

GRAM5

Introduction for client commands

GRAM5: general hints for client

- If your job seems to get stuck try to kill your job-manager processes:
`killall globus-job-manager`
- For logs see your home directory (`ls -lart gram*`)
- See also in `$HOME/.globus/job/`
- Gatekeeper log
 - `$GLOBUS_LOCATION/var/gatekeeper.log`
 - It might be visible for administrator only.

GRAM5: globus-job-run blocking submission

- With `globus-job-run` it is simple to submit a job
- It is a blocking command (i.e. it does not release the shell until the job finishes)
- Example: `globus-job-run <your hostname> /bin/date`
- It is possible to give various parameters e.g. directing standard output or error. See `-help` or User Guide <http://bit.ly/c8FYK0>

GRAM5: globus-job-submit non-blocking command submission (1)

- `globus-job-submit` returns command to shell right after the submission and gives **job contact string**
- `globus-job-status <job_contact_string>`
- `globus-job-get-output <job_contact_string>`
- `globus-job-clean <job_contact_string>`
 - needed after job status is DONE
- `gs002@gks-1-101:~> globus-job-submit gks-1-101.fzk.de /bin/date`
<https://gks-1-101.fzk.de:24384/16073723895661987071/15700714982003976859/>
- `gs002@gks-1-101:~> globus-job-status https://gks-1-101.fzk.de:24384/16073723895661987071/15700714982003976859/`

DONE

GRAM5: globus-job-submit non-blocking command submission (2)

- `gs002@gks-1-101:~> globus-job-get-output https://gks-1-101.fzk.de:24384/16073723895661987071/15700714982003976859/`
Sat Sep 4 21:02:43 CEST 2010
- `gs002@gks-1-101:~> globus-job-clean https://gks-1-101.fzk.de:24384/16073723895661987071/15700714982003976859/`

WARNING: Cleaning a job means:

- Kill the job if it still running, and
- Remove the cached output on the remote resource

Are you sure you want to cleanup the job now (Y/N) ?

Y

Cleanup successful.

GRAM5: globusrun and RSL (1)

- `globusrun` command is the most suitable for real "production" jobs
- It takes as a parameter a script written in Globus Resource Specification Language (RSL).
- GRAM5 uses different syntax than Globus version 4.
- RSL script can be passed:
 - from a command-line (in " ")

```
gs002@gks-1-101:~> globusrun -s -r gks-1-101.fzk.de "&(executable=/bin/date)"  
Sat Sep  4 21:10:40 CEST 2010
```
 - in an RSL file

GRAM5: globusrun and RSL (2)

- The simplest RSL script is specifying the executable:

&(executable=/bin/date)

– Please store this line to a file `job.rsl`

- The `&` is needed only on the first row.
- All rows are surrounded in `()`.

GRAM5: globusrun command line parameters

- Submission which streams (-s) standard output and error to the display

```
globusrun -s -r <your host> -f job.rsl
```

```
Thu Aug 12 17:04:13 CEST 2010
```

- For complete list of possible attributes see <http://bit.ly/d6cQbL>

GRAM5: globusrun and RSL (3)

- Some useful RSL attributes:

```
& (rsl_substitution = (DIR "/tmp/my_dir" ) )  
(environment = (MSG 'Hello' ))  
(stderr = $(DIR)/stderr.txt)  
(stdout = $(DIR)/stdout.txt)  
(executable=/usr/bin/env)  
(* (arguments="Hello ") *)
```

- A variable set in `environment` is not possible to use in RSL script.

GRAM5: globusrun non-blocking operation (1)

- With `-b` option non-blocking command is sent and a contact string is then returned.
- Edit `job.rsl`:
`&(executable=/bin/sleep)`
`(arguments=1000)`
- Run:
`globusrun -b -r <your host> -f job.rsl`

GRAM5: globusrun

C

non-blocking operation (2)

- Status query:

```
globusrun -status <job_contact_string>
```

- Possible job statuses: ACTIVE, FAILED, SUSPENDED, DONE, UNSUBMITTED, STAGE_IN, STAGE_OUT and UNKNOWN JOB STATE

- Cancelling the job:

```
globusrun -k <job_contact_string>
```

GRAM5: File staging (1)

