Geosci. Model Dev. Discuss., 8, 6551–6582, 2015
www.geosci-model-dev-discuss.net/8/6551/2015/
doi:10.5194/gmdd-8-6551-2015

Geoscientific Model Development Discussions
Open Access

WRF4G: WRF experiment management made simple

V. Fernández-Quiruelas[1], J. Fernández[1], A. S. Cofiño[1], C. Blanco[1], M. García-Díez[2], M. Magariño[1], L. Fita[3], and J. M. Gutiérrez[2]

[1]Dpto. Matemática Aplicada y C.C., Universidad de Cantabria, Santander, Spain
[2]Instituto de Física de Cantabria, CSIC-UC, Santander, Spain
[3]Laboratoire de Météorologie Dynamique, CNRS, UPMC-Jussieu, Paris, France

Received: 19 June 2015 – Accepted: 15 July 2015 – Published: 13 August 2015

Correspondence to: V. Fernández-Quiruelas (valvanuz.fernandez@unican.es)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

This work presents a framework, WRF4G, to manage the experiment workflow of the Weather Research and Forecasting (WRF) modelling system. WRF4G provides a flexible design, execution and monitoring for a general class of scientific experiments. It has been designed with the aim of facilitating the management and reproducibility of complex experiments. Furthermore, the concepts behind the design of this framework can be straightforwardly extended to other models.

We describe the user interface and the new concepts required to design parameter-sweep, hindcast and climate simulation experiments.

A number of examples are provided, based on the design used for existing (published) WRF experiments. This software is open-source and publicly available (http://www.meteo.unican.es/software/wrf4g).

1 Introduction

Numerical simulation experiments for weather research and operational purposes are among the most computationally demanding applications due to both the huge amount of data involved and the enormous requirements of computing power. Numerical climate models, in addition, require very long simulation times. The management of a weather or climate numerical simulation experiment is a challenging task. This task is often carried out in an ad hoc manner, which cannot be easily adapted to other types of experiments or even to a different computer system (Redler et al., 2012). This work presents the implementation and use of a framework, WRF4G, to manage the experiment workflow of a particular numerical weather and climate model, the Weather Research and Forecasting (WRF) modelling system. WRF4G provides a flexible design, execution and monitoring for a general class of scientific experiments. It has been designed with the aim of facilitating the management and reproducibility of complex

GMDD
8, 6551–6582, 2015

experiments. The use of parametrized templates for the experiments definition makes WRF4G ideal to perform all kind of parametrized studies.

Furthermore, the concepts behind the design of this framework can be straightforwardly extended to other models (Fernández-Quiruelas et al., 2011).

The WRF modelling system (Skamarock et al., 2008) is one of the most popular mesoscale atmospheric models, widely used by the weather and climate communities. WRF is developed by the National Center for Atmospheric Research (NCAR) in collaboration with several agencies and Universities in the US. Furthermore, WRF is a community model in the public domain and, therefore, benefits from the contributions from an active community (currently, there are more than 25 000 registered users worldwide). Unlike other application-oriented models, WRF provides a flexible and computationally-efficient framework which allows solving a variety of atmosphere simulation problems across different time-scales, from weather forecast to climate change projection. Section 2 provides a brief description of the components of this modelling system, which are essentially shared by other mesoscale atmospheric models.

From a computational point of view, climate models are complex applications formed by several components (including pre- and post-processing modules) which are executed sequentially exchanging information through configuration files and data sets. An example of this "model workflow" is represented in Fig. 1a for the WRF modelling system. The execution of this workflow is a time-consuming and error-prone task, since several configuration files need to be adjusted for the particular simulation to be carried out. Therefore, typically, WRF users develop ad hoc in-house scripts to execute their simulations in their computing infrastructure (workstation, cluster, ...).

Weather and climate simulation experiments often require more than a single simulation. Additionally, computational constraints, such as file size or computing wall-time restrictions, may force single-simulation experiments to be split into dependent pieces. Therefore, the model workflow usually needs to be executed several (sometimes many) times and the simulations involved in the experiment may depend on each other, giving rise to an "experiment workflow". Section 3 provides several common examples

GMDD
8, 6551–6582, 2015

of simulation experiments found in the literature. The management of this experiment workflow is, again, a time consuming task which requires an execution/monitoring system to abstract and automate the process. This kind of system is usually referred to as a workflow execution framework.

There are only few examples of these workflow execution frameworks in the literature. Since the needs are common to all models, the approach given for these frameworks is essentially the same. They provide mechanisms to easily configure, run and monitor different kinds of experiments with a given model, usually on HPC environments. The IC3 autosubmit[1] and WRFPortal[2] allow to execute the EC-Earth and WRF models, respectively.

Other experiment workflows incorporate data pre- and post-processing as well. But, these frameworks are devised to work with specific resources (in most cases the one used in the developers institution) and cannot be exported to others. For instance, the LEAD (now XSEDE) Portal (Christie and Marru, 2007), which runs the WRF, ARPS and DAS models, is a science gateway that uses TeraGrid resources[3]; FRE (Redler et al., 2012) that manages EMS model offers support for NOAA HPC resources; or the ecFlow[4] framework available only on the European Center (ECMWF model).

The efficient management of the simulations of a numerical climate modelling experiment requires a change not only in the application workflow, but also in the user's point of view (Fernández-Quiruelas et al., 2011). The design of different kinds of experiments using the same framework needs a way to describe the experiment in terms of a few parameters. The dependencies among simulations introduce new concepts, which are described in Sect. 4. These concepts guide the configuration of the experiment, which is implemented in two separate configuration files: one for the computer resources and

[1] http://ic3.cat/wikicfu/index.php/Tools/Autosubmit

[2] http://esrl.noaa.gov/gsd/wrfportal/

[3] https://www.xsede.org

[4] https://software.ecmwf.int/wiki/display/ECFLOW/Home

GMDD
8, 6551–6582, 2015

the other for the scientific experiment to be carried out. Some configuration examples are provided in Sect. 6, covering those experiment workflows introduced in Sect. 3.

