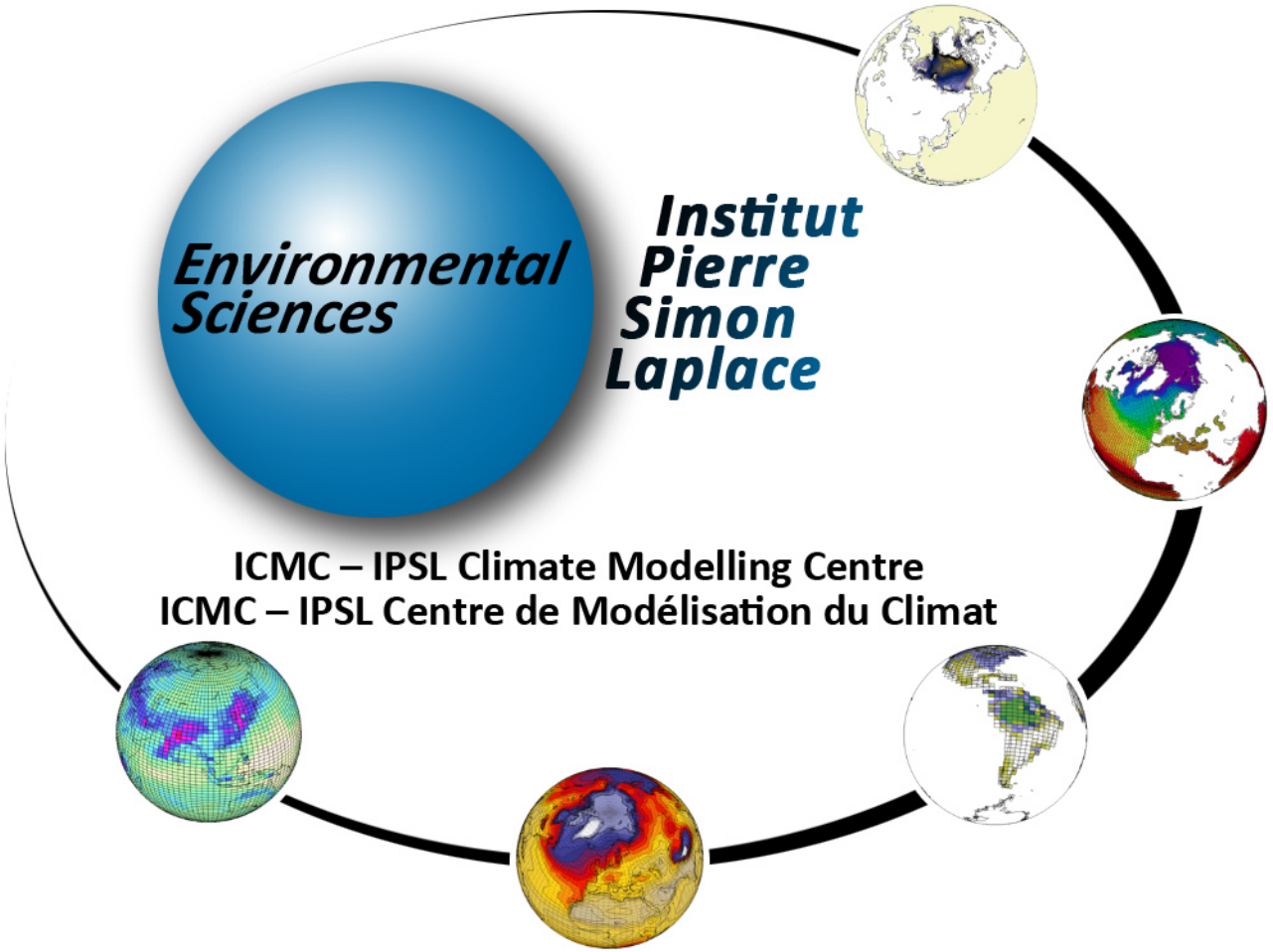


Training session day 1

Introduction to modipsl and libIGCM tools

November 27th 2018, IDRIS

IPSL « Plate-forme » group – Lola Falletti & Nicolas Lebas

Outline

1. **Introduction**
2. HPC context
3. Which supercomputer(s) for us ?
4. modipsl : install and compile
5. LibIGCM : create a simulation
6. LibIGCM : launch and follow a simulation

IPSL - Institut Pierre Simon Laplace

IPSL gathers 9 laboratories whose research topics concern the global environment.

CEREA / GEOPS / LERMA / LATMOS / LISA / LMD /
LOCEAN / LSCE / METIS

IPSL Climate Modeling Centre (ICMC http://icmc.ipsl.fr/)

Activities articulated around

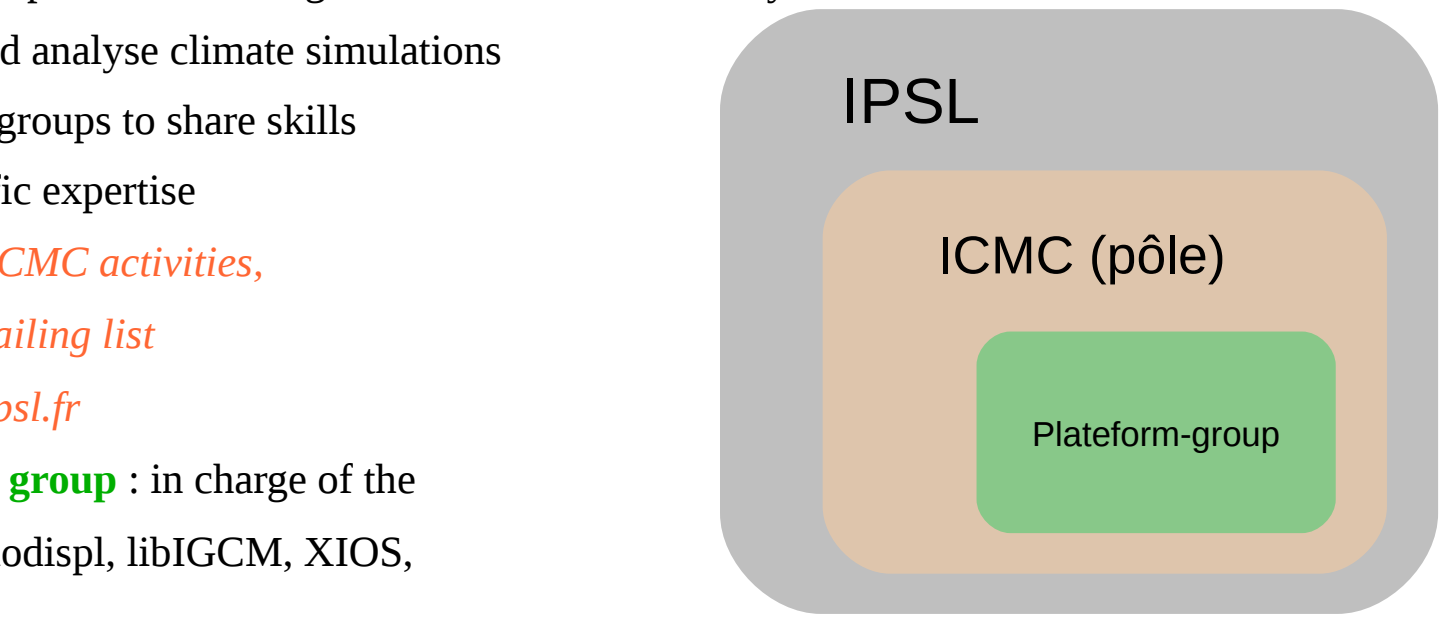
- The development of an integrated model of the Earth system
- To run and analyse climate simulations
- Working groups to share skills
- A scientific expertise

To be involved in ICMC activities,

subscribe to the mailing list

ipsl_cmc@listes.ipsl.fr

IPSL Plate-forme group : in charge of the development of modispl, libIGCM, XIOS, metrics tools



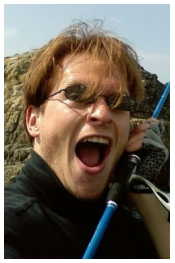
Plateform-group members

Arnaud Caubel

Anne Cozic

Sébastien Denvil

Christian Ethé



Jérôme Servonnat

Laurent Fairhead

Marie-Alice Foujols

Josefine Ghattas

Nicolas Lebas

Yann Meurdesoif

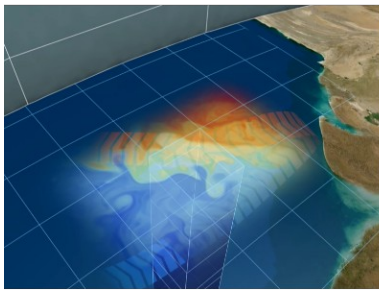
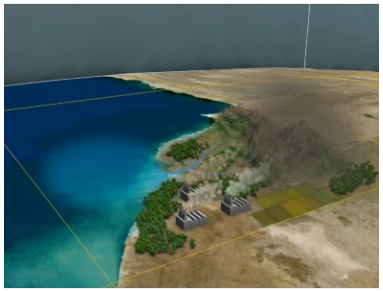
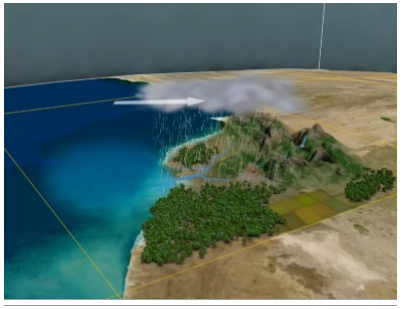
Olivier Marti

