# How to use Hercules

Grégory Desvignes[1]

[1]Max-Planck-Institut for Radio Astronomy

PSR Workshop, September 2017

# Getting access

Apply for account at :
`http://www.rzg.mpg.de/userspace/forms/onlineregistrationform`
Account will be approved by MPIfR Computer department.
`ssh <user-name>@hercules01.bc.rzg.mpg.de`
Transfer data to Hercules with 10 GbE link. Use hgw gateway.
`rsync <data> hgw:/hercules/results/<user-name>/.`

# Libraries

`gcc, icc, MKL, ipython`

Need something else : contact RZG
https ://helpdesk.rzg.mpg.de/rzg/index.html or

# Modules

`module avail` : List the available modules
`module load gsl` : Load GSL module
`module show gsl` : Show module environment variables
`module list` : List the loaded modules

# VNC Desktop

`module load vncsetup`

`vncsetup` : Need to be run once

`vncserver -geometry 1600x900` : Create the VNC session

`vncserver -kill:1` kills X display (needs to be adapted)

# Hercules : a description

- 3 login nodes + 184 compute nodes (4416 CPU cores)
- 1 node : 24 cores (wo. HT), 64 GB of RAM, 1 TB of scratch space ($TMPDIR) on each node
- 229 TB of GPFS storage
- No InfiniBand or fast interconnect !
- Slurm queue manager

# Hercules : the queues

3 partitions (queues) :

- short.q : up to 12 hr, max 1 node
- long.q up to 10 days, max 1 node
- p.64g : for MPI jobs, up to 2 days

Automatic selection for short or long queue depending on CPU time requested

# Hercules : a job example

```
#!/bin/bash -l
#SBATCH -D ./
#SBATCH --mail-user=yourmail@gmail.com
#SBATCH --mail-type=ALL (or NONE)
#SBATCH -t 2:30:0                  # CPU time
#SBATCH --partition=p.64g          # if MPI needed
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mem=10000                # in MB
```

# Hercules : some useful commands

`sinfo` : Shows the cluster status
`squeue` : Shows the entries in the queue
`scancel <job_id>` : Cancels job with id jobid
`scontrol show nodes` : Shows the nodes infos
`scontrol show partition` : Shows the partitions infos

# Hercules : job arrays

Automatically submit 1000 jobs, running 175 jobs max. at a time.

```
#SBATCH --array=0-1000:1%175
```

Then use `$SLURM_ARRAY_TASK_ID` in your script.

# Supplemental

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./J0613-0200/DE405_BIPM2015.out.%j
### change to current work dir
#SBATCH -D ./

#SBATCH --mail-type=ALL
#SBATCH --mail-user username@mpifr-bonn.mpg.de

#SBATCH -J I-J0613-0200/DE405_BIPM2015
# Queue (Partition):
#SBATCH --partition=p.64g
# Number of nodes and MPI tasks per node:
#SBATCH --nodes=20
#SBATCH --ntasks-per-node=24
#  wallclock 24 hours
#SBATCH -t 24:0:0
# virtual limit per node in MB
#SBATCH --mem=61400

source /u/caballrn/.bashrc

# export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
# export TEMPO2=/u/gdesvign/src/tempo2/T2runtime

cd /u/caballrn/path/
srun -l -n 480 tempo2 -gr temponest -f J0613-0200_DE405_BIPM2015.par J0613-0200.tim -npsr 1 > logfile.log
```

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./J0613-0200/DE405_BIPM2015.out.%j
### change to current work dir
#SBATCH -D ./

#SBATCH --mail-type=ALL
#SBATCH --mail-user username@mpifr-bonn.mpg.de

#SBATCH -J I-J0613-0200/DE405_BIPM2015
# Queue (Partition):
#SBATCH --partition=p.64g
# Number of nodes and MPI tasks per node:
#SBATCH --nodes=20
#SBATCH --ntasks-per-node=24
#  wallclock 24 hours
#SBATCH -t 24:0:0
# virtual limit per node in MB
#SBATCH --mem=61400

source /u/caballrn/.bashrc

# export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
# export TEMPO2=/u/gdesvign/src/tempo2/T2runtime

cd /u/caballrn/path/
srun -l -n 480 tempo2 -gr temponest -f J0613-0200_DE405_BIPM2015.par J0613-0200.tim -npsr 1 > logfile.log
```

Don’t reserve resources you don’t use!

