



Enabling Grids for E-science

GridWay Job Template Manager

The ultimate tool for parameter sweeping

A. Lorca (UCM)

Introduction to gLite & RESPECT Tools at EGEE'09
Conference (Barcelona)
19 September 2009

www.eu-egee.org



- **The situation:**

User wants to send **many** jobs to the grid

```
gwuser@ui:$ worker argument
```

- If the arguments are of this form:
0, 1, 2, 3, ... very easy, use Job arrays in template with index $\${TASK_ID}$ up to $\${TOTAL_TASKS}$
- If the arguments are of this form:
6, 9, 12, 15, ... possible, use Job arrays in template with **START=6** and **INCREMENT=3** as gwsubmit flags

- **What if the argument are of the kind...?**
- Non-positive
-1, -2, -3
- Or real
0.25, 0.5, 0.75, 1 ...
- Not linearly spaced:
1, 10, 100, 1000 ...
- Strings
cat, dog, bat

GridWay cannot help you so far...

- Scientists and other grid users **often** need such non-integer, real, not linearly separated or string-like ARGUMENTS for their jobs to compute interesting stuff
- They do **not** want to script themselves
- **Bugs** appear easily
- Loop over other than integers require some math engine
- Automate, Test, Submit, Control, Monitor, Delete...

Now a tool **does all this** and more:

- `--create` (job templates)
- `--delete` (job templates)
- `--submit` (jobs from job templates)
- `--purge` (jobs from job templates)
- `--kill` (jobs from job templates)
- `--info` (jobs from job templates)

That's why I call it **Manager**

The GridWay Job Template Manager requires:

- Command line instruction

gw_job_template_manager

- A subcommand as a 1st argument:

--create

- An argument file with input parameter description

arguments.in

- An executable file to submit to the grid with the arguments

worker

- Altogether:

```
gwuser@ui:$ gw_job_template_manager -create arguments.in worker
```

Arguments should be defined in the input file:

- **LOOPTYPE** specifies the structure of data, it can be:
 - RANGE (linear range)
 - EXPRANGE (exponential range)
 - LIST (set of individual values)
- **START** and **END** mark the initial and final values
 - Require LOOPTYPE=RANGE or EXPRANGE
- **POINTS** or **STEP** instruct the spacing between points
 - Require LOOPTYPE=RANGE or EXPRANGE
- **FUNCTION** performs transformation on RANGE points
- **VALUE** picks up a single datapoint or string in LIST
- **SKIP** jumps over a specific datapoint

The argument.in file should look like

- `LOOPTYPE=RANGE START=-1 END=-3 STEP=-1`
`-1, -2, -3`
- `LOOPTYPE=RANGE START=0.25 END=1 POINTS=4`
`0.25, 0.5, 0.75, 1`
- `LOOPTYPE=EXPRANGE START=1 END=1000 POINTS=4`
`1, 10, 100, 1000`
- `LOOPTYPE=LIST VALUE=cat VALUE=dog VALUE=bat`
`cat, dog, bat`

You can do much more... TRY IT!

Want to know more?

- Written in perl
- Apache License 2.0
- Development under <http://dev.gridway.org>
- Subprojects
- Web page: <http://gridway.org>
Ecosystem → Related Software

It's demo time:

<ssh demo.gridway.org>

**¡Muchísimas
Gracias!
Questions?**