Olivier Boucher

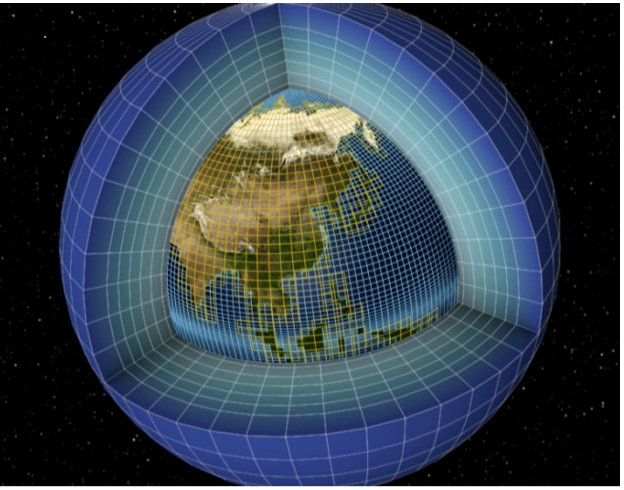
Thibaut Lurton

Lola Falletti

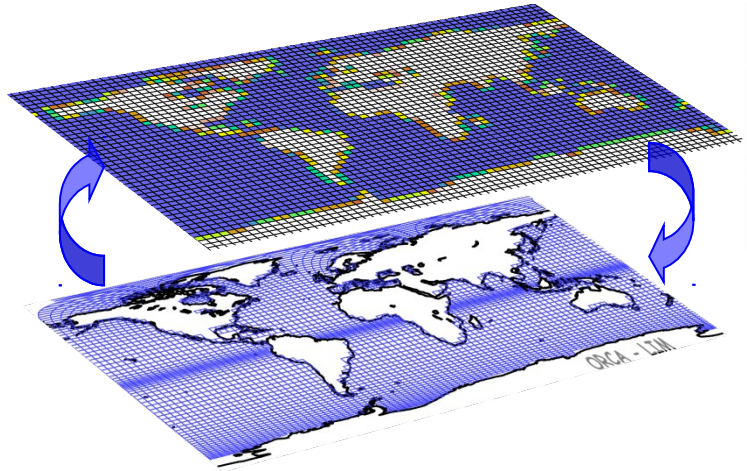
IPSL Earth System Model

INCA/REPROBUS (atmospheric chemistry, aerosols) ⟺ **LMDZ** (atmosphere) ⟺ **ORCHIDEE** (land surfaces, vegetation)



OASIS3-MCT (coupler)

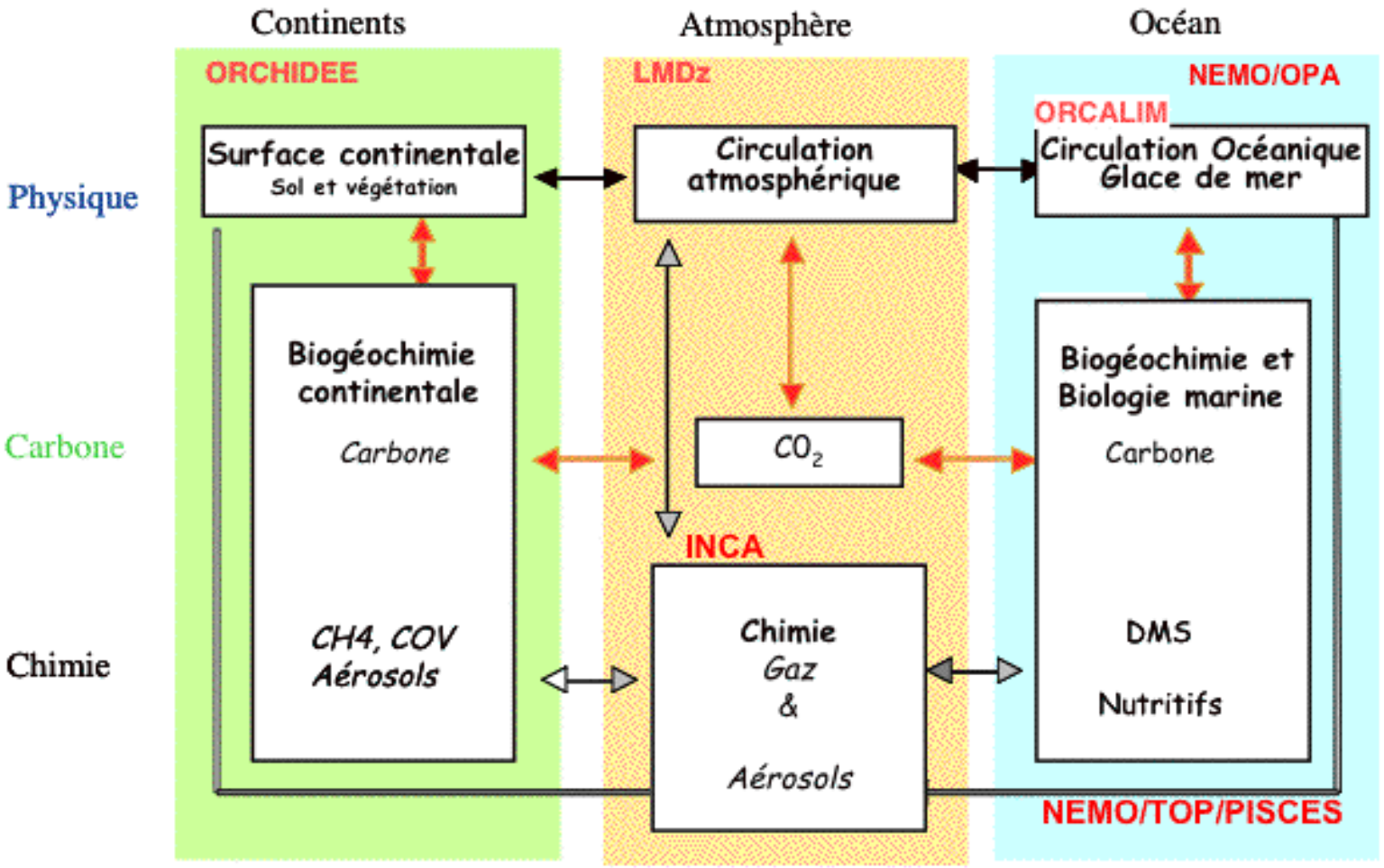


OPA (ocean)

PISCES (marine biogeochemistry) — **LIM** (sea ice)

NEMO

IPSL Earth System Model



IPSLCM history

The IPSL Climate Model (IPSL-CM) is developed since 1995.

IPSL-CM1 (1995-1997) was the first version of the IPSL Climate Model and were mainly used for paleo-climate studies.

IPSL-CM2 (1998-2001) includes a thermodynamic sea ice model, has been used to analyse the response of the climate to changes in insolation, and participates to the CMIP-2 and PMIP-1 model inter-comparison projects. It was also coupled to a carbon cycle model and has been used to perform the first climate simulations (present and future) with an interactive carbon cycle.

Modipsl (*2000*)

IPSL-CM3 (2002) includes the new LMDZ atmospheric model, with a Tiedtke scheme for the atmospheric convection, and was used for the first long term 1000-year simulation with the IPSL coupled model.

IPSL-CM4 (2003-2009) includes the new LMDZ-4 atmospheric model with a Emanuel convective scheme and a new cloud scheme, the ORCHIDEE land surface model, the new ORCA2 oceanic model with the LIM sea-ice model, all these models being coupled with the OASIS coupler. IPSL-CM4 participates to the CMIP3 and PMIP-II projects that give important inputs to the IPCC fourth assessment report (IPCC-AR4). A version of this model, IPSL-CM4-LOOP, also includes the carbon cycle and contributes to the C4MIP project.

libIGCM *(28 may 2008 first commit on svn)*

The **IPSL-CM5** (2010-) model is the last version of the IPSL model and is a full earth system model. Based on a physical atmosphere-land-ocean-sea ice model, it also includes a representation of the carbon cycle, the stratospheric chemistry and the tropospheric chemistry with aerosols. There are two versions of this model, with two different sets of physical models: the IPSL-CM5A is a direct extension of IPSL-CM4 whereas the IPSL-CM5B has an atmospheric model with very different physical parametrisations. The IPSL-CM5 model participates to the CMIP5 and the PMIP3 projects.

The **IPSL-CM6** (2016-) is the actual version of the IPSL model and is a full earth system model.

IPSL Compile and run environment

Software infrastucture based on **modipsl** and **libIGCM** which allows to :

modipsl

- **predefine and extract** standard configurations
- **compile** sources from different components, coupling interfaces

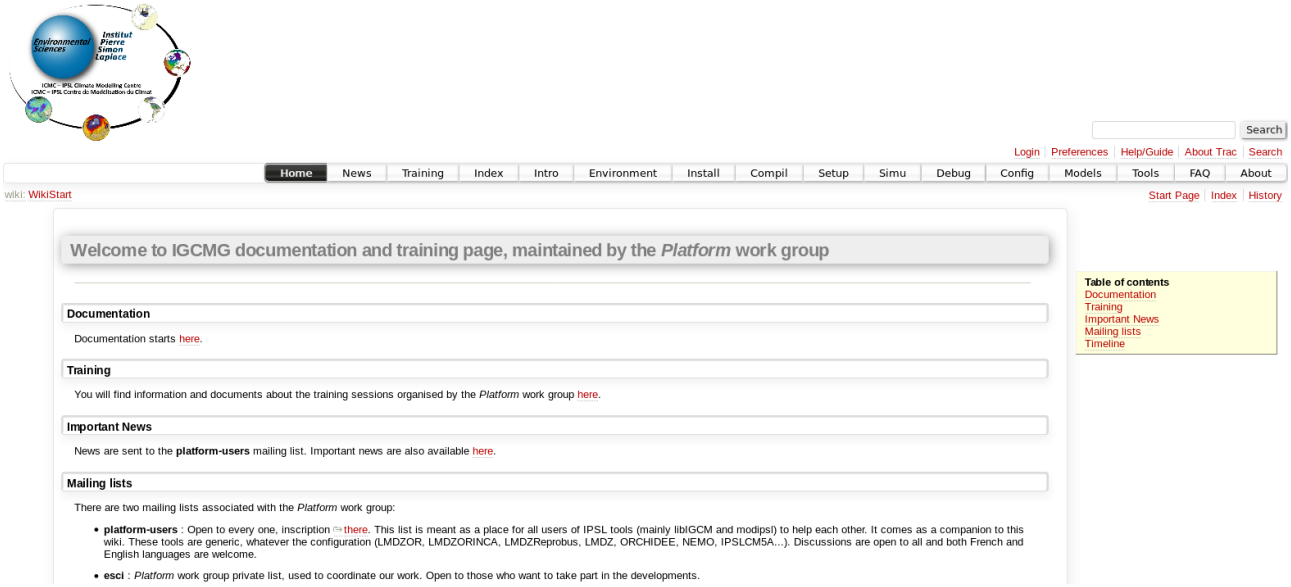
libIGCM

- **adapt and launch** predefined experiments
- **monitor** simulations
- **produce** and **store** results from models
- **produce**, **store** and **distribute** some analysis

Tools available for usage at TGCC, IDRIS, LSCE and IPSL cluster

Web documentations

- Modipsl / libIGCM : http://forge.ipsl.jussieu.fr/igcmg_doc



- LMDZ : http://lmdz.lmd.jussieu.fr
- ORCHIDEE : http://labex.ipsl.fr/orchidee/
- NEMO : http://www.nemo-ocean.eu
- INCA : http://forge.ipsl.jussieu.fr/inca

e-mail list

platform-users@ipsl.jussieu.fr

List for communication between users.

