

Working on the Irene machine

Table of Content

Working on the Irene machine	1
1. Introduction	2
2. Job manager commands	2
3. Suggested environment	2
4. File system	3
5. How the storage project is set by libIGCM	4
6. A figure to understand file systems used on Irene	4
7. Configurations	4
8. How to use old configurations	4

Last update 05/07/2018

1. Introduction

- On-line users manual: <https://www-tgcc.ccc.cea.fr/docs/irene.info.pdf> (you will need a TGCC login and password)
- Irene computing nodes: the nodes of partition **skylake** have 48 cores each, which is 3 times more than the computing nodes from the standard partition of Curie.
 - Skylake nodes for regular computation (coming from irene.info)
 - Partition name: skylake
 - CPUs: 2x24-cores Intel Skylake@2.7GHz (AVX512)
 - [Cores/Node?](#): 48
 - Nodes: 1 656
 - Total cores: 79488
 - RAM/Node: 192GB
 - RAM/Core: 4GB
- When submitting a job through `ccc_msub` or `ccc_mprun`, you must specify `-m work`, `-m scratch`, `-m store`, or combine them like in `-m work,scratch`. This constraint has the advantage that your jobs won't be suspended if a file system you don't need becomes unavailable. This is done in all jobs in libIGCM.
- Compute nodes are diskless, meaning that `/tmp` is not hosted on a local hard drive anymore, but on system memory instead. It offers up to 16 GB (compared to 64 GB on Curie). Please note that any data written to it reduces the size of the memory that remains available for computations. For libIGCM post-treatment jobs you might need to use more cores than used at curie or, preferably, use xlarge nodes.
- The default time limit for a job submission is 2hours (7200s) contrary to 24h (86400s) on curie
- Irene post-processing nodes : xlarge are free and usefull for post-processing operations. Since 2018/08/16, it's possible to use them in libIGCM.
 - Fat nodes for computation requiring a lot of shared memory (coming from irene.info)
 - Partition name: xlarge
 - CPUs: 4x28-cores Intel Skylake@2.1GHz
 - GPUs: 1x Nvidia Pascal P100
 - [Cores/Node?](#): 112
 - Nodes: 5
 - Total cores: 560
 - RAM/Node: 3TB
 - RAM/Core: 5.3GB
 - IO: 2 HDD de 1 TB + 1 SSD 1600 GB/NVMe

2. Job manager commands

- `ccc_msub job ->` submit a job
- `ccc_mdel ID ->` kill the job with the specified ID number
- `ccc_mstat -u login ->` display all jobs submitted by login, add `-f` to see full job name
- `ccc_mpp ->` display all jobs submitted on the machine. `ccc_mpp -n` to avoid colors.
- `ccc_mpp -u $(whoami) ->`display your jobs.

3. Suggested environment

Before working on Irene you need to prepare your environment. This is important to do before compilation to ensure the use of same modules as done by libIGCM running environment. We propose you 2 files which you can copy from the home igcmg. The first one called **bashrc** will source the second called **bashrc_irene**. Copy both files to your home, rename them by adding a dot as prefix. You can add personal settings in your `.bashrc_irene`. Do as follow:

```
cp /ccc/cont003/home/igcmg/igcmg/MachineEnvironment/irene/bashrc ~/.bashrc
cp /ccc/cont003/home/igcmg/igcmg/MachineEnvironment/irene/bashrc_irene ~/.bashrc_irene
```

The `.bashrc` will source your own `.bashrc_irene` which must be in your home.

After re-connexion or source of `.bashrc_irene`, check your loaded modules for intel, netcdf, mpi, hdf5 needed for the compilation:

```

module list
Currently Loaded Modulefiles:
 1) ccc                                10) mkl/17.0.6.256(default)          19) flavor/hdf5/parallel
 2) datadir/own(default)              11) flavor/buildcompiler/intel/17(default) 20) netcdf-c/4.6.0(default)
 3) dflldatadir/own(default)          12) intel/17.0.6.256(default)          21) netcdf-fortran/4.4.4(default)
 4) licdrv/intel                      13) hwloc/1.11.3(default)            22) hdf5/1.8.20(default)
 5) c++/intel/17.0.6.256(default)     14) feature/openmpi/mpi_compiler/intel(default) 23) feature/bridge/heterogenous_mpm
 6) c/intel/17.0.6.256(default)       15) feature/openmpi/net/mxm(default)    24) nco/4.6.0(default)
 7) fortran/intel/17.0.6.256(default) 16) .tuning/openmpi/2.0(default)       25) cdo/1.7.2rc6(default)
 8) feature/mkl/lp64                  17) flavor/buildmpi/openmpi/2.0       26) ghostscript/9.19(default)
 9) feature/mkl/sequential            18) mpi/openmpi/2.0.2                27) ferret/7.2(default)

```

The modules are specified in the file `/ccc/cont003/home/igcmg/igcmg/MachineEnvironment/irene/env_irene` which is sourced in `bashrc_irene`. The same file `env_irene` is sourced in `libIGCM`.

--> Be careful this environment can be update during next weeks according to TGCC recommendations

Create `~/forward` file in your main home containing only one line with your email address to receive emails from `libIGCM`.

4. File system

You have a main home where you arrive when connecting to irene, called "home de connexion" by the TGCC. You also have a home, a `storedir`, a `workdir`, a `scratchdir` by project. For example if you are working with project `gen2201` and `gen2212` you will have all following directories:

```

/ccc/cont003/home/***/login           # connexion home, where ***=your lab (lsce, ips1, etc..)

/ccc/cont003/home/gen2201/login
/ccc/cont003/home/gen2212/login

/ccc/store/cont003/gen2201/login
/ccc/store/cont003/gen2212/login

/ccc/work/cont003/gen2201/login       # use it to store sources
/ccc/work/cont003/gen2212/login

/ccc/scratch/cont003/gen2201/login
/ccc/scratch/cont003/gen2212/login

```

IMPORTANT : Check that you have read and write access to above directories (for your projects). Contact TGCC hotline if it is not the case.