As the number of simulations increase, users may need to use additional computer resources and there is an additional effort in configuring the model scripts to execute in the new infrastructures. Furthermore, if the user wants to use several computer infrastructures at the same time, new services are needed to schedule and dispatch the jobs according to the status of the resources. Section 5 shows how new computing and data resources can be easily added to WRF4G. In a recent work, Fernández-Quiruelas et al. (2011) describe the challenges and limitations posed by climate application workflows on distributed computing infrastructures. Fernandez-Quiruelas et al. (2015) provide specific details of the implementation of WRF4G. Therefore, in this work, we focus on the user interface and the new concepts required to design different experiments in a flexible way. The reader interested in a detailed description of the computational implementation, including the ability of WRF4G to distribute the load across distributed computer infrastructures, is referred to Fernandez-Quiruelas et al. (2015). WRF4G is an open-source and publicly available software that can be downloaded from the Santander Meteorology Group website[5].

2 WRF model workflow

The WRF model workflow (Fig. 1a) consists of a sequence of steps which take initial and boundary conditions from coarse global models and high-resolution static data (orography, land use, ...) and produce high-resolution meteorological fields consistent with these forcings.

The sequence is as follows. Two programs need to be run initially, in any order. Using `geogrid`, the user defines the model spatial domain, choosing a horizontal resolution and a grid projection, size and location. This program interpolates high-resolution static

[5] http://www.meteo.unican.es/software/wrf4g

GMDD
8, 6551–6582, 2015

data to this domain and stores theinformation in a NetCDF-formatted file (*geo_em*). Using `ungrib`, the coarse global meteorological data in Grib format are converted to a binary WRF intermediate format. This step can be replaced by a home-made program as long as it produces the intermediate format files. The `ungrib` program is just an example to get these files out of GRIB-formatted files.

Intermediate files are still in the original global meteorological data resolution. The next step requires both the intermediate and *geo_em* files. Using `metgrid`, the information of these two kinds of files is merged and the coarse global meteorological data are interpolated to the model domain.

WRF discretizes the vertical dimension into hybrid terrain-following eta levels. The interpolation in the vertical from the native levels of the meteorological data to the eta levels is performed by a program called `real`, which produces the initial and boundary conditions in the format (NetCDF with WRF metadata) required by the numerical core of the model. This numerical core (`wrf`) is the last step in the model workflow and solves the dynamics and physics of the atmosphere, producing high-resolution meteorological data.

The programs `geogrid`, `ungrib` and `metgrid` are known as the WRF Preprocessing System (WPS) and are configured using a common file (*namelist.wps*). Likewise, the `real` and `wrf` programs are configured using a different, but common, file (*namelist.input*).

This model workflow does not only apply to WRF, but also to many other mesoscale or regional climate models. For instance, in MM5 (Grell et al., 1995) the steps are `terrain` (similar to `geogrid`), `regrid` (`ungrib`), `interpf` (`metgrid+real`) and `mm5` (`wrf`). In RegCM4 (Giorgi et al., 2012), there is also the equivalence with the programs: `terrain` (`geogrid`), `icbc` (`metgrid+real`), and `regcm` (`wrf`).

GMDD
8, 6551–6582, 2015

3 Overview of WRF experiment workflows

The simplest atmospheric modeling experiment consists of a single execution of the model workflow (Sect. 2). However, current research studies or operational procedures are seldom based on a single simulation. Instead, several simulations with different dependencies among them are run. This Section illustrates three general classes of experiments which are commonly carried out with WRF and other models.

3.1 Parametric

In a general sense, a parametric experiment consists of a set of simulations where a given configuration parameter is changed. In this kind of experiments the sensitivity of the model output to the variation of the parameter is studied for different purposes: process understanding, model improvement, uncertainty assessment, etc.

A particular example are multi-physics ensembles, where the same dynamical core is run with different physical parameterizations. These parameterizations handle the effects of important small-scale phenomena on the model variables. Processes such as radiative heating/cooling, water condensation, turbulence or soil–atmosphere exchanges are handled by physical parameterizations, which are not unique. The performance of different parameterizations for the same physical process depends on many variables: geographical location, season, meteorological variable of interest, or even the statistic of interest, e.g. mean, variability, extremes, etc. (Fernández et al., 2007; Jerez et al., 2012). Therefore, a set of simulations with different schemes, a multi-physics ensemble, is common before applying the model to a particular region.

There are many recent examples of WRF multi-physics ensembles in the literature (Awan et al., 2011; Evans et al., 2011; Mooney et al., 2013; García-Díez et al., 2012). Physical parameterization changes are not the only parametric experiments with WRF in the literature. Other options such as changes in the number of vertical levels, the model top of the atmosphere, damping options, etc. are also commonly used (Awan et al., 2011).

GMDD
8, 6551–6582, 2015

3.2 Hindcast

Short to medium range weather forecasts are routinely produced at different institutions. A few of them produce global forecasts and many others use these global forecasts as boundary conditions for higher-resolution regional forecasts. In order to assess forecast errors, a long sample of previous forecasts with the same system is required. For these purpose, it is common to produce an ensemble of past historical forecasts, which are referred to as hindcasts. The same methodology is applied in seasonal forecasting (Díez et al., 2011).

Hindcasts are also a mean to produce past reconstructions of the atmospheric state. Jiménez and Dudhia (2012); García-Díez et al. (2012); Menéndez et al. (2014) are recent examples of WRF experiments run for this purpose. The frequent initialization of the model using observed data prevents it to drift away from the observed reality and constrains the internal variability.

In any case, this type of experiment requires the initialization of the model at regular time intervals and it usually involves many simulations. For example, a 20 year hindcast of daily simulations requires the management of more than 7000 simulations.

3.3 Climate simulation

Climate simulation requires long records to infer the average behaviour of the atmosphere. The models used are essentially the same as for short term forecasting, but they are usually coupled to other models dealing with slower components of the climate system (land surface, oceans, etc.). In the case of limited area models, most of these slow components are usually prescribed through the boundary conditions and, therefore, regional climate simulations consist of long, continuous simulations using the same model as in the previous experiment types.