All user's can ask questions and answer his/her collegues questions.

→ ***All users need to subscribe***

Subscription on this page : https://listes.ipsl.fr/sympa/info/platform-users

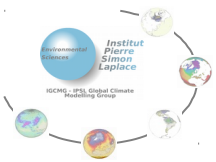
Training

Training courses at IPSL :

- Training course 1 : modipsl and libIGCM (this training cours)
- Training course 2 : advanced training course on IPSL running environment (tomorrow)
- LMDZ training course (contact Marie-Pierre.Lefebvre@lmd.jussieu.fr) next session in december
- ORCHIDEE Introduction 2-days course (contact orchidee-help@ipsl.jussieu.fr) next session after tomorrow

Other suggested training :

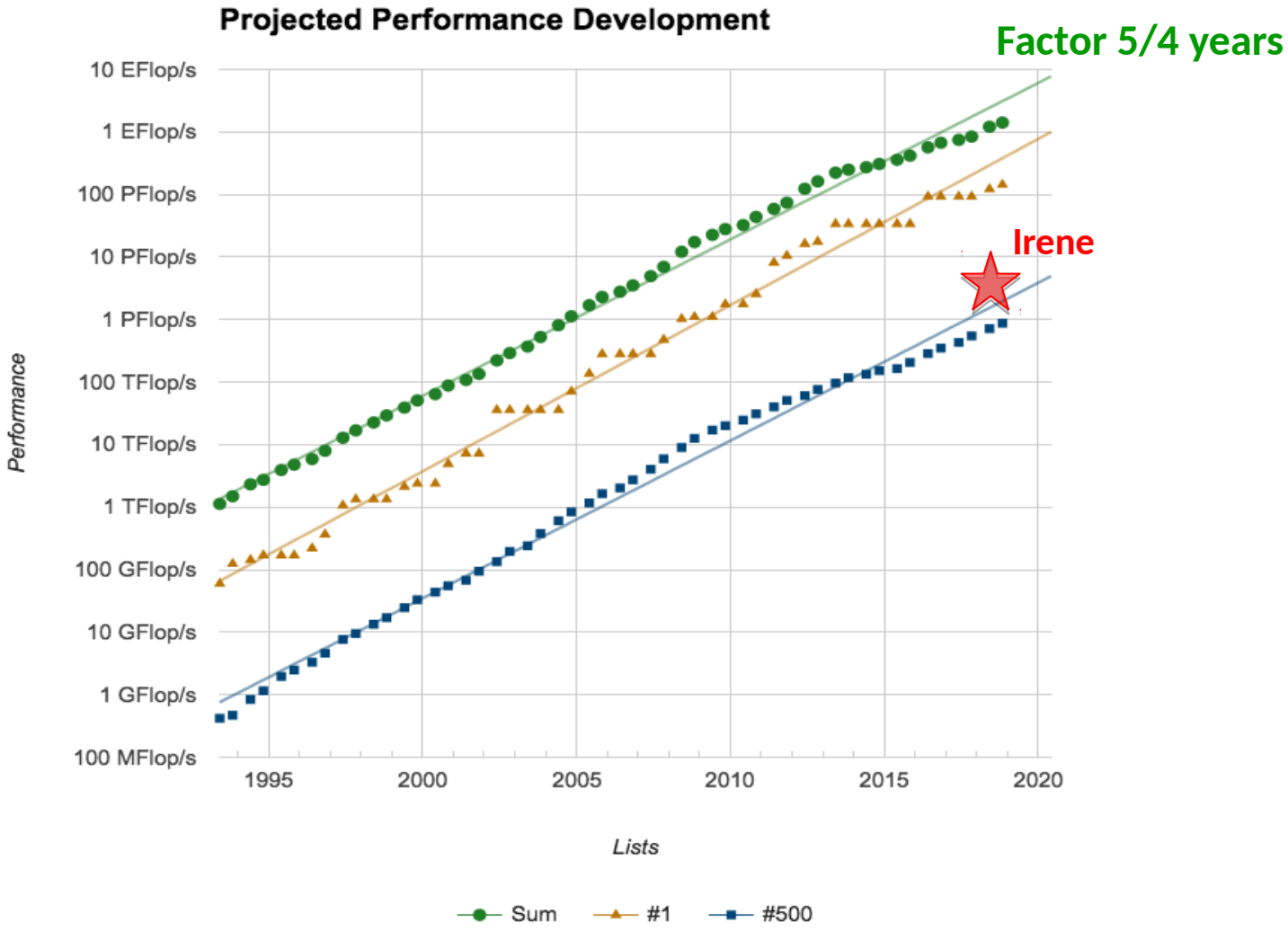
- Fortran (niv1, niv2) and MPI at IDRIS twice a year www.idris.fr
- Training course for using the computer centres
- UNIX course
- http://formation-calcul.fr/ → give an inventory of training course (numeric – calcul – hpc) in France



Outline

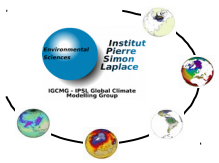
1. Introduction
2. **High Performance Computing context**
3. Which supercomputer(s) for us?
4. modipsl : install and compile
5. LibIGCM : create a simulation
6. LibIGCM : launch and follow a simulation

Supercomputers timeline: top 500



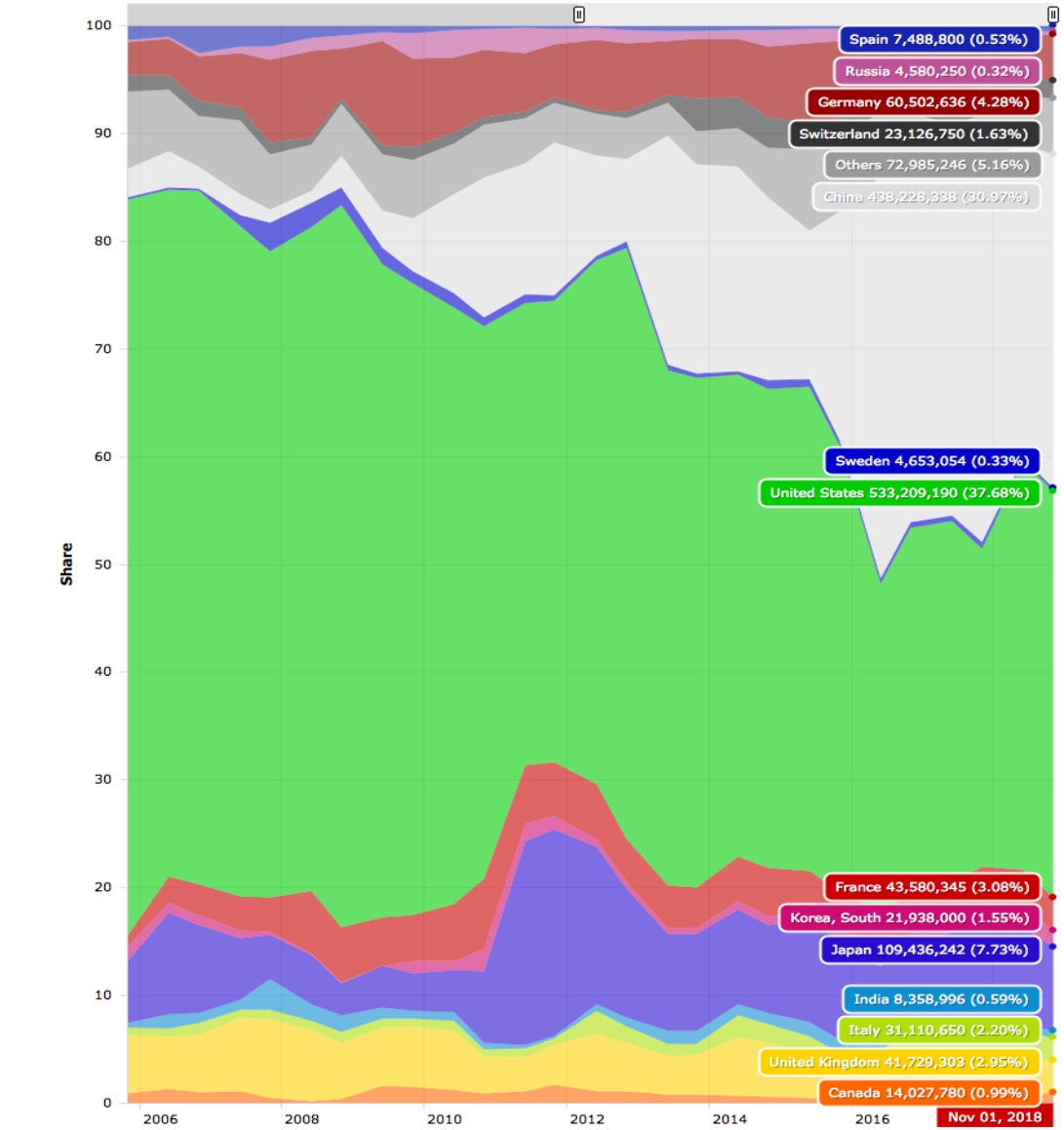
Supercomputers (countries)

