be specified.

A WRF4G *experiment* is divided into *realizations*. A realization is a WRF simulation that can be run independently. E.g if the experiment consists of a single continuous run for 20 years, there is only one realization. On the other hand, in an experiment consisting of a multi-physics ensemble, each of the ensemble members is a realization. For convenience, realizations are split into *chunks*, which are, by definition, dependent WRF simulations, each restarted from the previous chunk. Chunks avoid, for instance, generating huge boundary files for long simulations or exceedingly long running times on the queues.

b. *Abstraction layers*

i. Data management layer The data access is done through a virtual copy (vcp) command designed for WRF4G, which encapsulates several transfer protocols transparently. Thus, data locations in the configuration files can be remote and, depending on the protocol specified, they are transfered using cp, rsync or gridftp commands.

The preprocessing of input data to make them available for ungrib is handled by plugin scripts (preprocessor). Other plugin scripts (icbcprocessor) enable the modification of the initial and boundary conditions after real.exe and before wrf.exe is run. These can be used, for instance, to initialize the soil moisture and temperature from another run or, in general, to perturb the initial and/or boundary conditions. Finally, the output files can be postprocessed in the computing nodes before they are transferred to their final location (postprocessor plugin scripts).

Plugin scripts can be built by the user to meet his needs and can be any kind of executable, as long as it gets the expected arguments and creates the files they are expected to. Currently, the plugins are shell scripts using processing tools such as the Climate Data Operators, NetCDF Operators, ECMWF's grib_api or the p_interp post-processing tool to save pressure level data or a subset of the output.

ii. Execution management layer WRF users are used to different environments for running the model. Usually, the commands to be executed are sent as a job to the queue of a local resource manager (SGE, PBS, LoadLeveler, ...). On the grid, there are also resource managers in charge of distributing the load among the available resources. In this sense, the Grid is not much different to other computing systems. WRF4G takes advantage of this fact to transparently run on different resource managers, including the Grid as just another option.

Resource managers are hidden behind *submitters*, which abstract the differences between them. The submitter to be used is selected by the user in the `wrf4g.conf` configuration file. Notice that the whole experiment description is in the `wrf.input` file. Just changing some values in wrf4g.conf the experiment can be run (or continued!) in another computer resource.

Moreover, the model does not need to be installed in the computer resources to be used. Pre-packaged binaries (compiled for the architecture) are retrieved at the beginning of the simulation, the model is run, the out-

put and restart files are saved to their final destination and the running directory is deleted. Logging information is saved along with the output in order to debug possible problems. The pre-packaged binaries to use are part of the resource configuration and are selected in `wrf4g.conf`.

iii. Monitoring layer One of the most useful features of WRF4G is the ability to restart a broken experiment with minimal data loss. Completed chunks are kept and even chunk portions can be kept if sufficient restart frequency was set. Moreover, the restarting process is painless since the experiment is sent for restart exactly as for the first time and the completed chunks are detected and not re-run.

This ability is possible through a monitoring system keeping track of the state of the simulations. The monitor also triggers the postprocessor and uploads output and restart files as soon as they are produced, to minimize the data loss. This key feature is essential on the Grid where the computing resource stability is not guaranteed, but is also very useful on local resources.

A web interface to this monitoring system is under development to allow the user keeping track of his experiments.

3. Benefits of WRF4G

Due to its layered design, WRF4G allows the management of complex WRF simulation experiments consisting of a large number of individual simulations while decoupling the scientific experiment from the technical details of the computer resources to be used.

The plugin philosophy allows for a great variety of experiments. The available plugins serve as examples to develop new ones. Also, additional submitters can easily be created to meet the needs of new particular resources.

WRF4G is the first WRF application designed for high productivity experiments on the Grid and can facilitate WRF users the access to more computing resources. Moreover, Grid infrastructures promote scientific e-collaboration within a virtual team, sharing information, knowledge, applications, data and resources (Bourras et al. 2009). In this sense, WRF4G promotes the scientific collaboration among the WRF community (for instance, there are ongoing experiences within the EU projects EELA and EUAsiaGRID and the Spanish NGI)

4. Examples

WRF4G was designed to serve our high-resolution atmospheric numerical modelling needs in the Santander Meteorology Group (*Universidad de Cantabria*, Spain). The kind of experiments currently covered are driven by our needs, but the design is sufficiently generic so as to cover many possibilities.