There are also many recent studies using WRF to produce regional climate simulations over different parts of the world (Nikulin et al., 2012; Mooney et al., 2013; Argüeso et al., 2012a, b; Cardoso et al., 2012; Jiménez-Guerrero et al., 2013; Vautard et al.,

GMDD
8, 6551–6582, 2015

2013). Most of them deal with the model ability to represent the observed climate and to project future changes at regional scale, in response to increased concentrations of greenhouse gases.

Regional climate simulations can use a computer resource during months to produce several decades of simulated atmospheric states. This results in huge output data sets. Furthermore, since regional models require not only initial, but also boundary conditions along the whole simulated period, the input files for this type of experiment are also very large. Since a computer resource can hardly be available uninterruptedly for several months, WRF (and all models) has the possibility to be restarted from special files which are created according to the user needs. Therefore, the climate simulation can be divided into parts but, unlike the previous experiment type, these parts are dependent on the previous one, which will produce the restart file required to run the next.

4 Understanding WRF4G: new concepts

The execution of a single simulation with WRF is very vulnerable to a large amount of errors and problems, such as missing or corrupt input data, errors editing the *namelist* files, etc. These problems can make the model crash in any step of the workflow or, even worse, let the model work with an improper configuration or input data that might affect the scientific results. In the case of ensembles or multi-parameter experiments, if the simulations are handled separately, it is easy to find differences in the configuration of the members, what leads to errors in the output.

As the number of simulations of an experiment increases, users may need to use more computer resources. The effort of configuring new infrastructures to run WRF experiments is very high. It involves developing scripts that fit the new infrastructure characteristics (batch system, parallel environment, data repository, ...). Furthermore, if the user wants to use several computing infrastructures at the same time, new challenges have to be faced (Fernandez-Quiruelas et al., 2015).

GMDD
8, 6551–6582, 2015

WRF4G provides an easy handling and monitoring of WRF experiments and hides the complexity of leveraging several computing resources at the same time. To understand how WRF4G works and how to run experiments with it, it is necessary to get familiar with the concepts it has been developed on.

4.1 Experiment components

In WRF4G, a set of individual simulations which try to answer one or some specific scientific questions are called "experiment". We call each of this individual simulations a "realization". From the point of view of an automatic workflow, most of the interesting experiments can be defined by a parametrized *namelist* file. Thus, *experiment.wrf4g* contains the parameter ranges of the individual *namelist* files corresponding to the different realizations forming the experiment.

WRF is able to produce restarts, which are large NetCDF files containing all the data that WRF needs to start running from a given model timestep as it was during the simulation (as a kind of memory dump). When a simulation stops, WRF can continue the simulation from the last restart file, so there is no need to start from the beginning. Even the WPS and `real` steps are skipped when starting from a restart file. This capability is essential to perform long, continuous simulations, with weeks of walltime, and enables the use of WRF as a Regional Climate Model.

When a realization covers a very long period of time, which is the case of climate change projections, dealing with restarts is not enough. The boundary files with decades of data are huge and may cause storage space problems. Also, some file systems might have problems handling very large files. To solve this, the concept of "chunk" is defined in WRF4G. Realizations can be divided into chunks, which are consecutive, dependent WRF simulations, which read the last restart file generated by the previous simulation (chunk) in the chain.

GMDD
8, 6551–6582, 2015

4.2 WRF execution workflow

When a single computing resource is used to run an experiment, data is usually stored locally and data transfer time is negligible compared to the overall job execution time. When more than a single computing resource is used, data has to be accessible from all computing resources. If data repositories and computing resources are not connected through a fast network, data transfers become a bottleneck. To minimize data transfers, WRF4G integrates the postprocessing step (WRF output is filtered to reduce its size) after the simulation execution in the computing node. For the same reason, the `geogrid` program is not run inside the WRF workflow of WRF4G (see Fig. 1b). The output generated by `geogrid` can be recycled in different model runs, as the geographical data are independent from the driving model, time range covered and model configuration (except the resolution). These data must be produced by the user, and provided to WRF4G to specify the model spatial domain.

A preprocessor step has also been included in the WRF4G workflow to convert the input data used as initial and boundary conditions to a format readable by `ungrib`.

The WRF workflow execution in the computing node is driven by a `wrapper` script in charge of preparing the environment and orchestrating the run. The `wrapper` contains a `monitor` that tracks the events occurred during the model execution and updates them in a central database. This `wrapper` also uploads the restart and post-processed output files to the data respositories as their are being produced.

4.3 Computing and storage resources

Today, most researchers have access to several clusters and grid infrastructures and can rent on-demand cloud resources to temporarily solve peak workloads. The distributed nature of these infrastructures complicates tasks such as the monitoring and debugging of applications. And the interoperability among the computing resources is also an issue.

GMDD
8, 6551–6582, 2015

WRF4G hides this complexity and facilitates the use of several computing infrastructures at the same time. Resource management is performed transparently and in a centralized manner from the computer where WRF4G is installed (see Fig. 2). This computer will be referred to as WRF4G User Interface (WUI) in the following. Users can configure the WRF4G computing resources (WCR) where they want to run their experiments by listing them in a WUI configuration file. Each resource is defined by the protocol used to access it (*ssh*, *gsissh* or *local*) and by its resource manager (*none*, *sge*, *pbs*, *lsf*, *slurm*, *loadleveler*, *globus* or *cream*). Section 5.2.2 describes in detail how computing resources are configured in WRF4G.

The experiment chunks can be run in any of the configured resources or just in a subset of them. Filters and scheduling policies based on the resources characteristics (operating system, CPU speed, disk quotas, etc.) can be established. The only software requirement to run WRF4G in a WCR is the Python programming language, which is commonly available. Nothing else needs to be installed or configured.

