# **Relion Prometheus usage guide**

- Interactive Relion job with Relion GUI
- Relion in SLURM batch jobs
- Submitting SLURM jobs from Relion GUI

Relion could be used at Prometheus supercomputer in three ways:

- inside graphical interactive job using pro-viz service (Main documentation polish Obliczenia w trybie graficznym: pro-viz)
- in SLURM batch job though SLURM script submitted from command line
- in SLURM batch job submitted from Relion GUI started via pro-viz service in dedicated partition

### Interactive Relion job with Relion GUI

In order to start interactive Relion job with access to Relion GUI

1. Log into Prometheus login node

Log into Prometheus login node

ssh <login>@pro.cyfronet.pl

2. Load pro-viz module

Load pro-viz module

module load tools/pro-viz

- 3. Start pro-viz job
  - a. Submit pro-viz job to qeuue
    - i. CPU-only job

Submission of CPU pro-viz job

pro-viz start -N <number-of-nodes> -P <cores-per-node> -p <partition/queue> -t <maximaltime> -m <memory>

ii. GPU job

Submission of GPU pro-viz job

pro-viz start -N <number-of-nodes> -P <cores-per-node> -g <number-of-gpus-per-node> -p <partition/queue> -t <maximal-time> -m <memory>

b. Check status of submitted job

Status of pro-viz job(s)

pro-viz list

c. Get password to  ${\tt pro-viz}$  session (when job is already running)\

Pro-viz job password

pro-viz password <JobID>

exemple output

#### Pro-viz password example output

```
Web Access link:
https://viz.pro.cyfronet.pl/go?c=<hash>&token=<token>
link is valid until: Sun Nov 14 02:04:02 CET 2021
session password (for external client): <password>
full commandline (for external client): vncviewer -SecurityTypes=VNC,UnixLogin,None -via
<username>@pro.cyfronet.pl -password=<password> <worker-node>:<display>
```

- d. Connect to graphical  ${\tt pro-viz}$  session
  - i. you could use weblink obtained in previous point
  - ii. you could use VNC client (i.e. TurboVNC). Configuration of client described in Obliczenia w trybie graficznym: pro-viz (in polish)

#### 4. Setup Relion environment

a. When connected to GUI open Terminal and load Relion module

Load Relion module module load plgrid/tools/relion

b. Start Relion GUI in background

Start relion			
relion &			

5. Use Relion GUI for computation.

Terminal - plgnoga@p0002:~/scratch/relion/2021-11-09/relion30_tutorial	↑ _ □ ×	REI	ION-3.1.2:	e/plgnoga/relion/2021-11-09/r	elion 30_tutorial 🔷 🕳 🖂
File Edit View Terminal Tabs Help		File Jobs Schedules	I/O CTF	Optimisation Sampling Hel	ix Compute Running
<pre>[plgnoga@p0002 relion30 tutorial]\$ module add plgrid/tools/relion</pre>		( Income	tweet terms		and the second se
plgrid/tools/binutils/2.34 loaded.		Motion correction		Number of MPI procs:	3 - 7
plgrid/tools/gcc/9.3.0 loaded.		CTF estimation		Number of threads:	8 7
plgrid/tools/intel/19.1.1 loaded.		Manual picking		Submit to queue?	No.
plgrid/libs/ucx/1.8.0 loaded.		Particle extraction		00000 00000	alarid any
plgrid/tools/openmpi/4.0.3-intel-19.1.1 loaded.		Subset selection		Queue name.	pigna-gpu
plgrid/libs/mkl/2021.1.1 loaded.		2D classification		Queue submit command:	sbatch
sh: plg-show-default-grant: command not found		3D classification		Walltime	1:00:00 ?
plgrid/tools/relion/3.1.2 loaded.		3D auto-refine		Memory	115GB ?
[plgnogadp0002 relion30_tutorial]s relion &		3D multi-body		Account	2
[1] 16972		CIF refinement Bayesian polishing		Standard submission script:	n-scripts/relion_slurm_gpu.sh ? Browse
[brguoga@baaas_rectonsa_recortar]\$		Mask creation	Min	imum dedicated cores per node:	24 7
		Join star files		Additional arguments	
		Particle subtraction Post-processing		Additional arguments.	
		Local resolution			
		External			
					Schedule Check command Run!
		Underson Link and and	C	Chus alles have	
		Job actions	Current	Clive_allas_here	Display:
		Finished jobs		Running jobs	Input to this job
		019: Class2D/job019/			
		017: Class2D/job017/			
		003: MotionCorr/iob003/			
software		001: Import/movies/			
				Scheduled jobs	Output from this job
tools					
		stdout will go here: double-cl	ick this wind	ow to open stdout in a separate win	dow
tmp					
		stderr will go here: double-cli	ck this winde	ow to open stderr in a separate wind	ow
File System					