Rank	Site	System	Cores	Rmax (TFlop/s)	Power (kW)
1	**United States**	**IBM**	**2,397,824**	**143,500**	**9,783**
2	**United States**	IBM/NVIDIA	1,572,480	94,640.0	7,438
3	**China**	NRCPC	10,649,600	93,014.6	15,371
4	**China**	NUDT	4,981,760	61,444.5	18,482
5	**Switzerland**	Cray	387,872	21,230	2,384
6	**United States**	Cray	979,072	20,158.7	7,578
7	**Japan**	Fujitsu	391,680	19,880.0	1,649
8	**Germany**	Lenovo	305,856	19,476.6	-
9	**United States**	Cray	560,640	17,590.0	8,209
10	**United States**	IBM	1,572,864	17,173.2	7,890



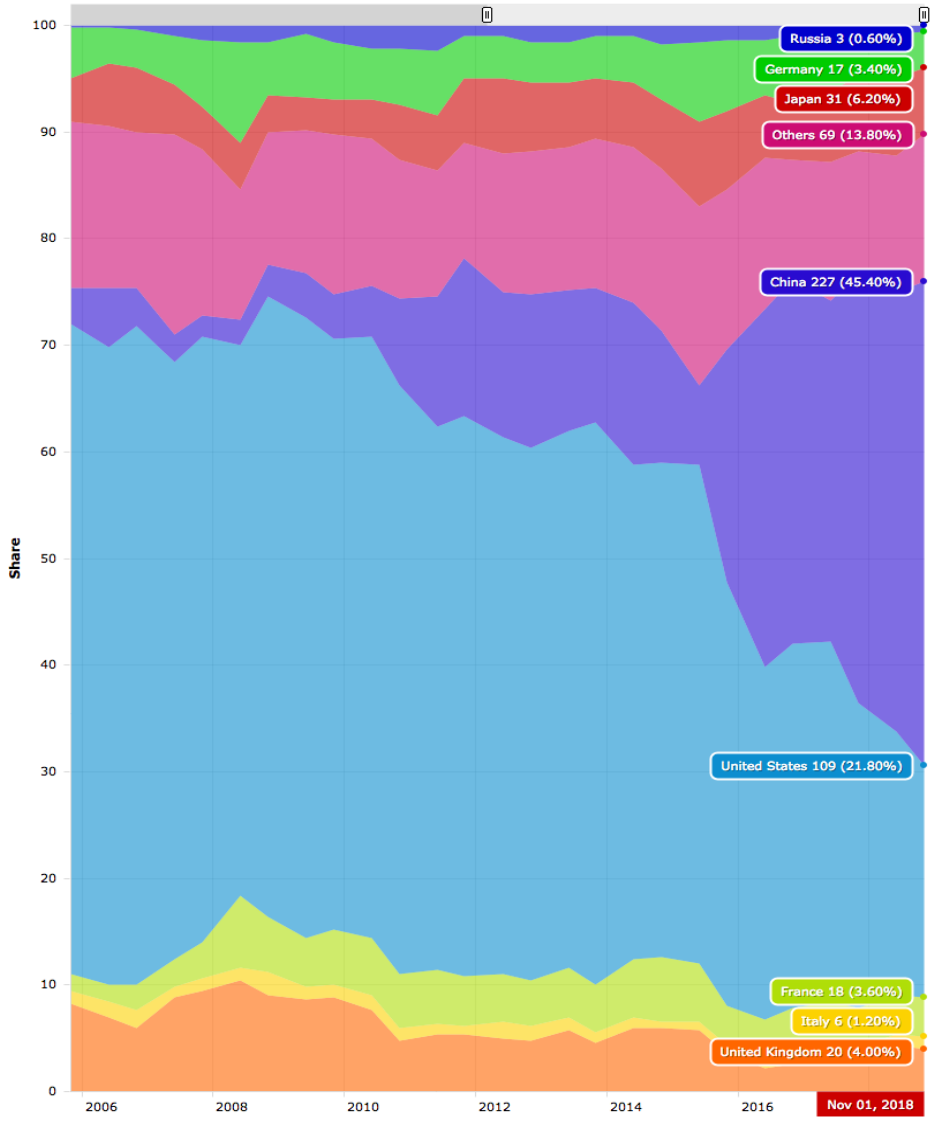
Supercomputers (ranking)

Rank	Site	System	Cores	Rmax (TFlop/s)
1	**DOE/SC/Oak Ridge National Laboratory United States**	**Summit** IBM / NVIDIA	**2,397,824**	**143,500**
2	DOE/NNSA/LLNL United States	**Sierra** IBM / NVIDIA	1,579,480	94,640
3	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** Sunway MPP	10,649,600	93,014
4	National Super Computer Center in Guangzhou China	**Tianhe-2A** TH-IVB-FEP (Intel)	4,981,760	61,444
5	Swiss National Supercomputing Centre Switzerland	**Piz Daint** Cray	387,872	21,230



Canada	14,027,780	United Kingdom	41,729,303	Italy	31,110,650	India	8,358,996
Japan	109,436,242	Korea, South	21,938,000	France	43,580,345	United States	533,209,190
Sweden	4,653,054	China	438,228,338	Others	72,985,246	Switzerland	23,126,750
Germany	60,502,636	Russia	4,580,250	Spain	7,488,800		

Cumulated performances by country over time



Number of HPC systems by country over time

United Kingdom	20	Italy	6	France	18	United States	109
China	227	Others	69	Japan	31	Germany	17
Russia	3						

Supercomputers (what about France?)

Rank	Site	System	Cores	Rmax (TFlop/s)
1	**DOE/SC/Oak Ridge National Laboratory United States**	**Summit** IBM / NVIDIA	**2,397,824**	**143,500**
…	…	…	…	…
16	CEA France	**Tera-1000-2** Bull Sequana	561,408	11,965
50	Total Exploration Production	**Pangea** SGI ICE X, Xeon	220,800	5,283
64	**CEA/TGCC-GENCI**	**JOLIOT-CURIE SKL** Bull Sequana	**79,488**	**4,065**
74	**CINES/GENCI**	**Occigen2** Bullx	**85,824**	**2,494**
99	Meteo-France	**Prolix2** Bullx	72,000	2,168

Outline

1. Introduction
2. High Performance Computing context
3. **Which supercomputer(s) for us?**
4. modipsl : install and compile
5. LibIGCM : create a simulation
6. LibIGCM : launch and follow a simulation

HPC performances

System factors

- Architecture : CPU (SMP, MPP), accelerators (GPU, MIC)
- Memory hierarchy (register - cache - main memory - disk)
- I/O configuration- Parallel file system (supporting parallel I/O)
- Compiler
- Connecting network between processors

Application factors

- Programming language (C/C++, Fortran, CUDA, ...)
- Algorithms and implementation
- Memory management
- Libraries (e.g. math libraries)
- Compiler optimization flags
- Use of I/O
- MPI / OpenMP

HPC performances

System factors

- Architecture : CPU (SMP, MPP), accelerators (GPU, MIC)
- Memory hierarchy (register - cache - main memory - disk)
- I/O configuration- Parallel file system (supporting parallel I/O)
- Compiler
- Connecting network between processors

Application factors

- Programming language (C/C++, Fortran, CUDA, ...)
- Algorithms and implementation
- Memory management
- Libraries (e.g. math libraries)
- Compiler optimization flags
- Use of I/O
- MPI / OpenMP

Improvments in IPSL models

Two computing centers

Computing **Ada (10 624 cores, 233 Tflops)** : 332 nodes, 4 proc Intel Sandy Bridge 8-cores 2,7 GHz (32 cores/node), 128Go/nodes (4Go/core)

Turing **(**65 536 cores, 836 Tflops) : 4.096 nœuds de calcul, PowerPC A2 (16 cores/node), 16 Go/node (1G/core)

Post Ada : 4 nœuds 4 proc Intel Westmere 8-cœurs à 2,67GHz (32cores/node), 1 To (32 Go/core)
Fichiers Ergon, dods : http://prodn.idris.fr/thredds...
Assistance assist@idris.fr, 01-69-35-85-55
Infos www.idris.fr

Computing **Irene Joliot-Curie SKL (79 488 cores, 4,1 Pflops)** 1 656 Intel Skylake 8168 bi-processors nodes - 2,7 GHz, 24 cores/proc. 192 GB of DDR4 memory / node.

Post Irene xlarge
Fichiers dods : http://esgf.extra.cea.fr/thredds...
Assistance , 01-77-57-42-42
Infos irene.info

TGCC

- Filesystems
  - *HOME : small space, back up*
  - *WORKDIR : working space and archiving of small files – quota 1Tb, no back up, no purge*
  - *STOREDIR : only for archive of big files – min 1Gb – quota 100 000 inodes, on tape*
  - *SCRATCHDIR : big working space, can be purged after 40 days*

- We advice you to copy the **IPSL platform environment** in the home of your account :

> cp /ccc/cont003/home/igcmg/igcmg/MachineEnvironment/irene/bashrc ~/.bashrc

> cp /ccc/cont003/home/igcmg/igcmg/MachineEnvironment/irene/bashrc_irene ~/.bashrc_irene

- Add your email in file **~/.forward** to receive message at the end of simulation

TGCC

- Documentation
  - https://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvBtgcc and http://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvBtgccBirene
  - Command at irene : « irene.*info* »
  - *https://www-tgcc.ccc.cea.fr*

- Assistance : 01 77 57 42 42, hotline.tgcc@cea.fr

- Connexion :
  - ssh -X login@irene-fr.ccc.cea.fr

IDRIS

- Filesystems :
  - **ada:** *HOME : small space, back up*
  - **ada:** *WORKDIR : working space, no back up, no purge*
  - **ergon:** *HOME : for archive, on tape*

*Transfert between ada and ergon using **mfget and mfput***

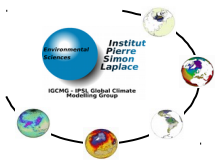
*From Ada it is possible to list files on ergon using **mfls***

*From Adapp (post-processing machine) it is possible to list files on ergon using **ls***

Adapp and Ada share the same disk space ($HOME and $WORKDIR)

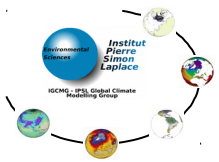
- We advice you to copy the IPSL platform environment in the home ada of your account :

```
ryyy999@ada: rm $HOME/.bash_profile
ryyy999@ada: cp ~rpsl035/.bash_login $HOME/.
```



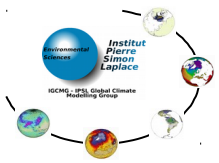
IDRIS

- Documentation :
  - *https://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvAidris*
  - *http://www.idris.fr*
- Assistance : 01 69 35 85 55, assist@idris.fr
- Connexion
  - ssh -X login@ada.idris.fr
  - ssh -X login@adapp.idris.fr
  - ssh -X login@ergon.idris.fr
- The password is the same on Ada, Adapp, Ergon and Turing. Use « *passwd* » on one of the machines to change it.
- Quota for the whole groupe. Use « *quota_u* » and « *quota_u -w* » to check.



