Introduction to Hydra, the IFT-UAM/CSIC cluster

MANUEL TRASHORRAS, IFT/UAM-CSIC

SCHOOL OF COSMOLOGY TOOLS, MARCH 2017

Contents

So, what is HYDRA?

Not this…

So, what is HYDRA?

But this.

HYDRA basics & specifications

The Hydra HPC cluster is made of 80 dual processors of 8 or 12 cores per node.

- **batch** –> 34 nodes

  time limit: 140 h

  nodes hydra[1-34]
- **batch2** –> 18 nodes

  time limit: 140 h

  nodes hydra[35-52]
- **batch3** –> 18 nodes

  time limit: 70 h

  nodes hydra[53-70]

HYDRA basics & specifications

The Hydra HPC cluster is made of 80 dual processors of 8 or 12 cores per node.

- **batch** –> 8 cores/node

34 nodes with a dual processor Intel® Xeon® E5540 at 2,53 GHz, 24 GB DDR3 RAM & 120 GB Solid State Disk hard drive.

- **batch2** –> 12 cores/node

18 nodes with a dual processor Intel® Xeon® E5645 at 2.4 GHz, 24 GB DDR3 RAM & 120 GB Solid State Disk hard drive.

- **batch3** –> 12 cores/node

18 nodes with a dual processor Intel® Xeon® E5-2640 at 2.5 GHz, 64 GB DDR3 RAM 128 GB Solid State Disk hard drive.

HYDRA basics & specifications

System software and compilers:

- GNU compiler 5.2
- Intel compiler 16
- MKL software 16
- OpenMPI software 1.8.8

Default environment:

GNU+ autotools + OpenMPI

Software is accessible through a **module system**, that allows users to dynamically **modify the user's environment**, loading and unloading the different modules.

The following command are useful

- module avail –> shows the list of available modules
- module list –> shows the list of user's loaded modules
- module load MODULENAME –> loads a specific module
- module help –> shows help

HYDRA basics & specifications

LOADED MODULES

In /opt/ohpc/pub/moduledeps/gnu:

- openmpi/1.8.8 (L)

In /opt/ohpc/pub/modulefiles:

- autotools (L)
- ohpc (L)
- gnu/5.2.0 (L)
- prun/1.0 (L)

SOME UNLOADED MODULES

In /opt/ohpc/pub/moduledeps/gnu-openmpi:

- fftw/3.3.4

In /opt/ohpc/pub/moduledeps/gnu:

- getdist/0.2.7
- hdf5/1.8.15
- impi/5.1.1.109
- mpi4py/2.0.0
- numpy/1.9.2
- gsl/1.16
- healpy/1.10.3
- mkl/16.0.0.109
- mvapich2/2.1

In /opt/ohpc/pub/modulefiles:

- intel/16.0.0.109

Log in & change passwords in HYDRA

To log in:

```
ssh –X USERNAME@hydra.ift.uam-csic.es
```

Your home directory is:

/home/USERNAME

Small jobs can be compiled & run in hydra0 interactively.

But for larger jobs, always use the submitting system, SLURM!

Log in & change passwords in HYDRA

Your home directory is:

/home/USERNAME

To change passwords:

ssh –X USERNAME@hydra.ift.uam-csic.es

> passwd

> Type old password

> Type new password

Small jobs can be compiled & run in hydra0 interactively.

