



Web interface for generic grid jobs

Antònia Tugores, Pere Colet
 IFISC (CSIC-UIB) Palma de Mallorca – Spain.
 antonia@ifisc.uib-csic.es,
 pere@ifisc.uib-csic.es

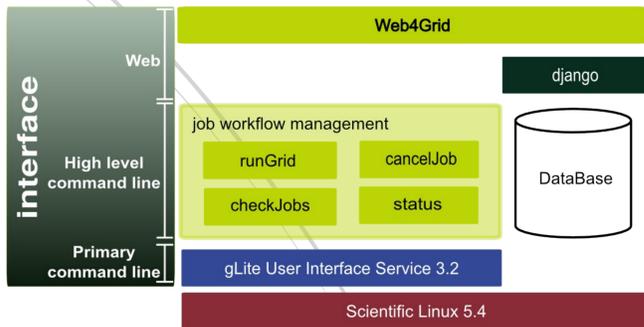


Introduction

Scientific grid is a useful tool in some very computationally demanding fields as for example in analysis of particle physics or astrophysics data. Grid is still associated to large projects, but there are many research areas in which small teams or even individual researchers may need to run many jobs in an efficient manner. While this high throughput computational needs are very much suitable for grid, few users take advantage of it because the access is cumbersome and requires a learning period that many researchers can not afford. Some user-friendly web applications have been developed for very specific tasks. General purpose graphical environments such as P-Grade are too complex to be used without prior training. Web4Grid is a user-friendly web interface allowing easy submission of jobs, automatic recovery of the results, monitoring of existing jobs and checking grid status. Web4Grid does not requires specific training on grid usage nor knowledge on gLite middleware.



Software Structure and Interaction

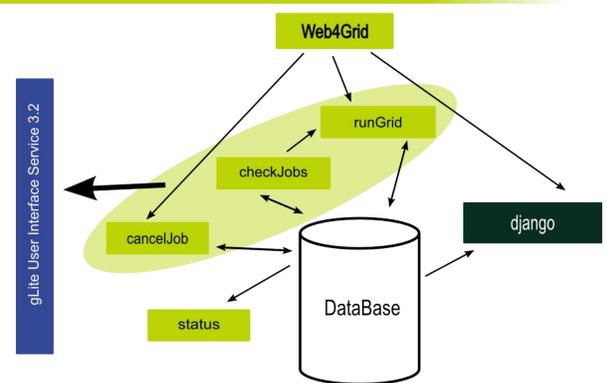


A high level command line interface is running over the primary gLite command line interface. Jobs can be submitted with:

```
runGrid -a application -p params -i inputFiles
```

On the top of these scripts, the web interface Web4Grid has been developed as a Django application.

A database is used to keep information about the jobs currently running. Most of the scripts interact with the database and the gLite User Interface package. Web4Grid just launches those scripts and presents information stored in the database.



Job management (scripts)

The set of scripts responsible for the job management allow users to:

- Submit jobs to the grid and monitor all the job workflow with an unique command
- Cancel jobs with an unique command
- Visualize owned jobs status formatted as a table

Main features:

- Long term proxy always used (password just needed once)
- Job submission just setting the application, the input files and the command line parameters
- Other variables (VO, SE, LFC, ...) are set automatically from the user configuration file or default values are used.
- Inputs are uploaded to the Storage Element without human interaction
- No user application modification is needed to run any executable in the grid, a wrapper manages the data in the Worker Node and runs the user program
- Results are retrieved and stored automatically in the directory the job was submitted from

The script used to submit jobs can be launched directly from the command line interface or indirectly from the web. After submitting the job to the grid, the script exits and a daemon checks the running jobs status every five minutes. When the job has finished, the daemon launches the main script again and it downloads the outputs to the directory the main script was submitted from. (See schema below)

runGrid FSM states	associated gLite commands
INIT	myproxy-init voms-proxy-info myproxy-logon
UPLOADING	lfc-mkdir lfc-cr lfc-chmod
READY	glite-wms-job-list-match glite-wms-job-submit
SUBMITTED	
RUNNING	glite-wms-job-status (1x user / 5min) lfc-cp lfc-cr
EXECUTED	
DOWNLOADING	glite-wms-job-output lfc-cp
CLEANING	lfc-del lfc-rm
DONE	myproxy-destroy voms-proxy-destroy

Job workflow is managed through a Finite States Machine implemented by two main scripts ran in the User Interface. These two scripts wrap up to 20 gLite commands.

gLite job status



Worker Node Wrapper

- set environment variables
- download inputs file (tgz) from SE
- untar inputs
- execute the application
- tar the results (do not tar input files)
- upload output file to SE
- return the application exitCode

A wrapper for the user application is created and executed in the Worker Node in order to download/upload inputs/outputs from/to the Storage Element.

Web4Grid

- Integrated LDAP authentication
- Job submission
- Job monitorization
- Job cancellation
- Check the grid status (free/busy cores)
- No need to upload files because home directory is shared all across the institute



Usage

Web4Grid implementation is recent and no statistics can be reported. The high level command line interface has been running for more than six months on the last three months activity has been analyzed. The most relevant data is the number of jobs run launched.

