




Using DALTON 2.0 in Grid.

Program DALTON 2.0 is now available in 4 Balticgrid clusters. These are: grid2.mif.vu.lt, ce01.grid.etf.rtu.lv, pupa.elen.ktu.lt and kriit.eenet.ee.

It can use with certificate which is published from **Balti Grid CA**. First step that must be done is installing grid user interface to your desktop computer. Very useful program for this procedure is UIPnP (User Interface Plug and Play) package with Grid Zeus Toolkit, which is integrated for using to manage large number of jobs. (Tarball can be download from <http://voms.balticgrid.org/UIPnP-BG/UIPnP.zeus.gLite.tar.gz>)

UIPnP instalation

Installing UIPnP is very easy. It is compatible with majority of Linux distributions. It is necessary to have approximately 1.2 Gb of free hard disk space for this operation. UIPnP is compiled before and ready to use after unpack. It is adjusted to using in Balticgrid VO members. For installing:

```
cd
wget http://voms.balticgrid.org/UIPnP-BG/UIPnP.zeus.gLite.tar.gz
tar -zxvf UIPnP.zeus.gLite.tar.gz
cd UIPnP
```

NB! It is suggested to check UIPnP.sh file and write there correct path where UIPnP is unpacked. (by default it is /home/{your_user_name}/UIPnP) Then:

```
source UIPnP.sh
```

Dalton`s current job directory is "\$UIPnP/jobs/dalton". All Dalton`s jobs should be copied into this directory. Every Dalton job should be a package that includes files what is needed by Dalton and job running script which will run current Dalton job in the grid. The script - dalton-runjob.sh - is current job running script. (check example)

Sending one job

For detailed description about sending jobs to the grid can be found from balticgrid homepage (http://www.balticgrid.org/Using_grid/Try_grid/) Submitting example job:

```
voms-proxy-init -voms balticgrid
cd $UIPnP/jobs/dalton
glite-job-submit -o jobs.out dalton-example.jdl
```

For checking job current status:

```
glite-job-status -i jobs.out
```

Get job result (download to current directory):

```
glite-job-output -i jobs.out -dir .
```

Sending several jobs

If job packets are compiled (every job is separate package) then "jobsub" should be modified. Each job is described by one line. The command-script that will run all jobs that is described in "jobsub" file, is "jobsub.pl" (For more information: [jobsub.pl --help](#)) For sending jobs do this:

```

$ ./jobsub
Jobid is: https://grid4.mif.vu.lt:9000/cy_-J3a1587t4EzZqICXWg
User id: 1
Jobid is: https://grid4.mif.vu.lt:9000/KeiRVmAhuFcOBswAEp4fVQ
User id: 2

```

Every job will be saved into database. Default database is "jobs" (the file with same name in current directory) For checking database status try:

```
$ check_db.pl jobs
```

For checking jobs status: (you should see something similar)

```

$ jobq.pl -all
00001 https://grid4.mif.vu.lt:9000/cy_-J3a1587t4EzZqICXWg RUNNING dalton-test-short
00002 https://grid4.mif.vu.lt:9000/KeiRVmAhuFcOBswAEp4fVQ READY dalton-test-medium

```

Detailed info about current job: jobq -v [job number]

```

$ jobq.pl -v 1
00001 https://grid4.mif.vu.lt:9000/cy_-J3a1587t4EzZqICXWg RUNNING dalton-test-short
info: cancelling = 0
info: destination = pupa.elen.ktu.lt:2119/jobmanager-lcgpbs-balticgrid
info: status = 5
info: lastUpdateTime = Mon Sep 18 13:47:38 2006
info: condorId = 216496
info: seed = uLU0BAr rdV98041PLThJ5Q
info: done_code = 0
info: expectUpdate = 0
info: status_reason = Job successfully submitted to Globus
info: owner = /DC=org/DC=balticgrid/OU=eenet.ee/CN=Joasas Kurtzawoziwas
info: cpuTime = 0
info: location = LRMS/worknode/pupa.elen.ktu.lt
info: resubmitted = 0
info: network_server = grid4.mif.vu.lt:7772
info: children_hist = 0
info: children_num = 0
info: jobtype = 0
info: subjob_failed = 0

```

Job status "DONE" means that jobs computing process is ended successfully. There is no errors from job sending to the end of computing. When job has status "DONE" is possible to get job output, retrieve and download job results to your computer. For this operation is command-script "jobget.pl" (For example if you want to get output of all jobs that have ended successfully, try: jobget.pl --all) Jobs are saved into directory "jobOutput/" When job has been downloaded its status will become "CLEARED". Then the job is removed from resource broker and it should be removed from job managing database. For this operation is command-script "jobpurge.pl" (For more: jobpurge.pl --help)

Every job output consists of at least two files - "std.out" and "std.err" First one is standard output that would be displayed in screen. It reflects the whole job process. Second consists of all the errors that occurred in job performance.

If job has been cancelled by some error then it has status "FAILED". Usually it means that problem is somewhere in grid infrastructure and job should be run again. For running job again do:

```
$ jobresub.pl [job number]
```