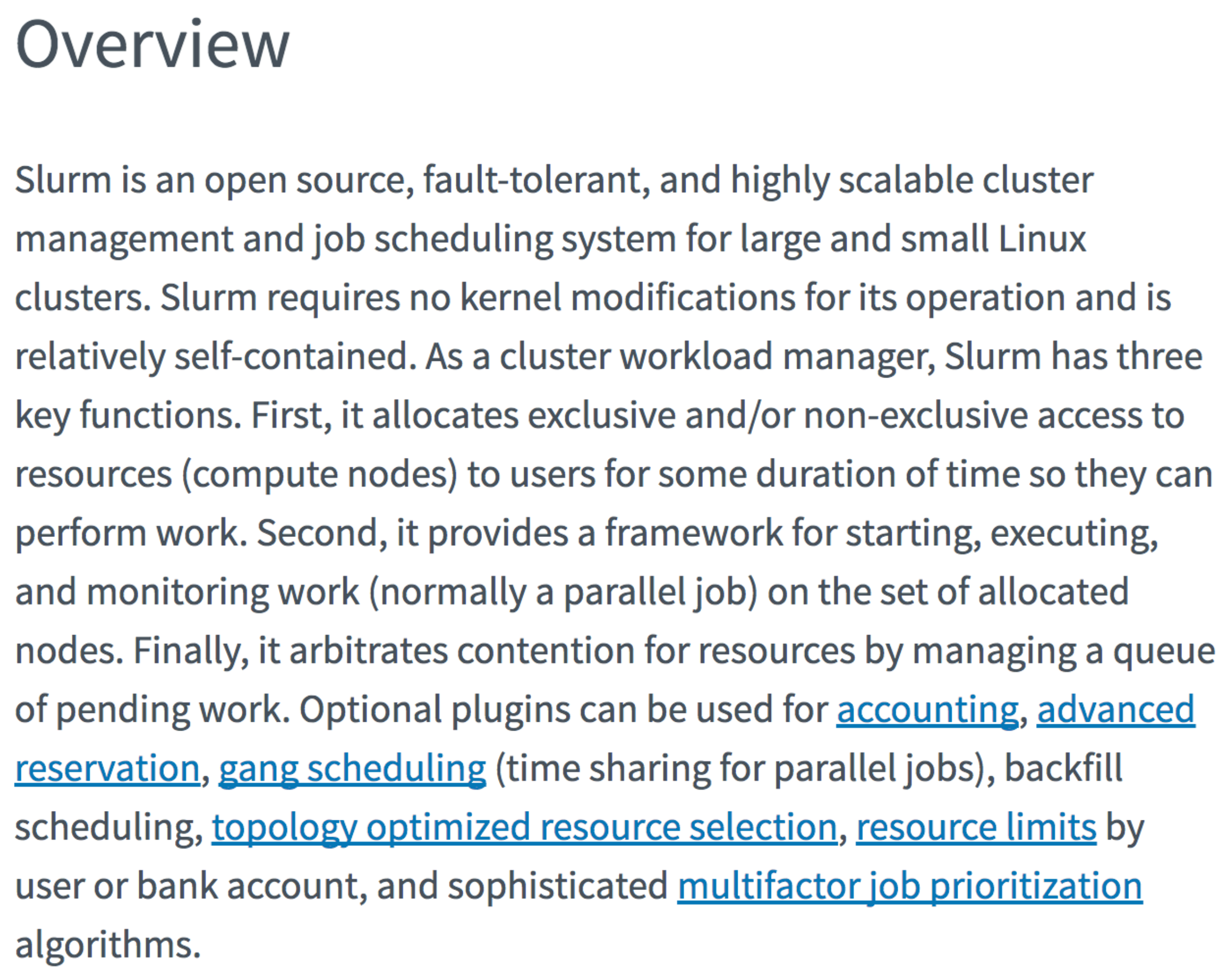
But for larger jobs, always use the submitting system, SLURM!

So, what is SLURM?

Not this...

So, what is SLURM?

But this.



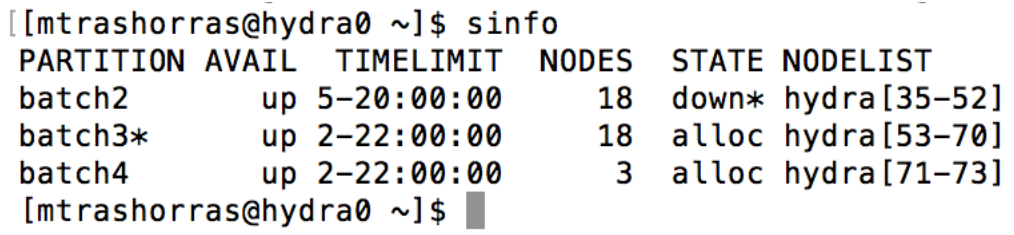
Overview

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters. Slurm requires no kernel modifications for its operation and is relatively self-contained. As a cluster workload manager, Slurm has three key functions. First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work. Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes. Finally, it arbitrates contention for resources by managing a queue of pending work. Optional plugins can be used for accounting, advanced reservation, gang scheduling (time sharing for parallel jobs), backfill scheduling, topology optimized resource selection, resource limits by user or bank account, and sophisticated multifactor job prioritization algorithms.

