Wikiprint Book

Title: Reforecast Tutorial

Subject: TracMeteo - WRF4GWRFReforecast

Version: 34

Date: 01/22/2022 07:20:30 PM

Table of Contents

Reforecast Tutorial

How to get driving model (NCEP) data.

In this example, the publicly available NCEP Reanalysis (run 1) data are going to be used. This data can be dowloaded from ?http://nomad3.ncep.noaa.gov/pub/reanalysis-1/6hr in GRIB format. These are monthly files that get updated each month nearly in real time. Two files are needed for each month, one with the pressure level data, labeled "pgb", and other one with 2D data, labeled "grb2d". extdata_path defined in experiment.wrf4g must point to the folder where these files are located. Alternatively, it is possible to write a preprocessor that downloads the data itself. Note that the file names must be parsed by the preprocessor. In this case, if both files are located into the same folder, and provided the extension ".grb" is appended to them, the default preprocessor will parse them correctly, since it looks for monthly files with year/month (YYYY/mm) into their names. For example, the files for December 2010 should be:

```
grb2d201001.grb
pgb.ft00.201001.grb
```

Creating a WRF experiment

Keeping organized

Before starting to create an experiment, is good practice to create some directories to be tidy. For example, if our project is called "seawind", we can create the following directory hierarchy.

```
projects/seawind/submit/exp1
projects/seawind/submit/exp2
...
projects/seawind/domains
projects/seawind/data
projects/seawind/scripts
projects/seawind/figures
```

Of course, many other combinations are possible, depending in the organization of the resources available to the user.

The test experiment

Before creating a large experiment, with many chunks and realizations, it is convenient to run a smaller test experiment with exactly the same model configuration. This way we can see that everything is working as we want. Frequently, some attempts are needed before WRF runs, because of mistakes in the configuration files or in the set up of input files.

Go to the "submit" folder and create another folder called "sw_test":

```
cd projects/seawind/submit
mkdir sw_test
cd sw_test
```

Now we need to copy here the templates of experiment.wrf4g and resources.wrf4g.

```
cp $WRF4G_LOCATION/experiments/wrfuc_single_serial/experiment.wrf4g .
cp $WRF4G_LOCATION/etc/resources.wrf4g .
```

Now we can configure our test experiment, following the instructions in WRF4Gexperiment_wrf4g and WRF4Gresources_wrf4g. Note that, as it is a reforecast, we need to use the multiple dates configuration variables.

Once we are finished, we can use the command line interface (CPI) to prepare and submit our test experiment. Before submitting check in the output of wrf4g_prepare that the chunks and realizations being created are those that we wanted.

```
wrf4g_prepare
wrf4g_submit -e sw_test
```

Now we can monitor the experiment using wrf4g_status, along with the "watch" command.

```
watch wrf4g_status -l -e sw_test
```

Probably it will fail in the first attempts, so don't worry. If it fails, see how to manage WRF4G errors to look for the error.

Once the test experiment has run successfully, we should check that the output looks fine and that it contains all the variables that we want. Tools like ?ncview and ?ncdump are very useful for this task.

The experiment

At this point we are done with the most difficult issues, and we are ready to set up the reforecast experiment itself. Simply create another folder inside "submit" named "sw_ncep", and copy there the configuration files of the test experiment.

```
cd ..
mkdir sw_ncep
cp -r ../sw_test/* .
```

Then change the experiment_name and extend the end_date to the end date of the full experiment. In this example we will only run one month (January 2010), but it could be many decades. So our dates would be:

```
start_date="2009-12-31_18:00:00"
end_date="2010-01-31_06:00:00"
```

Finally, prepare and submit the experiment with WRF4G CLI, as before.

```
wrf4g_prepare
wrf4g_submit -e sw_test
```

Experiment monitoring

wrf4g_status permits us to monitor the state of all the realizations of the experiment. If the experiment is large, wrf4g_status can be used in combination with shell tools as grep or awk to filter the list with some criteria. For example:

```
wrf4g_status -l -e sw_test | grep Finished
```

Returns all the Finished realizations. Or, more complicated

```
wrf4g_status -l -e sw_test | grep -v Finished | grep -v Submitted | grep -v Failed
```

Returns all the realizations that are currently in some stage of the WRF4G workflow (Down. Bin., ungrib, metgrid..., etc.)

If wrf4g_submit is called again, the realizations in Failed status are submitted again. This is very useful to resubmit simulations after some problems with the infrastructure.

Other more complicated situation can occur. For example, if there is a blackout, the jobs can fail before they can send the "Failed" signal to the database. In that case, they can appear to be indefinitely in "WRF" status. These kind of problems need a less confortable monitoring, such as entering to the working nodes and use the commands "top" or "ps -ef" to see it WRF is really running.

When a realization has failed but does not appear with the "Failed" status, it can be resubmitted using the --force flag of wrf4g_submit. Note also that wrf4g_submit can be used refering only to one realization or chunk, using the flags, -r or -c.