

Working on Ada

Table of Content

Working on Ada	1
1. IDRIS users' manual	2
2. Commands to manage jobs on ada	2
3. Example of a job to start an executable in MPI	2
4. Information on Ergon files from Adapp	3
5. Job Header for MPI - MPI/OMP with libGCM	3
5.1. Forced model	3
5.1.1. MPI	3
5.1.2. hybrid MPI-OMP	3
5.2. Coupled model	4
5.2.1. MPI	4
5.2.2. hybrid MPI-OMP	4
6. Specificities libGCM on Ada	5
7. Specificities for Adapp	6
7.1. IDRIS users' manual for adapp	7
7.2. Header for adapp job	7

1. IDRIS users' manual

- See: <http://www.idris.fr/eng/ada/> for ada : computing server
- See: <http://www.idris.fr/eng/adapp/> for adapp : pre-post-treatment
- See: <http://www.idris.fr/eng/ergon/> for ergon : file server

2. Commands to manage jobs on ada

- The job's time limit is measured in real time, for example 1 hour on 32 procs accounts for 32 hours. Be careful not to have too much time on 1 processor.
- `llsubmit -->` submit a job
- `llcancel -->` cancel a job
- `llq -u login -->` indicates all jobs in the queue or running for the login *login*
- Trick: parameterize the `llq` display to see the job names

```
llq -u $(whoami) -f %jn %id %st %c %dq %h -W
```

- Post-mortem : `idrjar` , `idrjar -l -j #jobid#`, to obtain detailed information: memory, real time, efficiency,...
- Example of `idrjar` output :

```
ada > idrjar
|-----|
|--- IDRIS/CNRS. Version du 18 mars 2015 ---|
|-----|

Sorties concernant l'identifiant rpslxxx pour la période du
==> 01 juin 2013 au 19 juin 2013

Owner   Job Name      JobId      Queue tEse  tCpu  #T  (%)  S
-----
rpslxxx ADA337      ada338.290170.0 c32t2  133   1232  32  28.95 C
rpslxxx ADA337      ada338.290333.0 c32t2  5425  165141  32  95.13 C
rpslxxx PACKDEBUG  ada338.290610.0 t2      11     2    1  18.18 C
rpslxxx ADA337      ada338.290438.0 c32t2  5471  166878  32  95.32 C
rpslxxx PACKRESTART  ada338.290611.0 t2      182    25    1  13.74 C
rpslxxx REBUILDWRK  ada338.290612.0 t2     1577   503    1  31.90 C
rpslxxx PACKOUTPUT  ada338.290730.0 t2     114    43    1  37.72 C
```

3. Example of a job to start an executable in MPI

Here is an example of a simple job to start an executable `orchidee_ol` (or `gcm.e` commented). The input files and the executable must be in the directory before starting the executable.

```
#!/bin/ksh
# #####
# ##  ADA IDRIS  ##
# #####
# Query's name
# @ job_name = test
# Job type
# @ job_type = parallel
# Standard output file
# @ output = Script_Output_test.$(jobid)
# Error output file (the same)
# @ error = Script_Output_test.$(jobid)
# Number of requested processes
```

```

#@ total_tasks = 8
# max. CPU time per MPI process hh:mm:ss
#@ wall_clock_limit = 1:00:00
# Number of task OpenMP/threads per MPI process
### @ parallel_threads = 4
# End of header
#@ queue

poe ./orchidee_ol
#poe ./gcm.e

```

4. Information on Ergon files from Adapp

Ergon files are visible from Adapp. Use \$ARCHIVE to reach Ergon files on Adapp. \$ARCHIVE is /arch/home/rech/lab/plabxxx on Adapp. All Unix command are available on Adapp to provides information on Ergon files.

5. Job Header for MPI - MPI/OMP with libGCM

5.1. Forced model

5.1.1. MPI

To launch a job on XXX MPI tasks, you need to use libGCM/ins_job script. Check your header. It should be :

```

#!/bin/ksh
# #####
# ##  ADA IDRIS  ##
# #####
# Job name
#@ job_name = MyJob
# Job type
#@ job_type = parallel
# Standard output file name
#@ output = Script_Output_MyJob.000001
# Error output file name
#@ error = Script_Output_MyJob.000001
# Total number of tasks
#@ total_tasks = XXX
#@ environment = "BATCH_NUM_PROC_TOT=XXX"
# Maximum CPU time per task hh:mm:ss
#@ wall_clock_limit = 1:00:00
# End of the header options
#@ queue

```

5.1.2. hybrid MPI-OMP

Hybrid version are only available with _v6 configurations

To launch a job on XXX MPI tasks and YYY threads OMP on each task

- first you need to modify your config.card

```
ATM= (gcm.e, lmdz.x, XXXMPI, YYYOMP)
```

second you need to use libGCM/ins_job script. Check your header. It should be :

```

#!/bin/ksh
# #####
# ##  ADA IDRIS  ##