Source: https://slurm.schedmd.com/overview.html

Using SLURM & submitting jobs

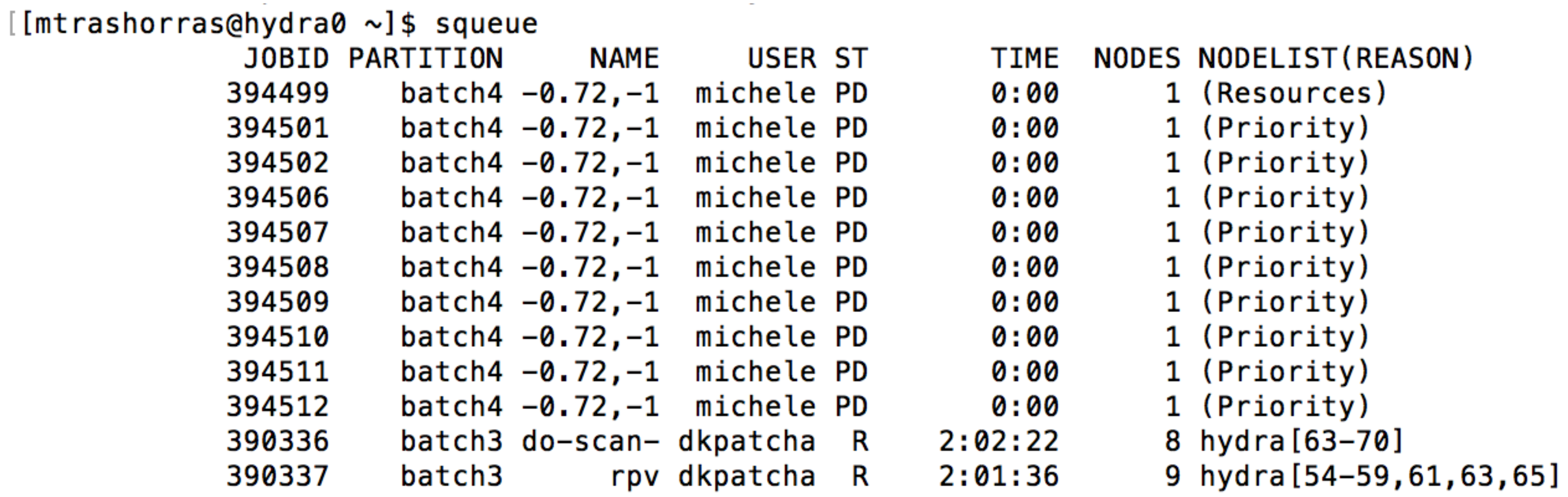
To check the status of the cluster: sinfo



```
[[mtrashorras@hydra0 ~]$ sinfo
 PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
 batch2       up 5-20:00:00     18  down* hydra[35-52]
 batch3*      up 2-22:00:00     18  alloc hydra[53-70]
 batch4       up 2-22:00:00      3  alloc hydra[71-73]
 [mtrashorras@hydra0 ~]$
```

Using SLURM & submitting jobs

To check the queues: squeue



```
[[mtrashorras@hydra0 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            394499    batch4 -0.72,-1  michele PD       0:00      1 (Resources)
            394501    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394502    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394506    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394507    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394508    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394509    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394510    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394511    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            394512    batch4 -0.72,-1  michele PD       0:00      1 (Priority)
            390336    batch3 do-scan- dkpatcha  R    2:02:22      8 hydra[63-70]
            390337    batch3      rpv dkpatcha  R    2:01:36      9 hydra[54-59,61,63,65]
```