The location of the executables and libraries required to run WRF can be customized in the experiment configuration file for each computing resource. The serial, MPI, OpenMP and Hybrid (MPI-OpenMP) execution environments of WRF are supported. If the location of the executables and libraries required to run WRF is not supplied, a precompiled binary is transferred to the computing nodes during the environment preparation. The precompiled binary comes along with OpenMPI.

The domains, input and boundary data and the precompiled WRF binaries can be stored in different storage resources and accessed through different protocols. These resources will be referred to as WRF4G Data Resources (WDR). Section 5.2.1 shows how the data management is performed with WRF4G.

5 WRF4G framework

WRF4G is a tool that simplifies the execution of atmospheric numerical simulation experiments with WRF. It provides full control of the configuration of the simulations and

GMDD
8, 6551–6582, 2015

means for restarting part or the whole experiment in case of failure. It also provides the ability of reproduce the experiment fully or partially.

WRF4G is an open-source and publicly available software that can be downloaded from the Santander Meteorology Group website[6] and deployed in any supported Linux system. Again, the only requirement for the WUI is Python. In order to simplify the learning process, WRF4G provides some examples ready to run in the WUI. In the default WRF4G configuration, the WUI also acts as a WCR and as a WDR. A small repository with the input and boundary datasets needed to run the examples together with a pre-compiled MPI version of WRF is included in the installation. Thus, users can get familiar with WRF4G by running small experiments in their PC.

WRF4G runs a database that persists all the experiment information and status (namelist files, toolchain used to create the executables, preprocessor, ...), thus facilitating the reproducibility of the results. Although, the database used by default is MySQL installed with the WRF4G, other instances of MySQL or any other relational database could be used. With the information stored in the database, the status of the experiment execution can be monitored in real-time.

WRF4G is layered to separate the experiment design from the execution environment. An atmospheric simulation experiment is defined through two configuration files: one contains the scientific configuration of the experiment (start and end dates, model configuration, experiment setup, input data, postprocessing to apply, etc.) and the other, the execution environment (number of MPI process to run, memory required, data repositories, etc.). Below we describe how to configure different kinds of experiments with WRF4G. More information about the WRF4G usage and components can be found in the WRF4G website.

[6] http://www.meteo.unican.es/software/wrf4g

GMDD
8, 6551–6582, 2015

5.1 Experiments definition

In WRF4G, the characteristics of an experiment are defined in the file *experiment.wrf4g*. The main fields to be defined are:

- `experiment_name` and `experiment_description`.
- `domain_name` is the folder containing the information of the domain (*namelist.wps* and geo_em files).
- `extdata_vtable`, `extdata_preprocessor`, `extdata_path`, `extdata_interval`: These variables describe the external data to feed the model (variable table to decode the input, the preprocessor to make it readable for ungrib, the location and the time interval between input records).
- `postprocessor` is the label identifying the postprocessor to be used to filter and reformat the model output.
- `start_date` and `end_date` define the time span of the experiment.
- `chunk_size_h` is the length of the chunks in hours.
- `multiple_dates` activates the hindcast mode. Many realizations are setup with different starting dates. These starting dates must be into the interval defined by the previous variables star_date and end_date. Another two variables (`simulation_interval_h` and `simulation_length_h`) define the interval between the starting dates and the length of each realization respectively (see examples is Sect. 6).
- `multiple_parameters` activates (if set to 1) parametric experiments. The realizations defined by the previous settings will be run with different parameters. The parameters to vary are defined using `multiparams_variables`, and they can be any parameter of the original *namelist.input* file. The different parameter

GMDD
8, 6551–6582, 2015

combinations are defined with the `multiparams_combinations` variable, and they can be labeled using `multiparams_labels`.

Apart from these specifications, any parameter of the *namelist* can be changed from its default value in a simple way. Section 6 shows a few examples of how the experiment types described in Sect. 3 would be configured.

5.2 Resources definition

In the default configuration of WRF4G, the only computer resource available to run experiments is the computer where WRF4G is installed. This computer is also used as storage resource. All the example experiments use the "/repository" folder, under the default installation location, as storage repository.

Below, we describe how other computing and data resources can be added.

5.2.1 Storage resources

The execution of each experiment requires several files: The input and boundary condition data, the domains, the preprocessor and postprocessor, etc. WRF4G distinguishes four kinds of data sources:

- WRF4G_APPS: In this location can be stored all the applications required to run the WRF workflow in the computing resources. The data repository installed by default contains some pre- and post-processors, the NetCDF library and a pre-compiled MPI version of WRF.
- WRF4G_DOMAINPATH: Location where the domains geo_em NetCDF files for this experiment are located. As shown in 4.2, before running a experiment, users have to run `geogrid` manually and upload the resulting files to this repository.
- WRF4G_INPUT: It can be used in *experiment.wrf4g* to set the path to access the input data from a global model.

GMDD
8, 6551–6582, 2015

– WRF4G_BASEPATH: Location where the files generated by the experiment are saved. These include output files along with log, restart and WPS files.

This information is configured for each experiment in the *resources.wrf4g* file. Each of these data sources can be located in a different server and exposed through different protocols (see Fig. 2). To date, the following URL protocols are supported: file, rsync, sftp, http and gridftp (Allcock et al., 2005). WRF4G performs the data transfers between the data repositories and the computing resources in a way that is transparent to the users. Note that local copies (file) should be only considered when the simulations are run in the same node where the repository resides.