MesoScale clusters

- Modipsl and libIGCM are also adapted to be used at
  - *Obelix* – LSCE cluster
    (http://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvClsce)
  - *Ciclad* and *ClimServ* – IPSL clusters
    (http://forge.ipsl.jussieu.fr/igcmg_doc/wiki/DocBenvDipsl)
- Following functionalities are adapted
  - Compilers
  - Computing jobs
  - Rebuilds
  - TS-SE
- Not adapted : pack, monitoring and full coupled-model.

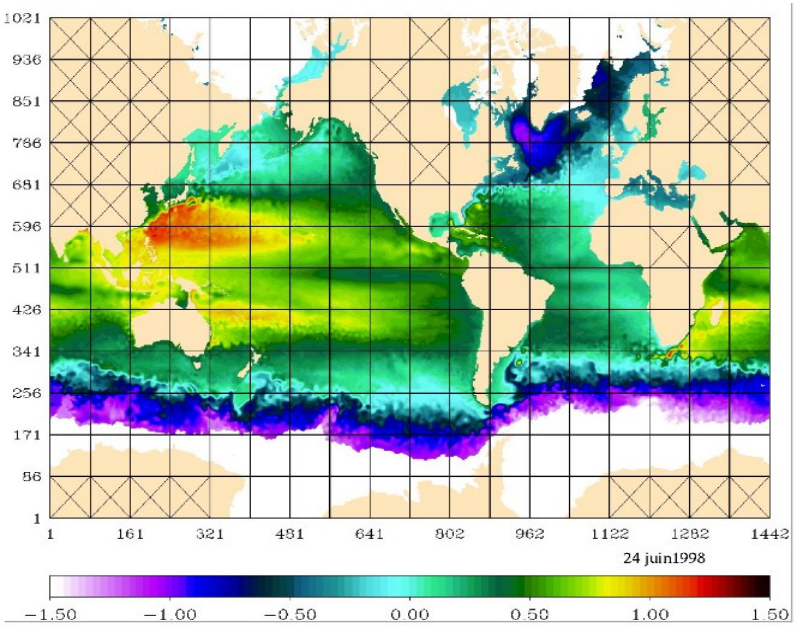


Why do we need supercomputer ?
⇒ parallelization!

All models are parallelised.

MPI parallelisation implemented in our modeles allows to run the same executable on several core MPI to reduce the real time of the execution.

→ The global domain is divided into sub-domains, each core treats one sub-domain

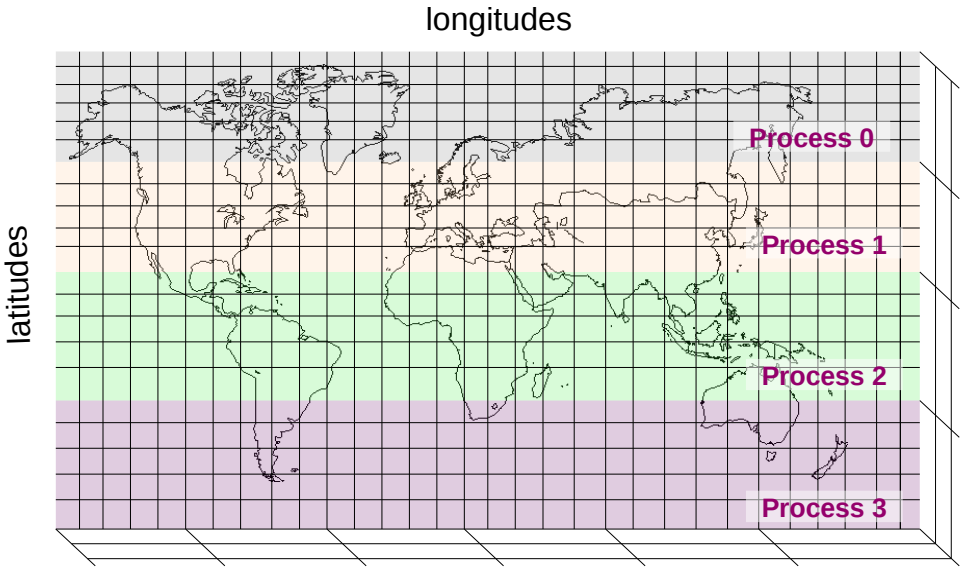


NEMO parallelism

Why do we need supercomputer ?
⇒ parallelization!

LMDZ model use hybrid MPI/Open MP parallelisation

MPI used to divide lon/lat grid splitting latitudes and Open MP to parallelise the vertical axis through shared memory threads.



LMDZ parallelisation grid

Outline

1. Introduction
2. High Performance Computing context
3. Which supercomputer(s) for us?
4. **modipsl : install and compile**
5. LibIGCM : create a simulation
6. LibIGCM : launch and follow a simulation

IPSL Compile and run environment

Software infrastucture based on **modipsl** and **libIGCM** which allows to :

modipsl

- **predefine and extract standard configurations**
- **compile sources from different components, coupling interfaces**

libIGCM

- **adapt and launch** predefined experiments
- **monitor** simulations
- **produce** and **store** results from models
- **produce, store** and **distribute** some analysis

Tools available for usage at TGCC, IDRIS, LSCE and IPSL cluster

Starting practical exercice: e-mail list

platform-users@ipsl.jussieu.fr

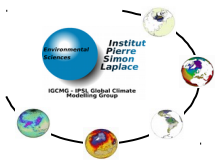
List for communication between users.

All user's can ask questions and answer his/her collegues questions.

→ ***All users need to subscribe***

Subscription on this page : https://listes.ipsl.fr/sympa/info/platform-users

Exercise 0 : subscribe to platform-users



modipsl

modipsl is a tool used **to download configurations, and prepare makefiles. Modeles and Tools are installed inside modipsl**.

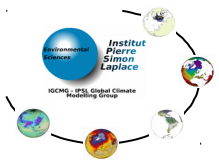
Exercise 1.0 – Install modipsl

Create a new directory in your $WORKDIR:

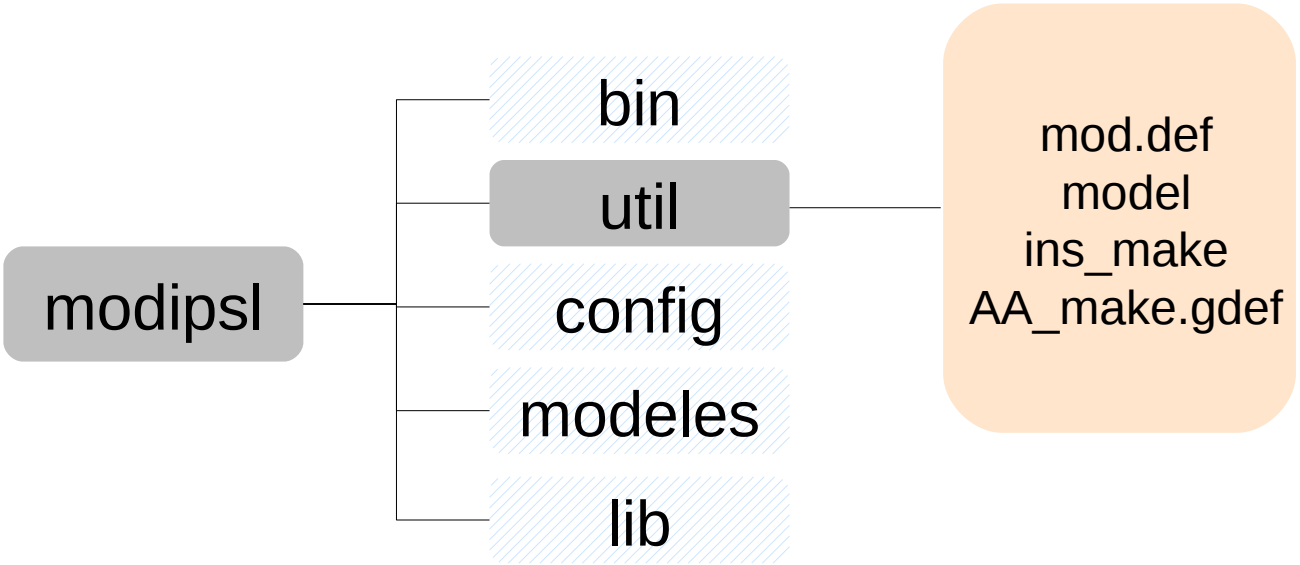
```
mkdir $WORKDIR/MYFIRSTTEST ; cd $WORKDIR/MYFIRSTTEST
```

Download modipsl from SVN repository:

```
svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
```



modipsl/util directory



mod. def → Description of components (tags and revision numbers) for each configuration

model → Script to extract available configurations

ins_make → Script to create makefiles adapted to the computer

AA_make.gdef → Compile options for all target machines

What do you mean by a configuration ?