Using SLURM & submitting jobs

To check the status of the cluster:	sinfo	
To check the queues:	squeue	
To submit a job:	sbatch JOBFILE	example: sbatch script.sh
To cancel a job:	scancel JOBID	example: scancel 123456
To check all your jobs:	squeue –u USERNAME	example: squeue –u mtrashorras
To cancel all your jobs:	scancel –u USERNAME	example: squeue –u mtrashorras
To check a specific job:	squeue –j JOBID	example: squeue –j 123456

Using SLURM & submitting jobs

EXAMPLES OF A COSMOMC JOB

cosmomc.sh

```
#!/bin/bash

#SBATCH -J JOBNAME              job name
#SBATCH -o ./stdout/job.%j.out  creates output  files
#SBATCH -e ./stdout/job.%j.err  creates error files
#SBATCH --nodes 1               total number of nodes
#SBATCH --ntasks 8              total number of cores
#SBATCH -time 140:00:00         max. running hours
#SBATCH -p batch                selects queue  “batch”

(prun) YOUR CODE HERE           execute the  code
```

cosmomc.sh

```
#!/bin/bash

#SBATCH -J JOBNAME              job name
#SBATCH -o ./stdout/job.%j.out  creates output  files
#SBATCH -e ./stdout/job.%j.err  creates error files
#SBATCH --nodes 2               total number of nodes
#SBATCH --ntasks 16             total number of cores
#SBATCH -time 140:00:00         max. running hours
#SBATCH -p batch                selects queue  “batch”

(prun) YOUR CODE HERE           execute the  code
```

Example: Compiling CosmoMC

DOWNLOADING COSMOMC

1. Go to http://cosmologist.info/cosmomc/submit.html
2. Fill in your data (name, surname and e-mail address), and submit.
3. Open the e-mail you have received, and click the link in it.
4. Download the latest version of CosmoMC: *cosmomc_Jul15_1.tar.gz*

Example: Compiling CosmoMC

DOWNLOADING THE PLANCK LIKELIHOOD

1. Go to http://pla.esac.esa.int/pla/#cosmology & click *SIGN IN* and *REGISTER*.
2. Create yourself an account. Fill in your name, surname, e-mail and submit.
3. Open the e-mail you have received, and copy the authentication code.
4. Open another mail you have received, and create yourself a password.
5. Sign in with your new account, and download:

   The likelihood code: *COM_Likelihood_Code-v2.0_R2.00.tar.bz2*

   The data files: *COM_Likelihood_Data-baseline_R2.00.tar.gz* (contains high_l, low_l, lensing)

Example: Compiling CosmoMC

COMPILING COSMOMC & THE PLANCK LIKELIHOOD

* Example of working directory: /home/USERNAME/

1. Copy the Planck likelihood code, Planck likelihood data and CosmoMC to your path.
   - COM_Likelihood_Code-v2.0_R2.00.tar.bz2
   - COM_Likelihood_Data-baseline_R2.00.tar.gz
   - cosmomc_Jul15_1.tar.gz
2. Uncompress the Planck likelihood code, Planck likelihood data and CosmoMC.
   1. cd /home/USERNAME/
   2. tar -xvf COM_Likelihood_Code-v2.0_R2.00.tar.bz2 ——> which creates a folder named plc-2.0
   3. tar -xvf COM_Likelihood_Data-baseline_R2.00.tar.gz ——> which creates a folder named plc_2.0
   4. tar -xvf cosmomc_Jul15_1.tar.gz ——> which creates a folder named cosmomc

Example: Compiling CosmoMC

COMPILING COSMOMC & THE PLANCK LIKELIHOOD

* Compiling the Plank likelihood:

3. Add this to your .bashr file or execute before you compile/run CosmoMC
   1. source /opt/ohpc/pub/compiler/intel/compilers_and_libraries_2016.0.109/linux/mpi/intel64/bin/mpivars.sh
   2. source /opt/ohpc/pub/compiler/intel/compilers_and_libraries_2016.0.109/linux/bin/compilervars.sh intel64
4. Install the Planck likelihood
   1. cd /home/USERNAME/plc-2.0
   2. ./waf configure --install_all_deps --lapack_mkl_version=10.3 --lapack_mkl=/opt/ohpc/pub/compiler/intel/compilers_and_libraries_2016.0.109/linux/mkl./waf install
5. Put a symbolic link to the Planck data in CosmoMC data:
   1. cd /
   2. ln -s /home/USERNAME/plc_2.0 /home/USERNAME/cosmomc/data/clik