5.2.2 Computing resources

WRF4G is not designed with the purpose of submitting jobs to a specific type of WCR. Instead, researchers can define a wide range of resources. WRF4G is a general-purpose framework that manages different types of CRs. Thus, an experiment may be performed on laptops, desktops, workstations, clusters, supercomputers, clouds and grids; being all of them configured in a file called *framework4g.conf*. This configuration resource file consists of sections, each led by a [section] header, followed by key=value entries. This is illustrated in Fig. 3. This example shows the *framework4g.conf* content of a WUI installed on a workstation. Specifically, it defines three CRs: *my_wokstation* where the WUI has been installed, and two clusters accessed through SSH protocol (*pbs_cluster* and *slrum_cluster*). CRs in WRF4G are characterized by the protocol used to access resources (*communicator*) and the resource manager used to handle jobs (*lrms*). These keys are always common for each resource. Note that in case of an *ssh* communicator, a *username* and a *frontend* for the connection must be provided. Usually, this information is not enough to define a WCR. Thus, a resource can contain other keys, such as *queue*, *max_jobs_running* and *max_jobs_in_queue*. The combination of them is used by the job scheduler to sort the resources. Several other attributes

GMDD
8, 6551–6582, 2015

are allowed, which are not present in this example. More information about them can be found on the WRF4G wiki[7].

In order to list CRs, their features (e.g. operating system, architecture, queues, etc.) and their status (number of running jobs, number of jobs in queue, etc.), WRF4G provides a command named wrf4g_resources (a sample output from this command is shown in Fig. 4).

Additionally, when an experiment is submitted, a scheduler selects the best-fitted resource according to a default scheduling policy. This default scheduling policy can be modified for each experiment by configuring the variables *REQUIREMENTS* and *RANK* in the *resources.wrf4g* file. Both are mathematical expressions that are evaluated for each resource. *REQUIREMENTS* evaluates which resources are considered to submit an experiment, whereas *RANK* sorts the candidates filtered by the REQUIREMENTS. Figure 5 shows an example of both expressions.

Finally, researchers can also define the number of MPI processes used to run WRF(*NP*). If *NP* is set to 1, the serial version of WRF will be run, otherwise, MPI will be used. The default WRF executables provided by WRF4G have been compiled against openMPI libraries.

5.3 Managing experiments with WRF4G

Once the configuration files defining the experiments have been created (*experiment.wrf4g* and *resources.wrf4g*), the user is ready to prepare the WRF4G environment to run the experiment using the wrf4g_prepare command. In this phase, the *experiment.wrf4g* configuration is analysed and the details about the resulting realizations and chunks are recorded in the database. WRF configuration files for each of the chunks are also created and transfered to the data repositories pointed by the "WRF4G_BASEPATH" variable in the *resources.wrf4g* file.

[7] https://meteo.unican.es/trac/wiki/WRF4G

GMDD
8, 6551–6582, 2015

Then, users can submit the whole experiment (by default) or just some realizations by running the `wrf4g_submit` command. In this phase, the database is queried to obtain a list of the chunks that need to be run. These chunks are efficiently scheduled to the computing resources taking into consideration the dependencies among them. The experiment (or just some realizations) can be killed at any moment using `wrf4g_kill` and its status can be monitored with `wrf4g_status`.

Figure 6 shows the output of `wrf4g_status` when the second Chunk of a experiment with a single realization is downloading the boundary conditions.

The granularity of the wrf4g commands allows to handle separately realizations and chunks. In this way, if, for example, one output file generated by a chunk is corrupt, we can rerun just this chunk again. This can be done because the restart file associated to the previous chunk is stored in the repository.

6 Use cases

6.1 Parametric

As previously explained, a parametric experiment is a set of simulations identical except for a parameter, or a parameterization, that is changed inside the model. García-Díez et al. (2014) is an example of this kind of experiment. These authors simulated the Euro-CORDEX domain with 6 different combinations of parameterizations, and compared the results with observations. The novelty of the study is that, apart from temperature and precipitation, they evaluated variables seldom studied as radiative fluxes and soil moisture. As shown in Fig. 7, the ensemble was easily designed by using the variable "multiparams_combinations" to choose the values of the namelist variables defined in "multiparams_variables". In this case, cumulus, radiation and microphysics parameterizations are varied, but any parameter in the WRF namelist can be changed.

GMDD
8, 6551–6582, 2015

6.2 Hindcast

In a hindcast experiment, the regional model is restarted frequently from the global model, running retrospective forecasts or re-forecasts. Menéndez et al. (2014) compared this running scheme with other techniques, known as spectral nudging and grid nudging. They found that the re-forecast scheme was the more accurate, apart from being the computationally cheapest. Thus, this scheme was used to produce a high resolution regional reanalysis over the Mediterranean sea, covering 20 years. This experiment was configured in WRF4G following Fig. 8. After setting the variable "multiple_dates" to one, "simulation_interval_h = 24" and "simulation_lenth = 42" tell WRF4G to start the model daily, and to run it for 42 h. Thus, in this case, we configured WRF4G to run 7305 individual simulations. These are independent and can be run in parallel.

6.3 Climate simulation

Finally, an example of a climate simulation, a continuous simulation with no restarts or multi-parameter settings, is the one that was used in Nikulin et al. (2012) (labeled as UC-WRF31). These authors evaluated the precipitation climatology of a regional multi-model ensemble over Africa for the first time. In Fig. 9 it is shown how WRF4G was configured in this case. As "multiple_dates" is set to zero, WRF4G does not restart WRF from the global model in any case, and simply runs continuously for twenty years. In this case "chunk_size_h" is set to 14 days (in hours). This means that the time interval is divided into 522 individual simulations, each one starting from the last restart file written by the previous one. Thus, in contrast with the hindcast, these are dependent and must be run sequentially.

GMDD
8, 6551–6582, 2015

7 Conclusions

In this paper we present a public domain WRF workflow execution framework (WRF4G[8]) that automatically manages the above mentioned problems by means of execution, monitoring and data management frameworks which provides transparent access to heterogeneous resources (local clusters, grid, clouds, HPC, etc.). Thus, the definition and execution of typical experiments commonly undertaken with WRF is highly simplified. This leads to an enormous save of time and facilitates the access to new infrastructures with no additional overhead. The WRF4G framework is based on layers to separate the experiment design from the execution environment, including a monitoring and management system to easily restart broken simulations until the experiment is completed, and the ability of running these experiments on heterogeneous distributed computing resources concurrently in a transparent way. Thus, the WRF4G capability to harness powerful computing resources such as grids and supercomputers, allow the WRF community to undertake more ambitious problems with higher impact in society.