```

```

# #####
# Job name
# @ job_name = MyJob
# Job type
# @ job_type = parallel
# Standard output file name
# @ output = Script_Output_MyJob.000001
# Error output file name
# @ error = Script_Output_MyJob.000001
# Total number of tasks
# @ total_tasks = XXX
# @ environment = "BATCH_NUM_PROC_TOT=XXX*YYY"
# Maximum CPU time per task hh:mm:ss
# @ wall_clock_limit = 1:00:00
# Specific option for OpenMP parallelization: Number of OpenMP threads per MPI task
# @ parallel_threads = YYY
# End of the header options
# @ queue

```

5.2. Coupled model

5.2.1. MPI

To launch a job on XXX (32) MPI tasks. 5 for NEMO, 1 for oasis and 26 MPI tasks for LMDZ by default for IPSLCM5A, you need to use libIGCM/ins_job script. Check your header. It should be :

```

#!/bin/ksh
# #####
# ## ADA IDRIS ##
# #####
# Job name
# @ job_name = MyCoupledJob
# Job type
# @ job_type = parallel
# Standard output file name
# @ output = Script_Output_MyCoupledJob.000001
# Error output file name
# @ error = Script_Output_MyCoupledJob.000001
# Total number of tasks
# @ total_tasks = 32
# @ environment = "BATCH_NUM_PROC_TOT=32"
# Maximum CPU time per task hh:mm:ss
# @ wall_clock_limit = 1:00:00
# End of the header options
# @ queue

```

5.2.2. hybrid MPI-OMP

Hybrid version are only available with _v6 configurations

To launch a job on XXX (47) MPI tasks and YYY (8) threads OMP for LMDZ, ZZZ (180) MPI tasks for NEMO and SSS (1) XIOS servers :

- first you need to modify your config.card. On ada, this is working for IPSLCM6 and _v6 configurations :

```

ATM= (gcm.e, lmdz.x, 47MPI, 8OMP)
SRF= ( " , " )
SBG= ( " , " )
OCE= (opa, opa.xx , 180MPI)
ICE= ( " , " )
MBG= ( " , " )

```

```
CPL= ("", "" )
IOS= (xios_server.exe, xios.x, 1MPI)
```

second you need to use `libIGCM/ins_job -m Intel` script. Check your header. It should be :

```
#!/bin/ksh
# #####
# ## ADA      IDRIS ##
# #####
# Job name
# @ job_name = MyCoupledJob
# Standard output file name
# @ output = Script_Output_MyCoupledJob.000001
# Error output file name
# @ error = Script_Output_MyCoupledJob.000001
# Job type
# @ job_type = mpich
# Total number of tasks
# @ node = 18
# Specific option for OpenMP parallelization: Number of OpenMP threads per MPI task
# Memory : as_limit=3.5gb max per process per core. With 4 threads per process use max as_limit=14gb
# Maximum CPU time per task hh:mm:ss
# @ wall_clock_limit = 1:00:00
# @ environment = "BATCH_NUM_PROC_TOT=228" ; wall_clock_limit=$(wall_clock_limit)
# End of the header options
# @ queue
```

Note : an authorization is required. Please ask `assist_at_idris.fr`.

Note : Intel environment 2016.2 is forced by `libIGCM` since `libIGCM v2.8` during execution

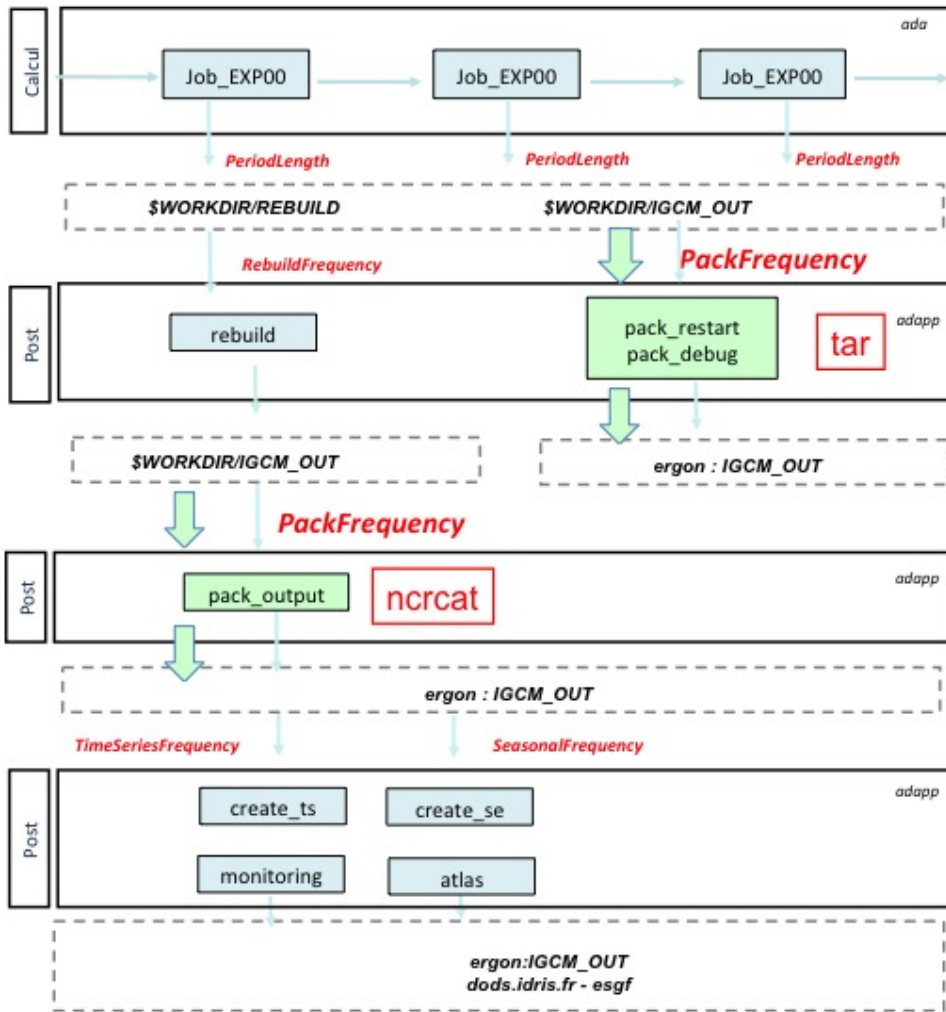
Note : this is working with all compiler.