Example: Compiling CosmoMC

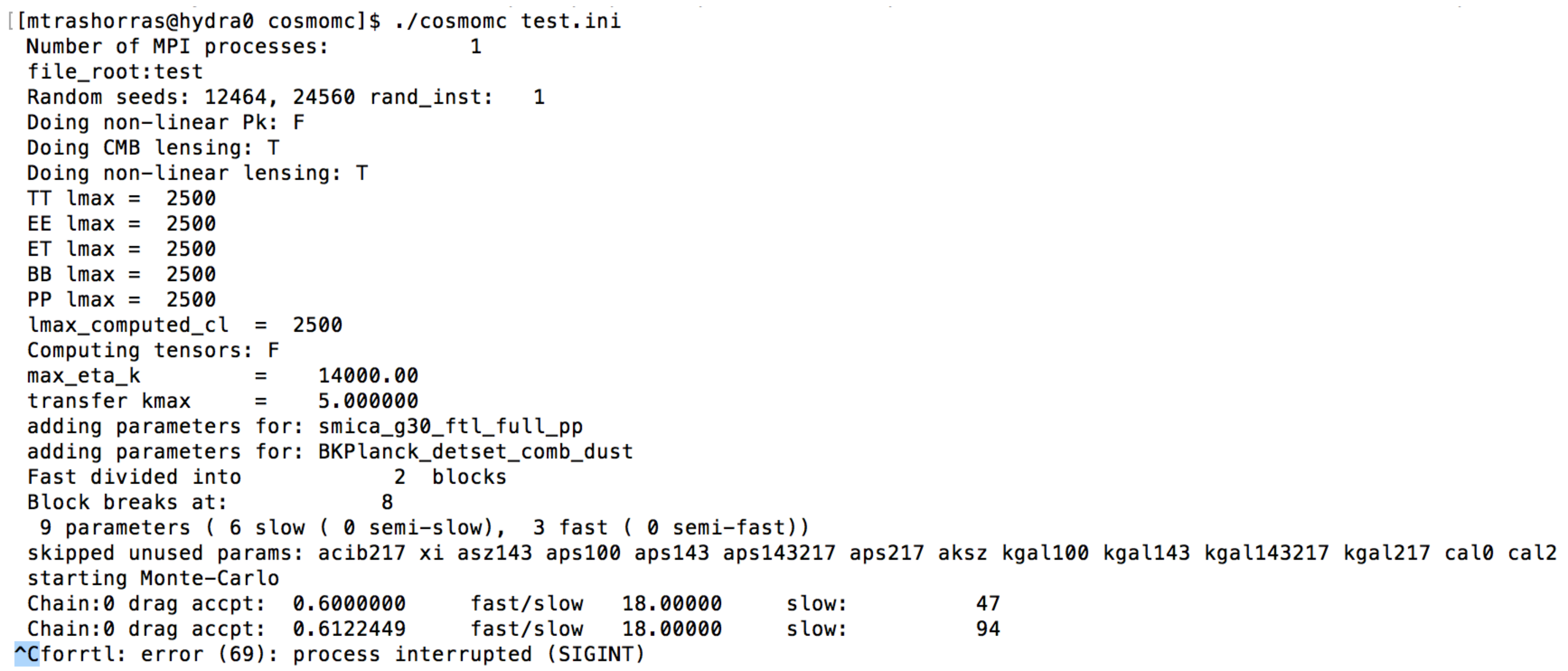
COMPILING COSMOMC & THE PLANCK LIKELIHOOD

* Compiling CosmoMC:

