Wikiprint Book

Title: 1. Introduction

Subject: Igcmg_doc - Doc/ComputingCenters/TGCC/Irene

Version: 68

Date: 12/04/21 12:04:54

Table of Content

Working on the Irene machine

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.
  - 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 is 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 2 hours (7 200 s).
- Irene post-processing nodes : xlarge are free and useful for post-processing operations. Since 2018/10/17 and libIGCM 1470, xlarge is proposed as default post-processing node.
  - 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

3.1. General 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 ~igcmg/MachineEnvironment/irene/bashrc ~/.bashrc
cp ~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) dfldatadir/own(default)             12) intel/17.0.6.256(default)                  21) netcdf-fortran/4.4.4(default)
4) licsrv/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 ~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.

3.2. Subversion version

Since only recent subversion version (i.e > 1.6) are installed on Irene supercomputer, some usual functionalities are not available anymore (ex : svn command on copy of subdirectories...). In order to keep these functionalities, subversion 1.6.9 has been installed. To use this version :

```
module unload subversion
module use ~igcmg/Modules/tools
module load subversion/1.6.9
irene190 : svn --version

svn, version 1.6.9 (r901367)

 compiled Oct 31 2018, 11:12:49
```

The use of this version has been added in default environment ~igcmg/MachineEnvironment/irene/bashrc_irene. Beware of coherence of subversion version you use from the extraction of your model/configuration to the use of svn commands in directories of your model/configuration.

4. 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.

For gencmip6 project, you have to set a subproject for computing and gencmip6 is forced for all directories.

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. # Example of ins_job

```
> ./ins_job
...
Wait for the next question ...
Hit Enter or give project ID (default is gen0000), possible projects are gen1111 gen2222 ... or other xxxcmip6 : aaacmip6
gen0000 (RETURN)
```

```
ProjectID is gen0000 at Irene
Hit Enter or give TYPE OF NODE required for post-processing (default is "xlarge"), possible types of nodes are "skylake" o
(RETURN)
ProjectNode for post-processing is xlarge at Irene
Hit Enter or give NUMBER OF CORES required for post-processing (default is "8")
possible numbers of cores are "1" to "112" for xlarge :
(RETURN)
ProjectCore for post-processing is 8
Wait for the next question ...
Hit Enter or give project ID (default is gen0000), possible projects are gen1111 gen2222 ... or other xxxcmip6 : aaacmip6
(RETURN)
PostID is gen0000 at Irene on xlarge for post-processing
```

5. File systems used on Irene

A figure to illustrate Irene filesystems is available here

6. Irene job headers

Here is an example of a job header as generated by libIGCM on the Irene machine:

```
######################
## IRENE   TGCC/CEA ##
######################
#MSUB -r MY-SIMULATION
#MSUB -o Script_Output_MY-SIMULATION.000001
#MSUB -e Script_Output_MY-SIMULATION.000001
#MSUB -eo
#MSUB -n 976
#MSUB -x
#MSUB -T 86400
#MSUB -A dekcmip6
#MSUB -q skylake
#MSUB -m store,work,scratch
```

The detail is as follows:

Control	Keyword	Argument	Example	Comments
Job name	`-r`	string	`#MSUB -r Job_MY-SIMULATION`	The string should not contain the underscore (_) character.
Standard output file name	`-o`	string	`#MSUB -o Script_Output_MY-SIMULATION.000001`	
Error output file name	`-e`	string	`#MSUB -e Script_Output_MY-SIMULATION.000001`	If both `-o` and `-e` names are the same, the two outputs will be merged. The `-eo` option of the example is as a matter of fact redundant with that.
Number of MPI tasks allocated	`-n`	integer	`#MSUB -n 976`	This is for the SPMD case; in the MPMD case, the integer is the number of cores.
Node exclusivity	`-x`	none	`#MSUB -x`	
Wall-time (maximum time allowed for execution)	`-T`	integer	`#MSUB -T 86400`	Time is expressed in number of seconds.

Project allocation	`-A`	string	`#MSUB -A dekcmip6`	
Partition used	`-q`	string	`#MSUB -q skylake`	Choice is amongst `skylake` or `xlarge` or `hybrid`.
Visible spaces	`-m`	(comma-separated) string(s)	`#MSUB -m store,work,scratch`	Specific to TGCC. The job can only "see" spaces specified on this line. Possible spaces are amongst `store` and/or `work` and/or `scratch`.
Job priority	`-U`	string	`#MSUB -U high`	Possible arguments are `high`, `medium` or `low`. (This command does not appear by default on jobs generated by libIGCM.)