A configuration is a combination of one or several models (components) coupled together

- *For example the configuration LMDZOR contains the two models LMDZ and ORCHIDEE.*

A configuration can be used for different experiments, using different set up, choice of parameters, etc.

- *For example with the configuration LMDZOR you can run experiments with different parametrizations for the physics in the atmosphere.*

- *For example with the configuration LMDZOR you can run an experiment with only LMDZ*

IPSL.ESM.6

LMDZ (Atmosphere)

ORCHIDEE (Surface)

NEMO – LIM – PISCES (Ocean)

INCA (chemistry – aerosol Tropo)

1 configuration – 1 executable

IPSL.ESM.6
LMDZ + ORCHIDEE + NEMO – LIM – PISCES + INCA

IPSLCM6
LMDZ + ORCHIDEE + NEMO – LIM – PISCES

LMDZORINCA
LMDZ + ORCHIDEE + INCA

LMDZOR
LMDZ + ORCHIDEE

LMDZ
LMDZ

Distributed configurations

→ « **v6 family** » : Recommended version of standard configurations. Parameters set up is the same for a component in all configurations in the same family. v6 configuration are actually use for CMIP6 and still under development.

IPSLCM6.0.X-LR	Version uses for CMIP6 of the coupled model *(currently IPSLCM6.0.15-LR)* Person in charge: A. Caubel
IPSLCM5A2	Standard version of the coupled model (VLR) Persons in charge : A. Caubel
LMDZOR_v6	LMDZ coupled with ORCHIDEE. Person in charge: J. Ghattas
LMDZORINCA_v6	LMDZOR_v6 coupled with INCA. Person in charge: A. Cozic

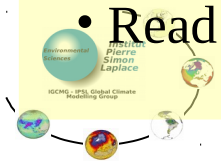
Distributed configurations

LMDZREPR_v5	LMDZ coupled with REPROBUS Person in charge: L. Falletti
NEMO_v6	Forced ocean model OPA-LIM-PISCES. Person in charge: C. Ethé.
ORCHIDEE_trunk	Forced continental surfaces model ORCHIDEE, with latest version on the trunk of ORCHIDEE. Person in charge: J. Ghattas.

General recommendation :

• *inform person in charge* before launching new studies based on one of these configurations, especially for coupled models.

• Read model and configuration documentation before using it !!!



Download a configuration
command “model”

`1) ./model -h` → list all available configurations

`2) ./model -h config` → give the list of components and their reference version for a specific configuration

`3) ./model config` → extract a configuration

Exercise 1.1 – Extract LMDZOR_v6 configuration

Some models need a password for extraction :

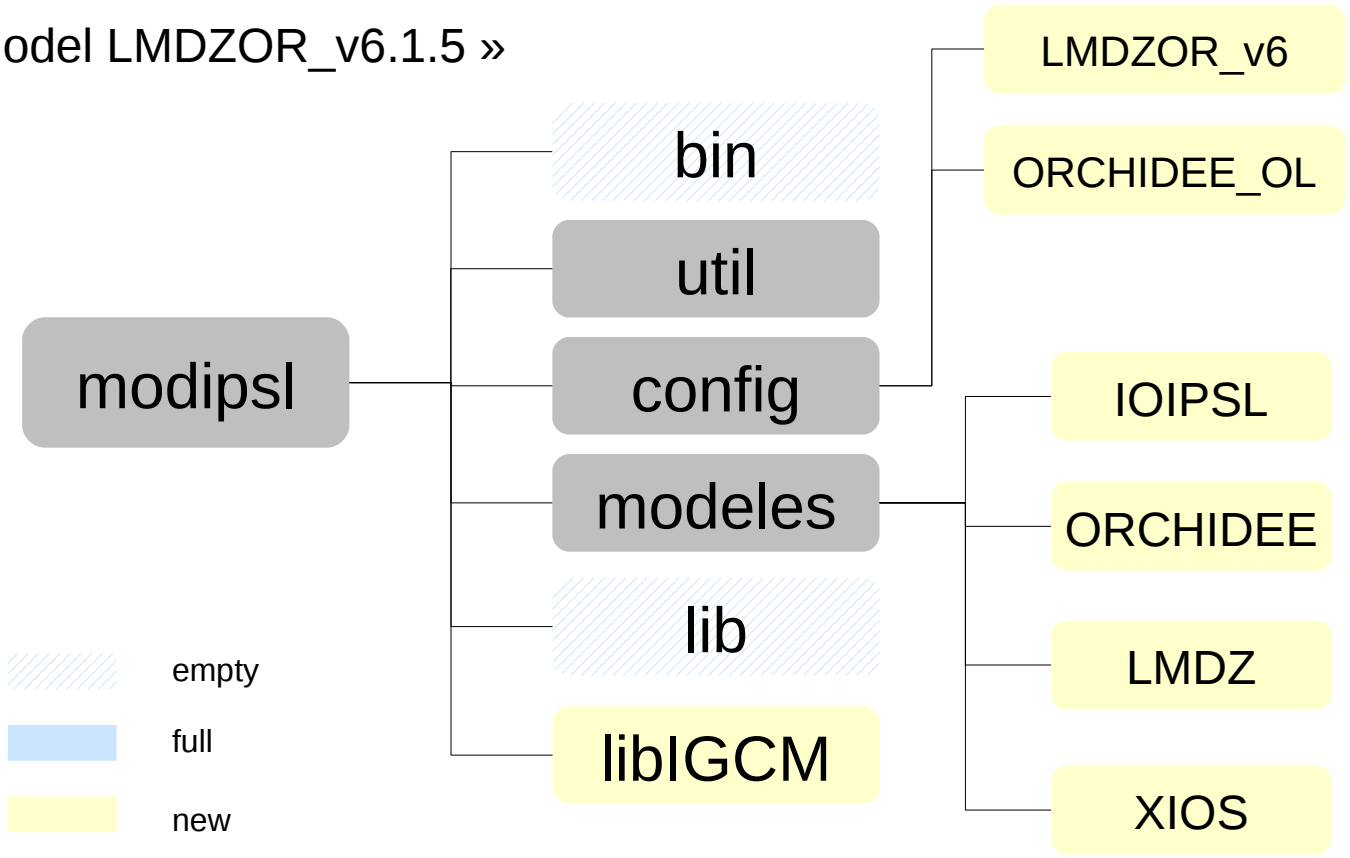
- ORCHIDEE (contact Josefine Ghattas)
- NEMO register on : http://www.nemo-ocean.eu
- INCA (contact Anne Cozic)
- REPROBUS (contact Lola Falletti)

Download a configuration

Example LMDZOR_v6.1.5 :

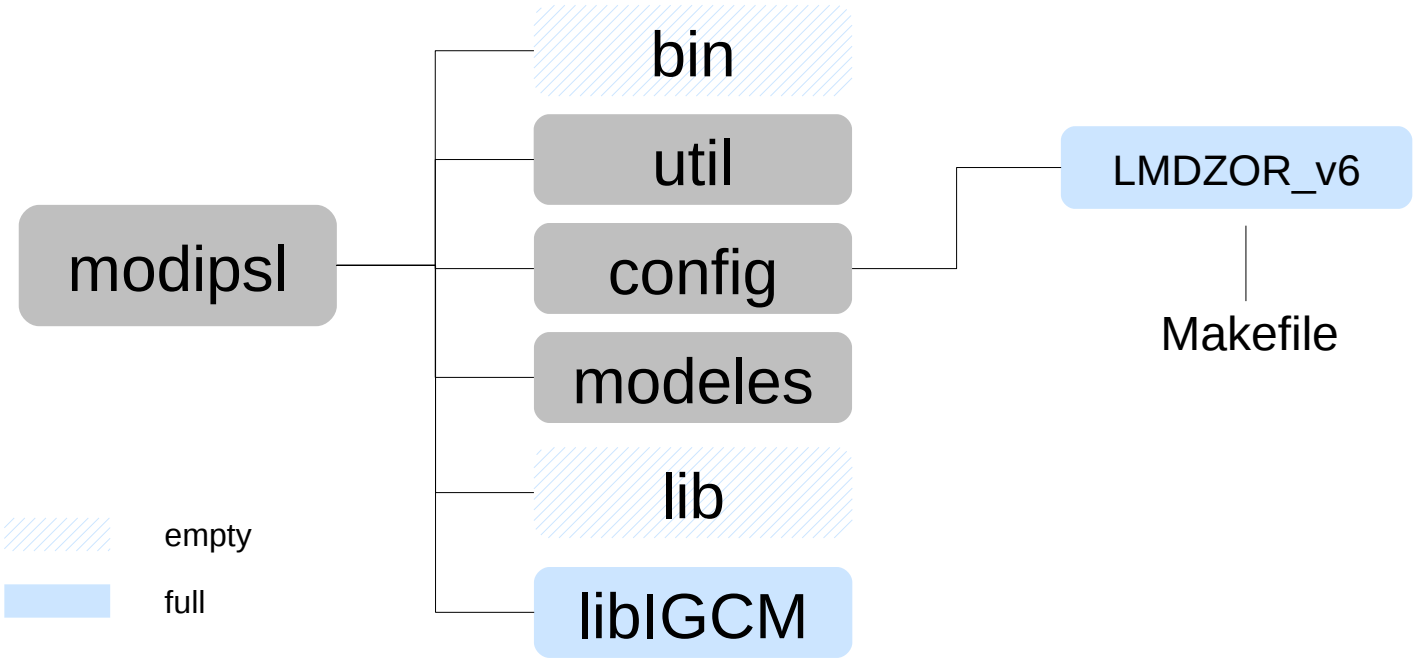
Yellow directories were extracted with

« ./model LMDZOR_v6.1.5 »



Compilation

Creation of Makefiles – adapt to the machine : Automatic at the end of extraction by the command *model*



Compilation

Exercise 1.2 : Compilation for the resolution 144x142x79

Choose the resolution of your model before compiling

→ Open the Makefile to know different choices

For example with LMDZOR_v6, in *Makefile* :