- The possible steps in a job are:
 - File **stage in**: files from the client to the GRAM5 server
 - File **stage out**: files from the GRAM5 server to the client
 - File **clean-up**: remove the files from the GRAM5 server
- Internal or external GridFTP can be used.
- To use internal file transfer mechanism (GASS) uses predefined variable:
`$(GLOBUSRUN_GASS_URL)/$(HOME)/input.txt`
- Used on the client side

GRAM5: File staging (2)

GASS example (1)

```
& (executable=$(HOME)/compile.sh)
(stdout=stdout.txt)
(stderr=stderr.txt)
(file_stage_in =
$(GLOBUSRUN_GASS_URL)/$(HOME)/compile.sh $(HOME)/
compile.sh))
(file_stage_out =
(stderr.txt $(GLOBUSRUN_GASS_URL)/$(HOME)/stderr.txt)
(stdout.txt $(GLOBUSRUN_GASS_URL)/$(HOME)/stdout.txt))
(file_clean_up=stdout.txt)
(file_clean_up=stderr.txt)
```

GRAM5: File staging (3)

GASS example (2)

- In previous job script the executable script (`compile.sh`) could have been for example:

```
#!/bin/bash -l  
mpicc mpi_test.c -o mpi_test  
chmod 755 mpi_test
```

- The command to submit the job:

```
globusrun -s -r <HOSTNAME> -f compile.rsl
```

GRAM5: File staging (4)

GridFTP example (1)

```
& (rsl_substitution = (GRIDFTP_SERVER gsiftp://<GFTP_HOST>))
(executable=/bin/cat)
(arguments=$(HOME)/input_file)
(stdout=stdout.txt)
(stderr=stderr.txt)
(file_stage_in = ($(GRIDFTP_SERVER)/$(HOME)/input_file $(HOME)/input_file))
(file_stage_out = (stderr.txt $(GRIDFTP_SERVER)/$(HOME)/stderr.txt)
                  (stdout.txt $(GRIDFTP_SERVER)/$(HOME)/stdout.txt))
(file_clean_up = $(HOME)/input_file)
```

Example how to use optional common variable `GRIDFTP_SERVER`

GRAM5: File staging (5)

GridFTP example (2)

- Try script from previous slide:
 - Download script: **wget http://tinyurl.com/filestage-rsl**
 - Fix the hostname.
- Create file **input.txt** to `$HOME` and put there some text.
- Run:
globusrun -s -r <HOSTNAME> -f filestage.rsl
- See output:
cat \$HOME/std*

GRAM5: MPI job example (1)

- MPI job should be submitted to batch scheduling systems. Since it will take time so `-b` option is used:

```
globusrun -s -b -r gram5.lrz.de/jobmanager-pbs  
-f mpigt5.rsl
```

- In the next slide is an example of MPI RSL job script.
 - The number of the MPI processes is set with `count`.
 - `job_type` must be set to `mpi`.
- You can set needed memory (in MBs) and wall-clock time (in minutes).

GRAM5: MPI job example (2)

RSL script gt5mpi.rsl (1)

```
&(executable=$(HOME)/mpi_test)
(job_type=mpi)
(count=2)
(max_wall_time=20)
(max_cpu_time=10)
(max_memory=10)
(stdout=stdout.txt)
(stderr=stderr.txt)
```

GRAM5: MPI job example (3)

RSL script gt5mpi.rsl (2)

```
(file_stage_out =  
(stderr.txt $(GLOBUSRUN_GASS_URL)/$(HOME)/  
stderr.txt)  
(stdout.txt $(GLOBUSRUN_GASS_URL)/$(HOME)/  
stdout.txt))  
(file_clean_up=stdout.txt)  
(file_clean_up=stderr.txt)
```

GRAM5: Advanced RSL: Proxy renewal operation & debug

- By default proxy certificate lives 12 hours
- If proxy expires and need to get results of the job:
 - `grid-proxy-init`
 - `globusrun -r <host> \`
`"&(restart=<job_contact_string>)"`
- New debug feature in v. 5.0.2:
to save Globus internal job descriptions add:
`(save_job_description=yes)`