Code availability

WRF4G is an open-source and publicly available software continually under development. Version control is used when updating the code. The version described in this article corresponds to revision number 1875, which can be downloaded at http://www.meteo.unican.es/software/wrf4g, along with the most recent version.

Author contributions. V. Fernández-Quiruelas, J. Fernández, A. S. Cofiño, C. Blanco, M. García-Díez and L. Fita developed different components of the software presented in this article. M. García-Díez, M. Magariño. and L. Fita tested the software and provided the simulation examples. V. Fernández-Quiruelas, M. García-Díez, J. Fernández and J. M. Gutiérrez prepared the manuscript with contributions from all co-authors.

[8] Available at http://www.meteo.unican.es/software/wrf4g, fully documented.

GMDD
8, 6551–6582, 2015

Acknowledgements. This work has been supported by the Spanish National R&D Plan under projects WRF4G (CGL2011-28864, co-funded by the European Regional Development Fund –ERDF–) and CORWES (CGL2010-22158-C02-01) and the IS-ENES2 project from the 7FP of the European Commission (grant agreement no. 312979). C. Blanco acknowledges financial support from programa de Personal Investigador en Formación Predoctoral from Universidad de Cantabria, co-funded by the regional government of Cantabria. The authors are thankful to the developers of third party software (e.g. GridWay, WRFV3, python and NetCDF), which was intensively used in this work.

References

Allcock, W., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I., and Foster, I.: The Globus Striped GridFTP Framework and Server, in: Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, IEEE Computer Society, p. 54, 2005. 6566

Argüeso, D., Hidalgo-Muñoz, J., Gámiz-Fortis, S., Esteban-Parra, M., and Castro-Díez, Y.: Evaluation of WRF mean and extreme precipitation over Spain: present climate (1970–99), J. Climate, 25, 4883–4897, 2012a. 6558

Argüeso, D., Hidalgo-Muñoz, J., Gámiz-Fortis, S., Esteban-Parra, M., and Castro-Díez, Y.: High-resolution projections of mean and extreme precipitation over Spain using the WRF model (2070–2099 versus 1970–1999), J. Geophys. Res., 117, D12108, doi:10.1029/2011JD017399, 2012b. 6558

Awan, N., Truhetz, H., and Gobiet, A.: Parameterization induced error-characteristics of MM5 and WRF operated in climate mode over the Alpine Region: an ensemble based analysis, J. Climate, 24, 3107–3123, doi:10.1175/2011JCLI3674.1, 2011. 6557

Cardoso, R., Soares, P., Miranda, P., and Belo-Pereira, M.: WRF high resolution simulation of Iberian mean and extreme precipitation climate, Int. J. Climatol., 33, 2591–2608, 2012. 6558

Christie, M. and Marru, S.: The LEAD Portal: a TeraGrid gateway and application service architecture, Concurr. Comp.-Pract. E., 19, 767–781, doi:10.1002/cpe.1084, 2007. 6554

Díez, E., Orfila, B., Frías, M., Fernández, J., Cofiño, A., and Gutiérrez, J.: Downscaling ECMWF seasonal precipitation forecasts in Europe using the RCA model, Tellus A, 63, 757–762, 2011. 6558

GMDD

8, 6551–6582, 2015

Evans, J., Ekström, M., and Ji, F.: Evaluating the performance of a WRF physics ensemble over South-East Australia, Clim. Dynam., 39, 1241–1258, doi:10.1007/s00382-011-1244-5, 2011. 6557

Fernández, J., Montávez, J. P., Sáenz, J., González-Rouco, J. F., and Zorita, E.: Sensitivity of MM5 mesoscale model to physical parameterizations for regional climate studies: annual cycle, J. Geophys. Res.-Atmos., 112, D04101, doi:10.1029/2005JD006649, 2007. 6557

Fernández-Quiruelas, V., Fernández, J., Cofiño, A., Fita, L., and Gutiérrez, J.: Benefits and requirements of grid computing for climate applications. An example with the community atmospheric model, Environ. Model. Softw., 26, 1057–1069, doi:10.1016/j.envsoft.2011.03.006, 2011. 6553, 6554, 6555

Fernandez-Quiruelas, V., Blanco, C., Cofino, A., and Fernández, J.: Large-scale climate simulations harnessing clusters, grid and cloud infrastructures, Future Generation Computer Systems, 51, 36–44, 2015. 6555, 6559

García-Díez, M., Fernández, J., Fita, L., and Yagüe, C.: Seasonal dependence of WRF model biases and sensitivity to PBL schemes over Europe, Q. J. Roy. Meteor. Soc., 139, 501–514, 2012. 6557, 6558

García-Díez, M., Fernández, J., and Vautard, R.: An RCM multi-physics ensemble over Europe: multi-variable evaluation to avoid error compensation, Clim. Dynam., 1–16, 2015. 6568, 6580

Giorgi, F., Coppola, E., Solmon, F., Mariotti, L., Sylla, M., Bi, X., Elguindi, N., Diro, G., Nair, V., Giuliani, G., Turuncoglu, U. U., Cozzini, S., Güttler, I., O'Brien, T. A., Tawfik, A. B.,Shalaby, A., Zakey, A. S., Steiner, A. L., Stordal, F., Sloan, L. C., and Brankovic, C.: RegCM4: model description and preliminary tests over multiple CORDEX domains, Clim. Res., 52, 7–29, 2012. 6556

Grell, G. A., Dudhia, J., and Stauffer, D. R.: A description of the fifth–generation Penn State/NCAR Mesoscale Model (MM5), Tech. Rep. NCAR/TN-398+STR, National Center for Atmospheric Research, 1995. 6556

Jerez, S., Montavez, J., Jimenez-Guerrero, P., Gomez-Navarro, J., Lorente-Plazas, R., and Zorita, E.: A multi-physics ensemble of present-day climate regional simulations over the Iberian Peninsula, A multi-physics ensemble of present-day climate regional simulations over the Iberian Peninsula, Clim. Dynam., 40, 3023–3046, doi:10.1007/s00382-012-1539-1, 2013. 6557