LMD5655 LMD9671 LMD9695 LMD9695-L39 LMD9695-L79 LMD128118-L39 LMD144142

LMD144142-L39 LMD144142-L59 LMD144142-L79 LMD512360-L79

Note :

- For each configuration there is a default resolution. It's indicate at the end of the line beginning by « all »

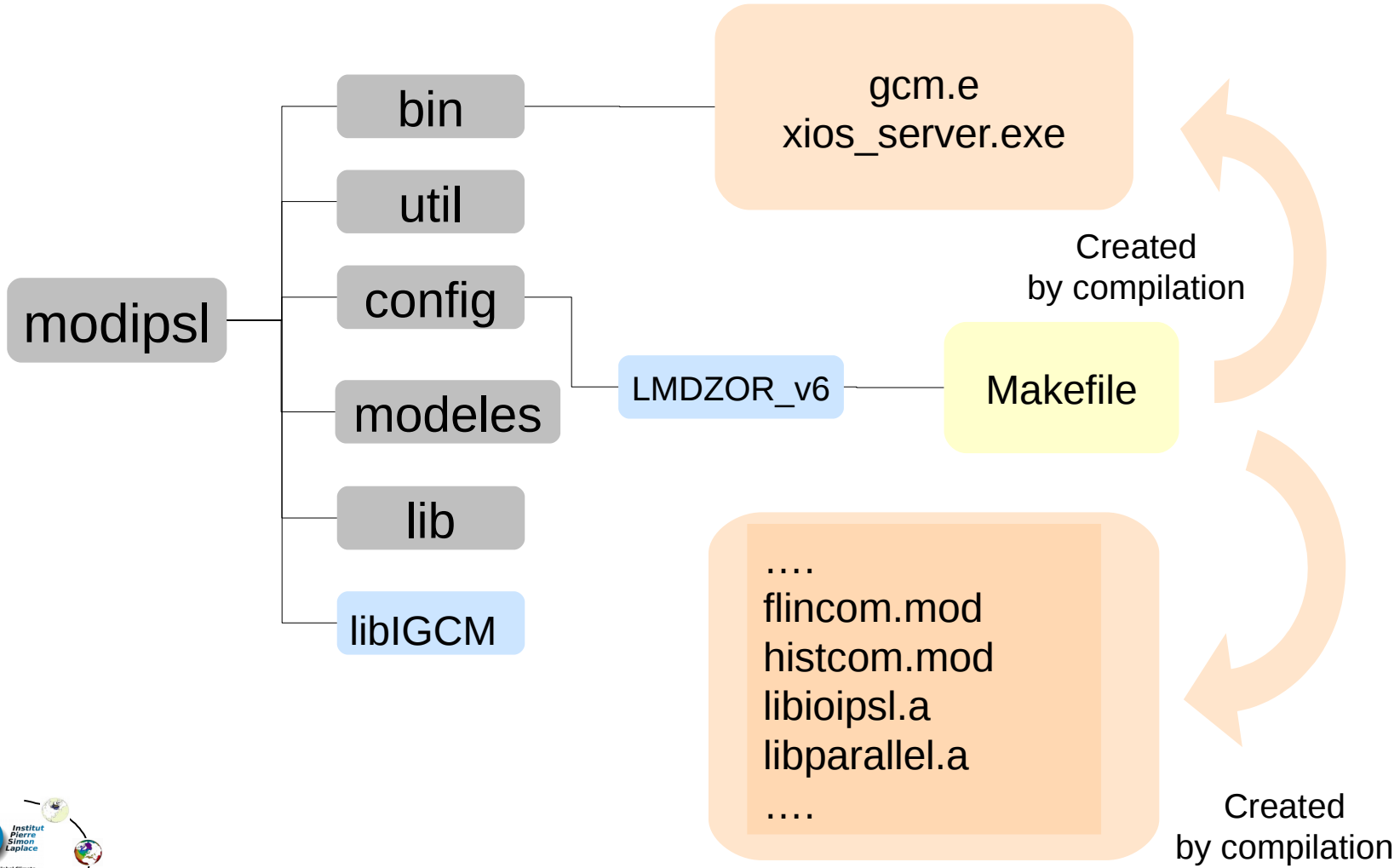
```
all :
    if [ -s ./.resol ] ; then $(M_K) `head -1 .resol |cut -c 8-` ; else $
(M_K) LMD144142-L79 ; fi
```

Compilation

File ***.resol*** (for all configurations except ORCHIDEE offline) is created at the end of compilation.

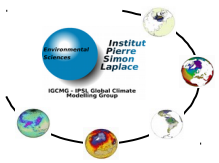
- If you want to recompile with the *.resol* resolution you can use the command « gmake » without any resolution target.
- If you recompile during an execution the next step of your simulation will use the new executable.

Executables in modipsl/bin



Outline

1. Introduction
2. High Performance Computing context
3. Which supercomputer(s) for us?
4. modipsl : install and compile
5. **LibIGCM : create a simulation**
6. LibIGCM : launch and follow a simulation



IPSL Compile and run environment

Software infrastucture based on **modipsl** and **libIGCM** which allows to :

modipsl

- **predefine and extract** standard configurations
- **compile** sources from different components, coupling interfaces

libIGCM

- **adapt and launch predefined experiments**
- **monitor simulations**
- **produce and store results from models**
- **produce, store and distribute some analysis**

Tools available for usage at TGCC, IDRIS, LSCE and IPSL cluster

libIGCM

Run and control environment
developped at IPSL for running the IPSL models.
Person in charge is S. Denvil.

Config directory

All configurations in «v6 families » have the same structure. Each of these configurations contains (in directory modipsl/config/xxx_v6) :

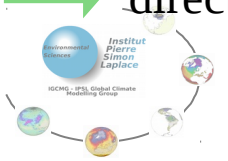
- What is needed **to compile** : AA_make, AA_make.ldef and for NEMO a directory *script.*

- **EXPERIMENTS** : In this directory you will find several directories for each experiment you can produce with the same executable. For example for LMDZOR_v6 you can choose experiments between LMDZOR et LMDZ.

- **GENERAL** : In this directory you will find scripts and parameters files independent of the experiment (divided into 3 directories **POST**, **PARAM** and **DRIVER**).

The submission directory doesn't exist and it has to be created. It will contain one directory from EXPERIMENTS and all directories from GENERAL.

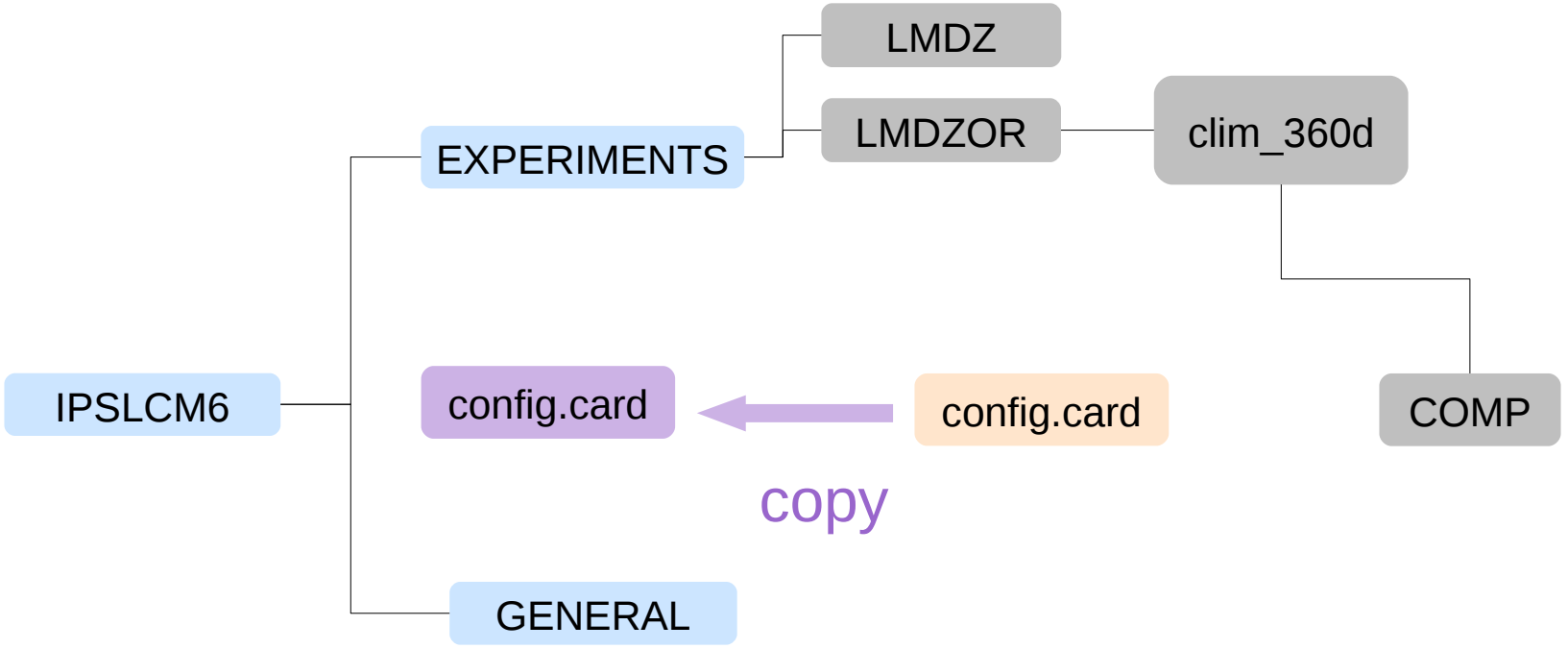
One configuration (Orchidee Offline) contain already several submission directories.