6. Specificities `libIGCM` on Ada

At IDRIS and for Ada, output files are 'packed' using `libIGCM_v2`, i.e. they are grouped by periods (in general 1 year) using the command `tar` or `ncrcat` for NetCDF output files.

This option implies that files must be temporarily stored on the `$WORKDIR` space, which means that a large storage is needed (at least 20 To).

The diagram below details all jobs including `pack_debug`, `pack_restart` and `pack_output` as well as the directories those jobs are using. Note that the files are temporarily stored in the `$WORKDIR/IGCM_OUT` directories before being grouped and sent on Ergon in the `IGCM_OUT` directories.



You will obtain annual output files with 12 monthly values in the Output/MO directory if you put `PeriodLength=1M` and `PackFrequency=1Y` in `config.card`. This is the default grouping period of most configurations but you can of course change it.

What you must remember:

- The tool [RunChecker.job](#) is meant to help you monitoring your simulations. It offers a synthetic view of the different post processing jobs' status.
- The tool [clean_latestPackperiod.job](#) is meant to help you clean until the last successfully computed pack period.
- If you detect anomalies and must rerun part of the simulation, you will have to make new complete pack periods (e.g. filling a gap by running 1 month of simulation is out of the question).
- The restart files are stored and grouped on Ergon in the directory `IGCM_OUT/.../RESTART`
- The different output text-files are stored and grouped on Ergon in the directory `IGCM_OUT/.../DEBUG`
- The listings for pack-jobs outputs stay on Ada in the directory `$WORKDIR/IGCM_OUT/.../Out`
- If you put the `SpaceName=TEST` parameter in `config.card` the pack jobs will not be started and your simulation will be stored in the `WORKDIR/IGCM_OUT` directory. This can be very useful for short tests.

To learn more about this Section, you can read the documentation on [Simulation and post-processing](#) and on [Monitor, debug and relaunching](#).

Finally, in case of panic, visit us or send your questions to the list platform-users.

7. Specificities for Adapp

- Adapp is dedicated to pre and post-treatment.
- Note that Ergon files are visible in read only mode through `$ARCHIVE`.
you can use `idrls` to know the status of a file stored on ergon. See `idrls -?. m` means migrated on tape only, `-` means on disk.

```

cd $ARCHIVE
idrls IGCM/RESTART/IPSLCM6/DEVT/piControl/01T03V14/*/Restart/*
M ACCESS      L USER      GROUP      SIZE      MOD_DATE   ACC_DATE   EXP_DATE   FILE_NAME
= ===== = ===== = ===== = ===== = ===== = ===== = =====
- -rwxrwxr-x 1 rpslxxx   psl       218188352 09.06.2015 22.01.2016 22.01.2017 IGCM/RESTART/IPSLCM6/DEVT/piControl/01T03
m -rwxrwxr-x 1 rpslxxx   psl       1411362796 09.06.2015 22.01.2016 22.01.2017 IGCM/RESTART/IPSLCM6/DEVT/piControl/01T03

```

- Use largely Adapp for analyses and interactive work
- Adapp is free of charge

7.1. IDRIS users' manual for adapp

- See: <http://www.idris.fr/eng/adapp/> for adapp : pre-post-treatment

7.2. Header for adapp job

A post-treatment jobs includes these header lines :

```

# @ job_type = serial
# @ requirements = (Feature == "prepost")

```