GMDD
8, 6551–6582, 2015

Jiménez, P. A. and Dudhia, J.: Improving the representation of resolved and unresolved topographic effects on surface wind in the WRF model, J. Appl. Meteor. Climatol., 51, 300–316, 2012. 6558

Jiménez-Guerrero, P., Montávez, J., Domínguez, M., Romera, R., Fita, L., Fernández, J., Cabos, W., Liguori, G., and Gaertner, M. A.: Description of mean fields and interannual variability in an ensemble of ERA-Interim forced simulations over peninsular Spain: results from the ESCENA project, Clim. Res., submitted, 2013. 6558

Menéndez, M., García-Díez, M., Fita, L., Fernández, J., Méndez, F. J., and Gutiérrez, J. M.: High-resolution sea wind hindcasts over the Mediterranean area, Clim. Dynam., 42, 1857–1872, doi:10.1007/s00382-013-1912-8, 2014. 6558, 6569, 6581

Mooney, P., Mulligan, F., and Fealy, R.: Evaluation of the sensitivity of the Weather Research and Forecasting model to parameterization schemes for regional climates of Europe over the period 1990–1995, J. Climate, 26, 1002–1017, 2013. 6557, 6558

Nikulin, G., Jones, C., Samuelsson, P., Giorgi, F., Sylla, M., Asrar, G., Büchner, M., Christensen, O., Déqué, M., Fernandez, J., Hänsler, A., van Meijgaard, E., Samuelsson, P., Sylla, M. B., and Sushama, L.: Precipitation climatology in an ensemble of CORDEX-Africa regional climate simulations, J. Climate, 25, 6057–6078, 2012. 6558, 6569, 6582

Redler, R., Budich, R., Ford, R., and Riley, G.: Earth System Modelling – Volume 5: Tools for Configuring, Building and Running Models, Springer, 1st Edn., 2012. 6552, 6554

Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Duda, M., Wang, W., and Powers, J.: A description of the Advanced Research WRF Version 3, Tech. rep., NCAR, 2008. 6553

Vautard, R., Gobiet, A., Jacob, D., Belda, M., Colette, A., Déqué, M., Fernández, J., García-Díez, M., Goergen, K., Güttler, I., Halenka, T., Karacostas, T., Katragkou, E., Keuler, K., Kotlarski, S., Mayer, S., Meijgaard, E., Nikulin, G., Patarcic, M., Scinocca, J., Sobolowski, S., Suklitsch, M., Teichmann, C., Warrach-Sagi, K., Wulfmeyer, V., and Yiou, P.: The simulation of European heat waves from an ensemble of regional climate models within the Euro-CORDEX project, Clim. Dynam., 41, 2555–2575, 2013. 6558

GMDD
8, 6551–6582, 2015

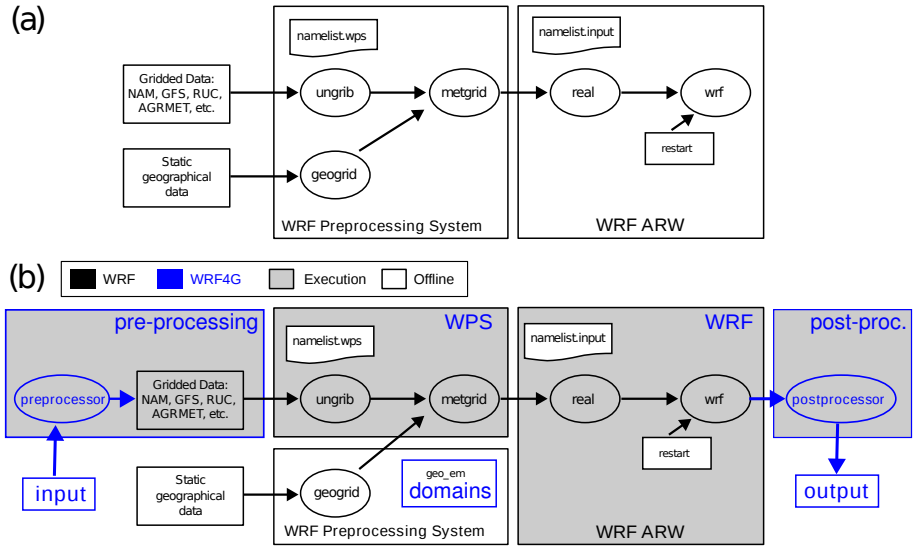

Figure 1. Schematic representation of the WRF workflow (for a real case experiment). The ellipses represent the executable binaries, and the rectangles the intermediate files produced by them. **(a)** represents the WRF workflow alone, while **(b)** represents the WRF workflow as it is embedded in WRF4G. The gray background area is the part of the workflow executed by WRF4G. The white background must be run and the output provided by the user.