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./J0613-0200/DE405_BIPM2015.out.%j
### change to current work dir
#SBATCH -D ./

#SBATCH --mail-type=ALL
#SBATCH --mail-user username@mpifr-bonn.mpg.de

#SBATCH -J I-J0613-0200/DE405_BIPM2015
# Queue (Partition):
#SBATCH --partition=p.64g
# Number of nodes and MPI tasks per node:
#SBATCH --nodes=20
#SBATCH --ntasks-per-node=24
#  wallclock 24 hours
#SBATCH -t 24:0:0
# virtual limit per node in MB
#SBATCH --mem=61400

source /u/caballrn/.bashrc

# export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
# export TEMPO2=/u/gdesvign/src/tempo2/T2runtime

cd /u/caballrn/path/
srun -l -n 480 tempo2 -gr temponest -f J0613-0200_DE405_BIPM2015.par J0613-0200.tim -npsr 1 > logfile.log
```

Don't reserve resources you don't use!

Limit=62 GB (2GB reserved)

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./location/name.out.%j
### change to current work dir
#SBATCH -D ./
#SBATCH --partition=long.q
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user username@mpifr-bonn.mpg.de
#SBATCH -J 999_MH_GW
#SBATCH --ntasks=24
#  wallclock hh:mm:ss hours
#SBATCH -t 16:0:0
# virtual limit per node in MB
#SBATCH --mem=8192
source /u/caballrn/.bashrc
cd /u/kjlee/KIAA308/
# export OMP_NUM_THREADS=2
--cpus-per-task==2

python2.7 Bayesian_Opt.py -dir > logfile.log
```

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./location/name.out.%j
### change to current work dir
#SBATCH -D ./
#SBATCH --partition=long.q
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user username@mpifr-bonn.mpg.de
#SBATCH -J 999_MH_GW
#SBATCH --ntasks=24
#  wallclock hh:mm:ss hours
#SBATCH -t 16:0:0
# virtual limit per node in MB
#SBATCH --mem=8192
source /u/caballrn/.bashrc
cd /u/kjlee/KIAA308/
# export OMP_NUM_THREADS=2
--cpus-per-task==2

python2.7 Bayesian_Opt.py -dir > logfile.log
```

Use the right partition!

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./location/name.out.%j
### change to current work dir
#SBATCH -D ./
#SBATCH --partition=long.q
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user username@mpifr-bonn.mpg.de
#SBATCH -J 999_MH_GW
#SBATCH --ntasks=24
#  wallclock hh:mm:ss hours
#SBATCH -t 16:0:0
# virtual limit per node in MB
#SBATCH --mem=8192
source /u/caballrn/.bashrc
cd /u/kjlee/KIAA308/
# export OMP_NUM_THREADS=2
--cpus-per-task==2

python2.7 Bayesian_Opt.py -dir > logfile.log
```

Use the right partition!

Be carefule not to spam your inbox, or OTHER people inboxes!

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./location/name.out.%j
### change to current work dir
#SBATCH -D ./
#SBATCH --partition=long.q
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user username@mpifr-bonn.mpg.de
#SBATCH -J 999_MH_GW
#SBATCH --ntasks=24
#  wallclock hh:mm:ss hours
#SBATCH -t 16:0:0
# virtual limit per node in MB
#SBATCH --mem=8192
source /u/caballrn/.bashrc
cd /u/kjlee/KIAA308/
# export OMP_NUM_THREADS=2
--cpus-per-task==2

python2.7 Bayesian_Opt.py -dir > logfile.log
```

Use the right partition!

Be carefule not to spam your inbox, or OTHER people inboxes!

Don’t waste memory!

```
#!/bin/bash -l
### join stdout and stderr
#SBATCH -o ./location/name.out.%j
### change to current work dir
#SBATCH -D ./
#SBATCH --partition=long.q
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user username@mpifr-bonn.mpg.de
#SBATCH -J 999_MH_GW
#SBATCH --ntasks=24
#  wallclock hh:mm:ss hours
#SBATCH -t 16:0:0
# virtual limit per node in MB
#SBATCH --mem=8192
source /u/caballrn/.bashrc
cd /u/kjlee/KIAA308/
# export OMP_NUM_THREADS=2
--cpus-per-task==2
python2.7 Bayesian_Opt.py -dir > logfile.log
```

Use the right partition!

Be carefule not to spam your inbox, or OTHER people inboxes!

Don’t waste memory!

If applicable, make use of multithread capability