

Wikiprint Book

Title: Resource Configuration

Subject: TracMeteo - WRF4G2.0/Resources

Version: 16

Date: 08/12/2022 08:58:29 PM

Table of Contents

Resource Configuration	3
Configuration format	3
DEFAULT section	3
Resource section	3
Examples	3

Resource Configuration

The configuration file `resources.conf` is used to describe computing resources. When you start WRF4G, `resources.conf` file is copied under `~/wrf4g/etc` directory if it does not exist. The file can be edit directly or by executing `wrf4g resource edit` command.

Configuration format

The configuration resource file consists of sections, each led by a `[section]` header, followed by `key = value` entries. Lines beginning with `#` are ignored. Allowing sections are `[DEFAULT]` and `[resource_name]`.

DEFAULT section

The DEFAULT section provides default values for all other resource sections.

Resource section

Each resource section has to begin with the line `[resource_name]` followed by `key = value` entries.

Configuration keys common to all resources:

- `enable`: true or false in order to enable or disable a resource.
- `communicator` or authentication type :
 - `local`: The resource will be accessed directly.
 - `ssh`: The resource will be accessed through ssh protocol.
- `username`: username to log on the front-end.
- `frontend`: The front-end either of a cluster or a grid user interface . The syntax is "host:port", by default the port used is 22.
- `private_key`: Private key identity file to log on the front-end.
- `scratch`: Directory used to store temporary files for jobs during their execution, by default, it is `$HOME/~/wrf4g/jobs`
- `lrms` or Local Resource Management System :
 - `pbs`: TORQUE/PBS cluster.
 - `sge`: Grid Engine cluster.
 - `slurm`: SLURM cluster.
 - `slurm_res`: [?RES\(Red Española de Supercomputación\)](#) resources.
 - `loadleveler`: LoadLeveler cluster.
 - `lsf`: LSF cluster.
 - `fork`: SHELL.
 - `cream`: CREAM Compute Elements (CE).

Keys for non-grid resources such as HPC resources:

- `queue`: Queue available on the resource. If there are several queues, you have to use a "," as follows "queue = short,medium,long".
- `max_jobs_in_queue`: Max number of jobs in the queue.
- `max_jobs_running`: Max number of running jobs in the queue.
- `parallel_env`: It defines the parallel environments available for Grid Engine cluster.
- `project`: It specifies the project variable and is for TORQUE/PBS, Grid Engine and LSF clusters.

Keys for grid resources:

- `vo`: Virtual Organization (VO) name.
- `host_filter`: A host list for the VO. Each host is separated by a ",". Here is an example: "host_filter = prod-ce-01.pd.infn.it, creamce2.gina.sara.nl".
- `bdii`: It indicates the BDII host to be used. The syntax is "bdii:port". If you do not specify this variable, the `LCG_GFAL_INFOSYS` environment variable defined on the grid user interface will be used by default.
- `myproxy_server`: Server to store grid credentials. If you do not specify this variable, the `MYPROXY_SERVER` environment variable defined on the grid user interface will be used by default.

Examples

By default, WRF4G is going to use the local machine as fork lrms:

```
[localmachine]
enable          = true
communicator    = local
frontend        = localhost
lrms            = fork
max_jobs_running = 1
```

TORQUE/PBS cluster, accessed through ssh protocol:

```
[meteo]
enable          = true
communicator    = ssh
username        = user
frontend        = mar.meteo.unican.es
private_key     = ~/.ssh/id_rsa
lrms            = pbs
queue           = short, medium, long
max_jobs_running = 2, 10, 20
max_jobs_in_queue = 6, 20, 40
```

ESR virtual organization, accessed through a grid user interface:

```
[esrVO]
enable          = true
communicator    = local
username        = user
frontend        = ui.meteo.unican.es
lrms            = cream
vo              = esr
bdii            = bdii.grid.sara.nl:2170
myproxy_server  = px.grid.sara.nl
```