GMDD
8, 6551–6582, 2015

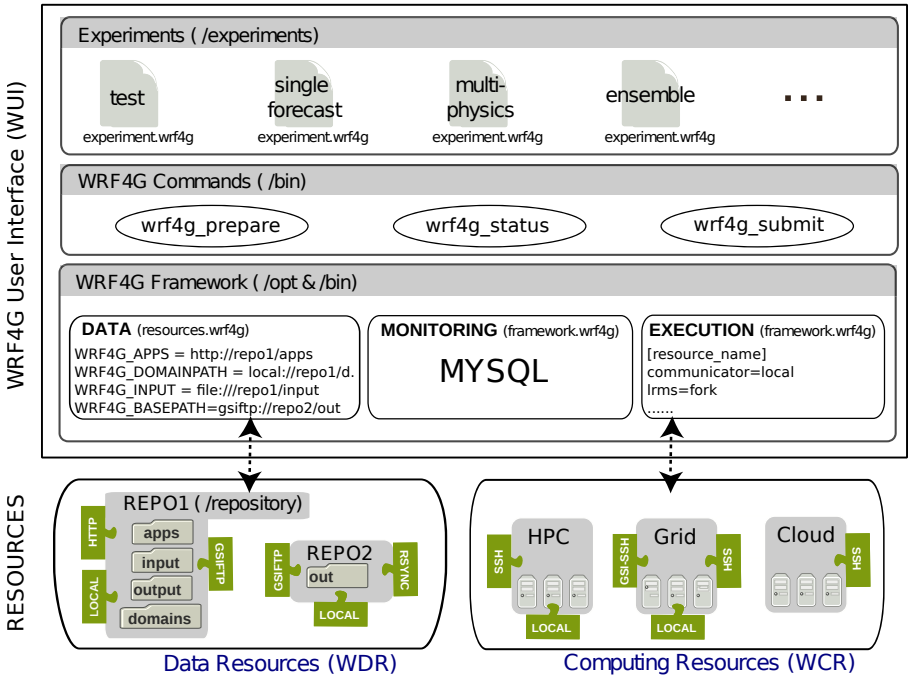

Figure 2. Scheme of the WRF4G framework.

GMDD
8, 6551–6582, 2015

```
[my_workstation]
communicator      = local
lrms              = fork
max_jobs_running  = 2

[pbs_cluster]
communicator      = ssh
username          = user1
frontend          = pbs.cluster.frontend
lrms              = pbs
queue             = medium
max_jobs_running  = 30
max_jobs_in_queue = 10

[slurm_cluster]
communicator      = ssh
username          = user2
frontend          = slurm.cluster.frontend
lrms              = slurm
queue             = long
max_jobs_running  = 20
max_jobs_in_queue = 5
```

Figure 3. Example of a *framework4g.conf* configuration file composed of three sections, describing each of them a WCR. The first one (my_workstation) is accessed locally and the others (pbs and slurm clusters) are accessed through SSH. Both clusters have been configured to use a queue defined by the keys *max_jobs_running* and *max_jobs_in_queue*. Regarding the workstation resource, it only has the *max_jobs_running* value due to the fact that a *fork lrms* is not a queue system.

GMDD
8, 6551–6582, 2015

```
$ wrf4g_resources
HID OS          ARCH     JOBS(R/Q/T) LRMS   HOSTNAME
0   Debian7     x86_64       2/0/2  FORK   my_workstation
1   Centos6.5   x86_64    30/10/40  PBS    pbs_cluster
2   Scien.6.8   x86_64     20/1/21  SLURM  slurm_cluster
```

Figure 4. List of CRs obtained from the wrf4g_resources command. Field information: *HID* stands for host unique identification; OS stands for operating system; *ARCH* means architecture; *JOBS(R/Q/T)* number of jobs: R = running, Q = in queue, T = total; *LRMS* stands for local resource management system; *HOSTNAME* means the name of the resource.

GMDD
8, 6551–6582, 2015

```
REQUIREMENTS = ( ARCH = "x86_64" ) & ( LRMS = "PBS" )
RANK = QUEUE_MAXRUNNINGJOBS
```

Figure 5. Resource selection parameters. *REQUIREMENTS* indicates that the experiment has to run on resources with a *x86_64* architecture and a *PBS* queue system. And *RANK* will sort these resources by the maximum number of running jobs allowed in the resource.

GMDD
8, 6551–6582, 2015

```
$ wrf4g_status -l exp1
Realization JOB_ID  Stat Chunks       Comp.Res     WN  Run.Sta     ext  %
rea_exp1         1     R    2/2 my_workstation node01  Down. Bound.  -  50.00
```

Figure 6. `wrf4g_status` output when the second chunk of exp1 is downloading the boundary conditions. The columns indicate the realization name (Realization), a unique job identification (JOB_ID), the chunk status (Stat: P = prepared, W = waiting, R = running, D = done), the chunk running (Chunks: R = running, T = total), the computing resource running the chunk (Comp.Res), the worker node within the WCR (WN), the running stage of WRF workflow (Run.Sta.), the exit code (ext) and the percentage of the realization completed (%).

GMDD
8, 6551–6582, 2015

```
start_date="2001-01-01_00:00:00"
end_date="2007-01-01_00:00:00"
chunk_size_h=1440
multiple_dates=0
  simulation_interval_h=87672
  simulation_length_h=105192
rerun=0
multiple_parameters=1
  multiparams_variables="cu_physics,mp_physics,ra_sw_physics,ra_lw_physics"
  multiparams_nitems="${max_dom},${max_dom},${max_dom},${max_dom},${max_dom},${max_dom},${max_dom
  multiparams_combinations="1,3,3,3/1,6,3,3/1,10,3,3/2,6,3,3/3,6,3,3/3,4,4,4/3,10,3,3"
  multiparams_labels="BCCR/CRPGL/UHOH/IDL/UC/IPSL/UHOHmp"
```

Figure 7. A fragment of the *experiment.wrf4g* used to produce the simulations used in García-Díez et al. (2014).

GMDD
8, 6551–6582, 2015

```
start_date = "1989-01-01_06:00:00"
end_date = "2009-12-31_18:00:00"
chunk_size_h = 42
multiple_dates = 1
  simulation_interval_h = 24
  simulation_length_h = 42
multiple_parameters = 0
  multiparams_variables = "sf_sfclay_physics,bl_pbl_physics"
  multiparams_nitems = "${max_dom},${max_dom}"
  multiparams_combinations = "1,1/2,2/5,6/7,7"
```

Figure 8. A fragment of the *experiment.wrf4g* used to produce some of the simulations in Menéndez et al. (2014).

GMDD
8, 6551–6582, 2015

```
start_date = "1989-01-01_00:00:00"
end_date = "2009-12-31_18:00:00"
chunk_size_h = 336 # 14 days
multiple_dates = 0
  simulation_interval_h = 24
  simulation_length_h = 42
multiple_parameters = 0
```

Figure 9. A fragment of the *experiment.wrf4g* used to produce the simulation labeled UC-WRF31 in Nikulin et al. (2012).

GMDD
8, 6551–6582, 2015