6. After finishing work terminate job

Pro-viz job password	
pro-viz stop <jobid></jobid>	

## Relion in SLURM batch jobs

Most of Relion jobs could be run as batch jobs using SLURM

1. Log into Prometheus login node

```
Log into Prometheus login node
```

ssh <login>@pro.cyfronet.pl

2. Move to Relion project directory

#### Change directories

cd \$SCRATCH/<relion-project>

#### Usage of filesystems

Relion project during computations should be stored in \$SCRATCH filesystem on Prometheus. More info - https://kdm.cyfronet.pl /portal/Prometheus:Basics#Disk\_storage. For longer storage user should use \$PLG\_GROUPS\_STORAGE/<team\_name> filesystem.

#### 3. Submit job

#### Job submision

sbatch script.slurm

a. Example CPU-only SLURM script

#### **Relion CPU-only SLURM script**

```
#!/bin/bash
# Number of allocated nodes
#SBATCH --nodes=1
# Number of MPI processes per node
#SBATCH --ntasks-per-node=4
# Number of threads per MPI process
#SBATCH --cpus-per-task=6
# Partition
#SBATCH --partition=plgrid
# Requested maximal walltime
#SBATCH --time=0-1
# Requested memory per node
#SBATCH --mem=110GB
# Computational grant
#SBATCH --account=<name-of-grant>
export RELION_SCRATCH_DIR=$SCRATCHDIR
module load plgrid/tools/relion/3.1.2
mpirun <relion-command>
```

b. Example GPU SLURM script

#### **Relion GPU SLURM script**

```
#!/bin/bash
# Number of allocated nodes
#SBATCH --nodes=1
\ensuremath{\texttt{\#}} Number of MPI processes per node
#SBATCH --ntasks-per-node=4
# Number of threads per MPI process
#SBATCH --cpus-per-task=6
# Partition
#SBATCH --partition=plgrid-gpu
# Number of GPUs per node
#SBATCH --gres=gpu:2
# Requested maximal walltime
#SBATCH --time=0-1
# Requested memory per node
#SBATCH --mem=110GB
# Computational grant
#SBATCH --account=<name-of-grant>
export RELION_SCRATCH_DIR=$SCRATCHDIR
```

module load plgrid/tools/relion/3.1.2
mpirun <relion-command> --gpu \$CUDA\_VISIBLE\_DEIVCES

#### GPUs usage

GPUs are available only for selected grants in partitions plgrid-gpu and plgrid-gpu-v100. One should aways use --gpu \$CUDA\_VISIBLE\_DEIVCES to request GPUs allocated for job.

#### Relion command

Relion command syntax could be checked using GUI and copied to script



4. Check job status

Job submision	
squeue	
or	
Job submision	
pro-jobs	

## Submitting SLURM jobs from Relion GUI

- 1. Start job as in pro-viz session but using plgrid-services partition/queue.
- In Relion GUI use "Submit to queue" in "Running" tab
   a. Select submission scripts from directory

I/O	CTF	Optimisation	Sampling	He	ix Compute Running		
Number of MPI procs: 3 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7							
Submit to queue? Yes 🗧 7							
Queue name:			Queue nan	ne:	plgrid-gpu ?		
Queue submit command:			bmit comma	nd:	sbatch ?		
	Walltime			me	1:00:00		
Memory			Mem	ory	115GB ?		
Account			Acco	unt			
Standard submission script:			bmission scr	ipt:	n-scripts/relion_slurm_gpu.sh ? Browse		
	Mini	mum dedicated	cores per no	de:	24 ?		
		Additio	onal argume	nts:			
					$\frown$		
				(	Schedule Check command Run!		

3. Monitor jobs either from Relion GUI or command line using  $\tt squeue \ or \ pro-jobs \ commands$