As a regional climate modelling tool, WRF4G is currently being used within the ESCENA project to generate regional climate change scenarios for Spain and the CORDEX international initiative, contributing to the generation of regional scenarios for the African continent. Even though these are continuous simulations and the experiments consist of a single realization, the simple management of restarts and the encapsulation of the whole workflow (only new preprocessors for the different forcing GCMs are required) make this framework useful for regional climate simulation. For these long simulations monthly chunks were used. These projects are currently run on local resources, but the plan is to run CORDEX simulations on the Grid

Prior to the CORDEX long term experiment (a 20-year downscaling over Africa nested into the ERA-Interim (Simmons et al. 2007) reanalysis was performed), a sensitivity study to the physical parameterizations was carried out. It consisted of 8 1-year simulations with varying physics options. This sensitivity experiment can be configured as a single WRF4G experiment. In fact, sensitivity experiments can be constructed varying any parameter in the namelist.

The experiments involving a larger number of realizations are better suited for the Grid. As an example, a reforecast for the Mediterranean basin at 15 km resolution was carried out for a project focused on wind and was run partly on the Grid (in the EGEE infrastructure) and partly on local resources. The experiment consisted of hindcasts started every day at 06 UTC for the ERA-Interim 20-year period and running for 42 hours. In this experiment, each day is an independent realization. Hundreds

of these realizations run simultaneously on the Grid.

A sensitivity experiment to the PBL scheme was carried out before the Mediterranean re-forecast for a 3-month test period. In this case, each day and each PBL scheme was an independent realization of a single WRF4G experiment. In this experiment, simulations were short enough to use a single chunk for each realization (i.e. if a realization failed, it was restarted from the beginning).

This framework is also being used to run a daily high-resolution 84h weather forecast over the northern Iberian Peninsula. Two configurations with different physics are run on local resources and an ensemble of another 14 members is sent to the Grid. Other studies such as the sensitivity of case studies to parameterizations or the sensitivity of the WRF model to the number and distribution of vertical levels, proof the versatility of the WRF4G framework.

5. Summary and future work

WRF for Grid (WRF4G) enables the execution and management of WRF experiments encapsulating the model workflow and allowing a flexible design of parametric realizations. It is structured with layers segregating the execution layer from the experiment design. Due to this segregation, WRF4G can be used under a wide range of environments, including Grid infrastructures. WRF4G manages the workflow of the simulations and facilitates the unattended execution of experiments. Thus it is well suited for most uses of WRF, especially those involving a large number of simulations such as re-analysis, regional climate modelling, multi-physics ensembles or sensitivity studies to perturbations of the initial or boundary conditions.

The next steps in the development include the improvement of the metadata management for a more efficient monitoring of the experiments and improve the data management on the Grid with a replica service. It is also planned to develop a web-based monitoring service.

References

Bourras, C., E. Giannaka, and E. Tsiatos, 2009: e-Collaboration Concepts, Systems and applications. *Information Science Reference, E-Collaboration: Concepts, Methodologies, Tools, and applications- Vol1, Section I, Chap. 1.2*, N. Kock, Ed.

Fernández-Quiruelas, V., J. Fernández, A. S. Cofiño, L. Fita, and J. Gutiérrez, 2010: Benefits and requirements of Grid computing for climate applications. An example with the Community Atmospheric Model. *Environ. Modell. Softw.*, **submitted**.

Foster, I. and C. Kesselman, 1999: *The grid. Blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, San Francisco, 677 pp.

Simmons, A. J., S. M. Uppala, D. P. Dee, and S. Kobayashi, 2007: ERA-Interim: New ECMWF reanalysis products from 1989 onwards. *ECMWF Newsletter*, **110**, 25–35.