On the SCRATCH space any files that stays 60 days without being read or modified will be purged(deleted), as well as any directory that remains empty for 30 days.

After connexion to irene, load your project environment as default using the module `dflldatadir`. For example if you will work on the project `gen2201`, do following (we suggest you to add the command into your `.bashrc_irene`):

```
module switch dflldatadir dflldatadir/gen2201
```

By changing the `dflldatadir`, the variables `$CCCHOME`, `$CCWORKDIR`, `$CCCSTOREDIR` and `$CCCSCRATCHDIR` point to the corresponding project directories. `$HOME` is always the main connexion home.

You will also have new environment variables to access working directories :

```

GEN2201_ALL_CCCSCRATCHDIR=/ccc/scratch/cont003/gen2201/gen2201
GEN2201_CCCWORKDIR=/ccc/work/cont003/gen2201/login
GEN2201_ALL_HOME=/ccc/cont003/home/gen2201/gen2201
GEN2201_CCCSTOREDIR=/ccc/store/cont003/gen2201/login
GEN2201_CCCSCRATCHDIR=/ccc/scratch/cont003/gen2201/login
GEN2201_ALL_CCCWORKDIR=/ccc/work/cont003/gen2201/gen2201
GEN2201_HOME=/ccc/cont003/home/gen2201/login

```

```
GEN2201_ALL_CCCSTOREDIR=/ccc/store/cont003/gen2201/gen2201
```

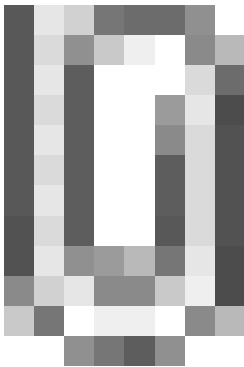
If you previously worked at curie and your directories were in /cont003/dsm/login you will now find your data in a specific new project file system "dsmipsl". We recommend to move your data in your genci project file system. The TGCC hotline can help you if you want.

5. How the storage project is set by libIGCM

When you use libIGCM it is recommended to dedicate one modipsl/libIGCM to one project allocation. By default, the output folders IGCM_OUT will be created in the directories \$CCCSCRATCHDIR, \$CCCWORKDIR and \$CCCSTOREDIR corresponding to the project used in the main job. It is important that the same project is used in the post-processing jobs in libIGCM.

If you need to use another project for the computing than the storage, it is possible to set the variable DataProject in config.card UserChoices section, for example DataProject=gen6328, read more [here](#). This project will be used for all output directories for the computing job and post-processing jobs even if they have another project for computing in the headers. The variable DataProject can also be used if you work with different project allocations in the same modipsl. Only exception (harmless) is the first RUN_DIR folder which is always created in the \$CCCSCRATCHDIR corresponding to the dfldatadir loaded in when submitting main job. When the job resubmits itself, the RUN_DIR will be in the same project space as the rest of the output.

6. A figure to understand file systems used on Irene



7. Configurations

Following configurations are technically ready (scientific validation in progress) to be used at Irene. On 2018/7/2, performances are slow.

- IPSLCM6.1.5-LR
- IPSLCM5A2.1
- LMDZOR_v6.1.5
- ORCHIDEE_trunk
- ORCHIDEE_2_0
- NEMO_v6_OMIP
- LMDZORINCA_v6
- LMDZ5A2.1_ISO
- IPSLES6
- IPSLCM5A2CHT

8. How to use old configurations

- you need to update files AA_make.gdef and w_h_i in util directory

```
cd modipsl/util
svn update AA_make.gdef
svn update w_i_h
```

--> if you receive an svn error with this specific command, extract a new modipsl and copy these two files from it

- you need to change your version of libIGCM

```
cd modipsl
mv libIGCM libIGCM_curie
svn co http://forge.ipsl.jussieu.fr/libigcm/svn/trunk/libIGCM
```

- you need to download arch files for IRENE for LMDZ, ORCHIDEE, INCA

```
cd modipsl
mkdir modele_arch
cd modele_arch
svn co svn://forge.ipsl.jussieu.fr/orchidee/trunk/ORCHIDEE/arch/ ORCHIDEE_arch
svn co http://forge.ipsl.jussieu.fr/inca/svn/trunk/INCA5/arch/ INCA_arch
svn co http://svn.lmd.jussieu.fr/LMDZ/LMDZ6/branches/IPSLCM6.0.15/arch/ LMDZ_arch
svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/branches/xios-2.5/arch/ XIOS_arch
mv ORCHIDEE_arch/*IRENE* ../modeles/ORCHIDEE/arch/.
mv INCA_arch/*IRENE* ../modeles/INCA/arch/.
mv LMDZ_arch/*IRENE* ../modeles/LMDZ/arch/.
mv XIOS_arch/*IRENE* ../modeles/XIOS/arch/.
```

- if you are working with INCA you need a file pre_proc_X64_IRENE.x

```
cd modipsl/modeles/INCA
cp pre_proc_X64_CURIE.x pre_proc_X64_IRENE.x
```

- create a new makefile

```
cd modipsl/config/**
mv Makefile Makefile_curie
../../util/ins_make
gmake clean
```

- create a new experiment directory
- now you can work as on Curie