



Enabling Grids for E-scienceE

GridWay Job Template Manager

The ultimate tool for parameter sweeping

A. Lorca (UCM)

EGEE'09 Conference (Barcelona)
22 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:

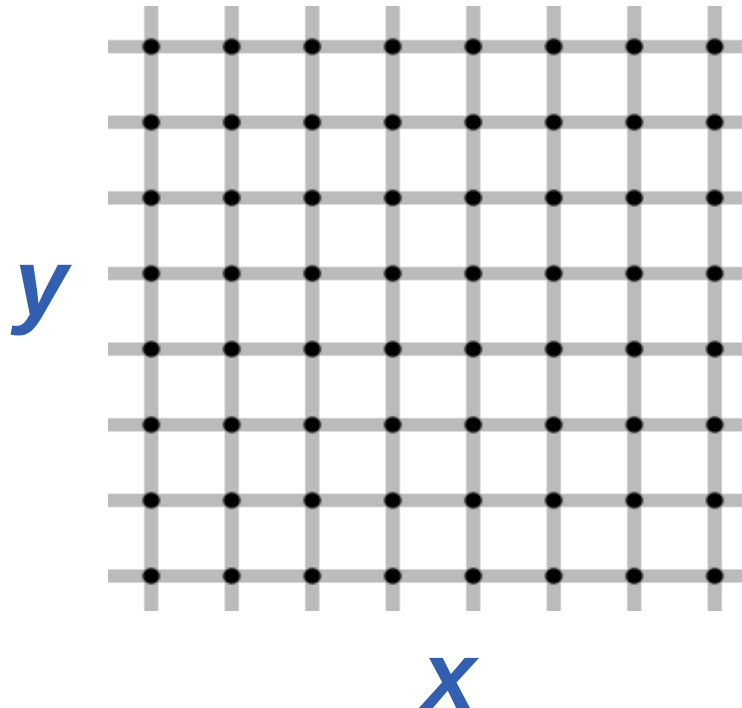
- **LOOPTYPE** specifies the structure of data, it can be:
 - RANGE (linear range)
 - EXPRANGE (exponential range)
 - LIST (set of individual values)
- **START** and **END** mark the initial and final values
 - Require LOOPTYPE=RANGE or EXPRANGE
- **POINTS** or **STEP** instruct the spacing between points
 - 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!

- Obviously, one-argument is a limited situation
- What if we need to scan a set of (x,y) pairs?



just add a 2nd line to the arguments' file!

- You are allowed to combine several LOOPTYPE styles into the arguments file:

```

LOOPTYPE=LIST VALUE=Taylorexpansion VALUE=Exactanalytic
LOOPTYPE=EXPRANGE START=1 END=10000 STEP=2
LOOPTYPE=RANGE START=-0.5 END=0.5 STEP=0.5 SKTP=0
    
```

You GET then these templates:

00~worker~Taylorexpansion_1_-0.5.jt	08~worker~Exactanalytic_1_-0.5.jt
01~worker~Taylorexpansion_1_0.5.jt	09~worker~Exactanalytic_1_0.5.jt
02~worker~Taylorexpansion_10_-0.5.jt	10~worker~Exactanalytic_10_-0.5.jt
03~worker~Taylorexpansion_10_0.5.jt	11~worker~Exactanalytic_10_0.5.jt
04~worker~Taylorexpansion_100_-0.5.jt	12~worker~Exactanalytic_100_-0.5.jt
05~worker~Taylorexpansion_100_0.5.jt	13~worker~Exactanalytic_100_0.5.jt
06~worker~Taylorexpansion_1000_-0.5.jt	14~worker~Exactanalytic_1000_-0.5.jt
07~worker~Taylorexpansion_1000_0.5.jt	15~worker~Exactanalytic_1000_0.5.jt

- Sometimes user needs a way to refer to the arguments (`_${TASK_ID}` or `_PARAM_` in JDL) or job identifier
- Implemented via \$variables:
 - `_${1}` to `_${n}` stands for the argument index
 - `_${JT_ID}` stands for the job template id
- Or to evaluate datapoints with an specific function:
Here a random generator for 5 values between 0 and 1:
`LOOPTYPE=RANGE START=1 END=1 POINTS=5 FUNCTION=rand`

```
0~worker~0.105178940183581.jt
1~worker~0.820890408332588.jt
2~worker~0.493093521605385.jt
3~worker~0.374937940944569.jt
4~worker~0.755138848338067.jt
```

Want to know more?

- Written in **perl**
- Free: Apache License 2.0
- **Development** under <http://dev.gridway.org>
Subprojects
- **Web** page: <http://gridway.org>
Ecosystem → Related Software
- Demo successful at the “*Introduction to RESPECT tools last saturday*”

- **A tool for parameter sweeping:**
 - Several value **types**: **LIST**, **RANGE** and **EXPRANGE**
 - **Multi-dimensional**
 - **Wildcards** allowed
 - **Function** evaluation allowed
 - Requires an **argument file** and an **executable** to be present
- **Using it is a command line experience**
 - Similar and equivalent to GridWay cli
 - Compatible con GridWay 5.4 and above
 - Create, Delete, Submit, Purge, Kill and Info subcommands
- **Enhance GridWay functionality**
 - Designed to provide the user with a powerful tool

**¡Muchísimas
Gracias!
Questions?**