6. Modify the CosmoMC Makefile
   3. cd /home/USERNAME/cosmomc/source/
   4. vi Makefile replace in line 7: MPIF90C ?= mpif90 ——> MPIF90C ?= mpif90 -f90=ifort
7. Compile CosmoMC
   3. make clean
   4. make
8. Run a toy chain (leave it running for a minute, this is not a parallel run). Done! 😎
   3. cd ..
   4. vi test.ini replace in line 37: action = 0 (test) ——> action = 4 (full MCMC)
   5. ./cosmomc test.ini

Example: Compiling CosmoMC

COMPILING COSMOMC & THE PLANCK LIKELIHOOD



```
[mtrashorras@hydra0 cosmomc]$ ./cosmomc test.ini
 Number of MPI processes:           1
 file_root:test
 Random seeds: 12464, 24560 rand_inst:    1
 Doing non-linear Pk: F
 Doing CMB lensing: T
 Doing non-linear lensing: T
 TT lmax =  2500
 EE lmax =  2500
 ET lmax =  2500
 BB lmax =  2500
 PP lmax =  2500
 lmax_computed_cl  =  2500
 Computing tensors: F
 max_eta_k         =    14000.00
 transfer kmax     =    5.000000
 adding parameters for: smica_g30_ftl_full_pp
 adding parameters for: BKPlanck_detset_comb_dust
 Fast divided into             2  blocks
 Block breaks at:            8
  9 parameters ( 6 slow ( 0 semi-slow),  3 fast ( 0 semi-fast))
 skipped unused params: acib217 xi asz143 aps100 aps143 aps143217 aps217 aksz kgal100 kgal143 kgal143217 kgal217 cal0 cal2
 starting Monte-Carlo
 Chain:0 drag accpt:  0.6000000     fast/slow   18.00000     slow:           47
 Chain:0 drag accpt:  0.6122449     fast/slow   18.00000     slow:           94
^Cforrtl: error (69): process interrupted (SIGINT)
```

Example: Sending a CosmoMC job

EXAMPLES OF A COSMOMC JOB

cosmomc.sh

```
#!/bin/bash

#SBATCH -J cosmomc_test          job name
#SBATCH -o ./stdout/job.%j.out   creates output  files
#SBATCH -e ./stdout/job.%j.err   creates error files
#SBATCH --nodes 1                total number of nodes
#SBATCH --ntasks 8               total number of cores
#SBATCH -time 140:00:00          max. running hours
#SBATCH -p batch                 selects queue  "batch"

./cosmomc test.ini               execute the  code
```

cosmomc.sh

```
#!/bin/bash

#SBATCH -J cosmomc_test          job name
#SBATCH -o ./stdout/job.%j.out   creates output  files
#SBATCH -e ./stdout/job.%j.err   creates error files
#SBATCH --nodes 2                total number of nodes
#SBATCH --ntasks 16              total number of cores
#SBATCH -time 140:00:00          max. running hours
#SBATCH -p batch                 selects queue  "batch"

./cosmomc test.ini               execute the  code
```

Example: Sending a CosmoMC job

EXAMPLE OF A PARALELL JOB

cosmomc.sh

```
#!/bin/bash

#SBATCH -J cosmomc_test        job name
#SBATCH -o ./stdout/job.%j.out creates output files
#SBATCH -e ./stdout/job.%j.err creates error files
#SBATCH --nodes 1              total number of nodes
#SBATCH --ntasks 8             total number of cores
#SBATCH -time 140:00:00        max. running hours
#SBATCH -p batch               selects queue "batch"
export $OMP_NUM_THREADS=8

prun–n 1 ./cosmomc test.ini    execute the code
```

cosmomc.sh

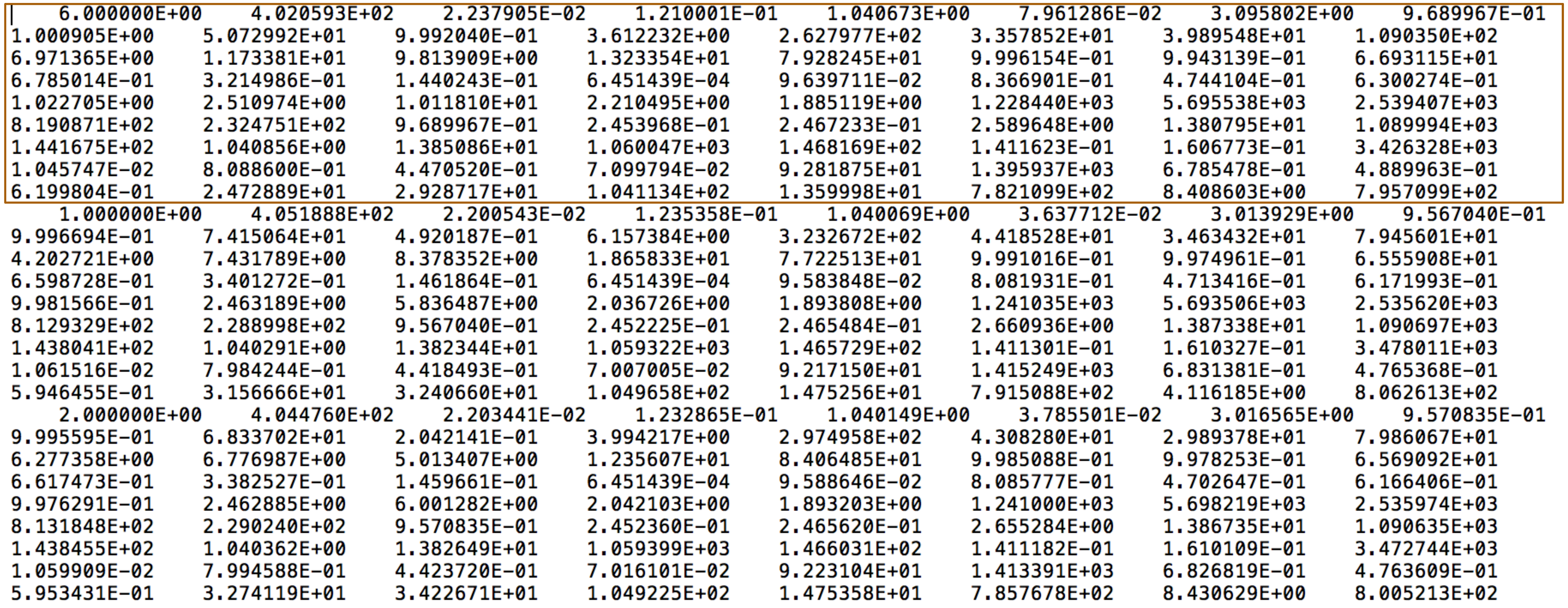
```
#!/bin/bash

#SBATCH -J cosmomc_test        job name
#SBATCH -o ./stdout/job.%j.out creates output files
#SBATCH -e ./stdout/job.%j.err creates error files
#SBATCH --nodes 2              total number of nodes
#SBATCH --ntasks 16            total number of cores
#SBATCH -time 140:00:00        max. running hours
#SBATCH -p batch               selects queue "batch"
export $OMP_NUM_THREADS=16

prun2 ./cosmomc test.ini       execute the code
```

Example: CosmoMC output

Output files: **test_1.txt**, test_2.txt, …, test.inputparams, test.ranges, test.paramnames, test.likelihoods.



```
    6.000000E+00    4.020593E+02    2.237905E-02    1.210001E-01    1.040673E+00    7.961286E-02    3.095802E+00    9.689967E-01
1.000905E+00    5.072992E+01    9.992040E-01    3.612232E+00    2.627977E+02    3.357852E+01    3.989548E+01    1.090350E+02
6.971365E+00    1.173381E+01    9.813909E+00    1.323354E+01    7.928245E+01    9.996154E-01    9.943139E-01    6.693115E+01
6.785014E-01    3.214986E-01    1.440243E-01    6.451439E-04    9.639711E-02    8.366901E-01    4.744104E-01    6.300274E-01
1.022705E+00    2.510974E+00    1.011810E+01    2.210495E+00    1.885119E+00    1.228440E+03    5.695538E+03    2.539407E+03
8.190871E+02    2.324751E+02    9.689967E-01    2.453968E-01    2.467233E-01    2.589648E+00    1.380795E+01    1.089994E+03
1.441675E+02    1.040856E+00    1.385086E+01    1.060047E+03    1.468169E+02    1.411623E-01    1.606773E-01    3.426328E+03
1.045747E-02    8.088600E-01    4.470520E-01    7.099794E-02    9.281875E+01    1.395937E+03    6.785478E-01    4.889963E-01
6.199804E-01    2.472889E+01    2.928717E+01    1.041134E+02    1.359998E+01    7.821099E+02    8.408603E+00    7.957099E+02
    1.000000E+00    4.051888E+02    2.200543E-02    1.235358E-01    1.040069E+00    3.637712E-02    3.013929E+00    9.567040E-01
9.996694E-01    7.415064E+01    4.920187E-01    6.157384E+00    3.232672E+02    4.418528E+01    3.463432E+01    7.945601E+01
4.202721E+00    7.431789E+00    8.378352E+00    1.865833E+01    7.722513E+01    9.991016E-01    9.974961E-01    6.555908E+01
6.598728E-01    3.401272E-01    1.461864E-01    6.451439E-04    9.583848E-02    8.081931E-01    4.713416E-01    6.171993E-01
9.981566E-01    2.463189E+00    5.836487E+00    2.036726E+00    1.893808E+00    1.241035E+03    5.693506E+03    2.535620E+03
8.129329E+02    2.288998E+02    9.567040E-01    2.452225E-01    2.465484E-01    2.660936E+00    1.387338E+01    1.090697E+03
1.438041E+02    1.040291E+00    1.382344E+01    1.059322E+03    1.465729E+02    1.411301E-01    1.610327E-01    3.478011E+03
1.061516E-02    7.984244E-01    4.418493E-01    7.007005E-02    9.217150E+01    1.415249E+03    6.831381E-01    4.765368E-01
5.946455E-01    3.156666E+01    3.240660E+01    1.049658E+02    1.475256E+01    7.915088E+02    4.116185E+00    8.062613E+02
    2.000000E+00    4.044760E+02    2.203441E-02    1.232865E-01    1.040149E+00    3.785501E-02    3.016565E+00    9.570835E-01
9.995595E-01    6.833702E+01    2.042141E-01    3.994217E+00    2.974958E+02    4.308280E+01    2.989378E+01    7.986067E+01
6.277358E+00    6.776987E+00    5.013407E+00    1.235607E+01    8.406485E+01    9.985088E-01    9.978253E-01    6.569092E+01
6.617473E-01    3.382527E-01    1.459661E-01    6.451439E-04    9.588646E-02    8.085777E-01    4.702647E-01    6.166406E-01
9.976291E-01    2.462885E+00    6.001282E+00    2.042103E+00    1.893203E+00    1.241000E+03    5.698219E+03    2.535974E+03
8.131848E+02    2.290240E+02    9.570835E-01    2.452360E-01    2.465620E-01    2.655284E+00    1.386735E+01    1.090635E+03
1.438455E+02    1.040362E+00    1.382649E+01    1.059399E+03    1.466031E+02    1.411182E-01    1.610109E-01    3.472744E+03
1.059909E-02    7.994588E-01    4.423720E-01    7.016101E-02    9.223104E+01    1.413391E+03    6.826819E-01    4.763609E-01
5.953431E-01    3.274119E+01    3.422671E+01    1.049225E+02    1.475358E+01    7.857678E+02    8.430629E+00    8.005213E+02
```

Useful links

SLURM Guide:	https://slurm.schedmd.com/quickstart.html
Modules Guide:	http://modules.sourceforge.net/
Run jobs interactively:	www.ift.uam-csic.es/hydra/quickstart_uHydra_interactive.pdf
Run jobs in paralell:	www.ift.uam-csic.es/hydra/quickstart_uHydra_batch.pdf
CosmoCoffee wiki:	http://cosmocoffee.info/
CosmoMC Readme:	http://cosmologist.info/cosmomc/readme.html
Planck Readme:	http://cosmologist.info/cosmomc/readme_planck.html
Python Readme:	http://cosmologist.info/cosmomc/readme_python.html
GetDist GUI Readme :	http://cosmologist.info/cosmomc/readme_gui.html