First step : Create submission directory

Exercise 2.1 – Create submission directory for experiment clim_360d

1. **Choose among predefined experiments in directory EXPERIMENT**
2. **Copy config.card**



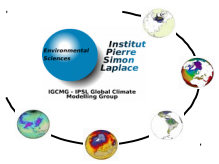
First step : Create submission directory

Exercise 2.1 – Create submission directory for experiment clim_360d

1. Choose among predefined experiments in directory EXPERIMENT
2. Copy config.card and **change at least *JobName* and *Parallelization options***

```
#D-- UserChoices -
[UserChoices]
#============================
JobName=MyJobTest

#D- For each component,
ATM= (gcm.e, gcm.e, 71MPI, 2OMP)
SRF= ("", "")
SBG= ("", "")
IOS= (xios_server.exe, xios.x, 2MPI)
```

Note : If you compile with the default resolution you don't need to change the parallelization options for MPI.



Standards parallelization :
- 144x142x79 → 71 MPI x 8 OMP + 1 MPI
- 96x95x39 → 31 MPI x 4 OMP + 1 MPI
- 39L → 4 OMP
- 79 L → 8 OMP

First step : Create submission directory

Exercise 2.1 – Create submission directory for experiment clim_360d

1. Choose among predefined experiments in directory EXPERIMENT
2. Copy config.card and change at least *JobName* and *parallelisation options*
3. **Use the command ins_job to create the submission directory. Submission directory will have the same name as JobName.**

../../libIGCM/ins_job

ins_job

When you launch ins_job command, it will ask you some questions :

Ada : (1 question)

Hit Enter or give MPI Environement (default is IBM), possible MPI environments are IBM (MPI IBM) and Intel (MPI Intel)

→ **by default enter for IBM**

Irene : (2 questions)

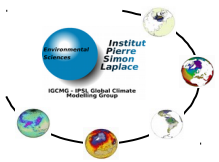
Hit Enter or give project ID (default is gencmip6), possible projects are gen2201 gen7719 gencmip6 :

→ **indicate on which project you will work**

ProjectID is gen2201 and ProjectNode for PostProcessing is standard

Hit Enter or give NUMBER OF CORES required for post-processing (default is "4"), possible numbers of cores are "1" to "48" :

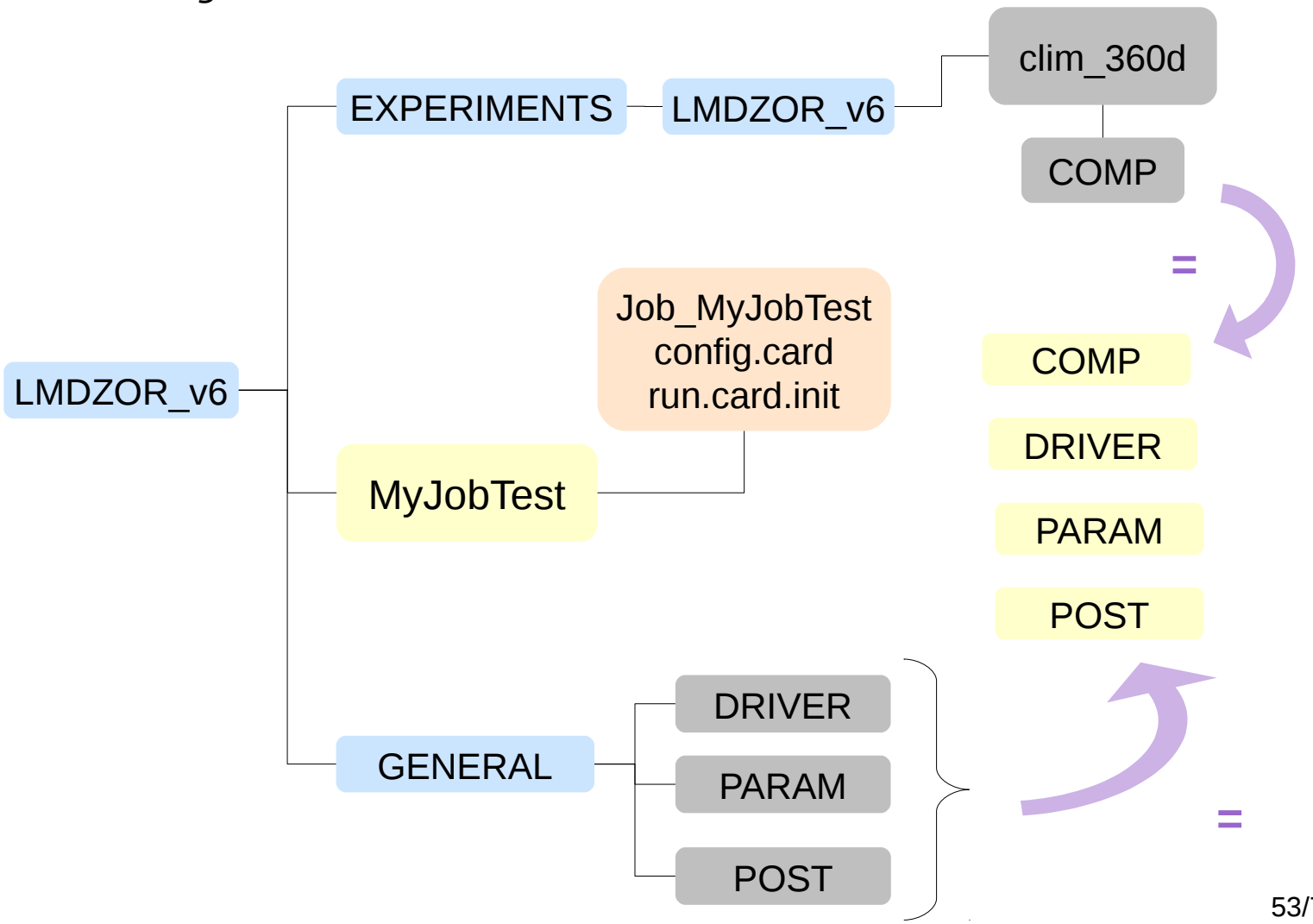
→ **choose the default except if you have some problem of time and memory**



Submission directory

After launch :

../../libIGCM/ins_job



Submission directory

Same organization for all coupled or forced configurations

COMP/ → Contains « *card* » files to **describe I/O** files for each component of the configuration and specific model options

DRIVER/ → Contains « *driver* » files to drive the functioning of each component of the configuration

PARAM/ → Contains parameter files for each component of the configuration.

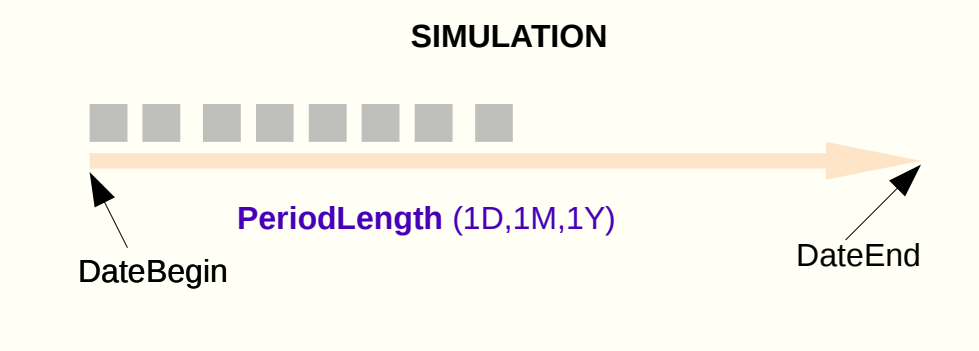
Job_*Jobname* → Main job

config.card → Main set up for the simulation

run.card.init / run.card → Execution details during run time

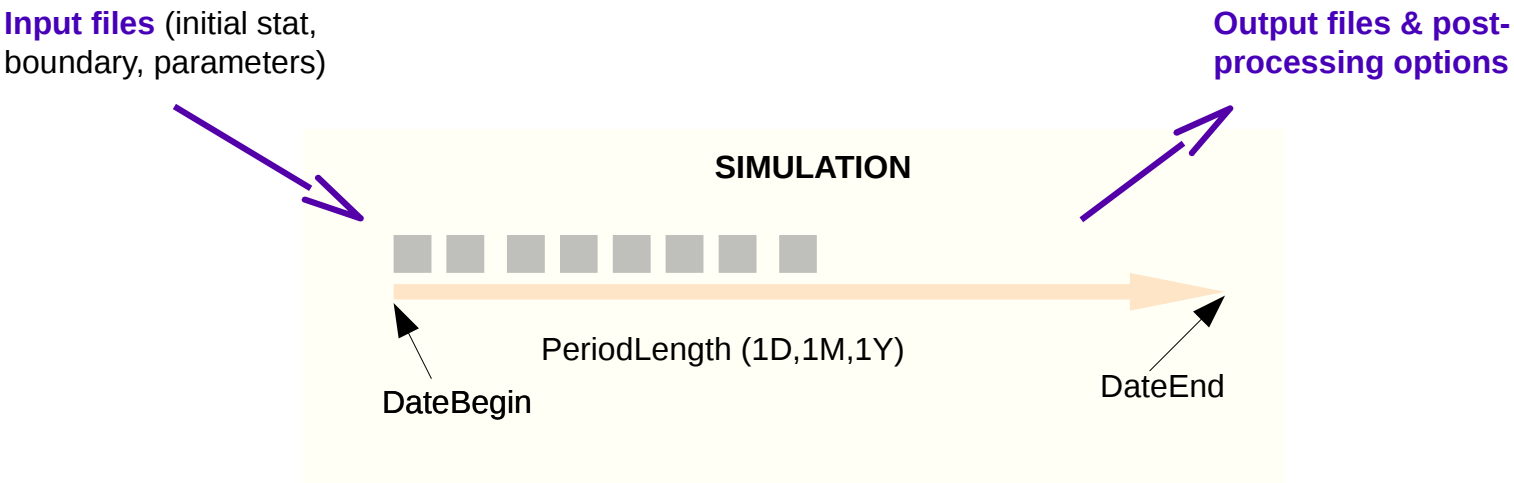
What is a simulation ?

- A simulation is a **succession of periods** between two dates with a configuration of models.



What is a simulation ?

- A simulation is a **succession of periods** between two dates with a configuration of models.
- You need to define **input** and **output files**.



What is a simulation ?

- A simulation is a **succession of periods** between two dates with a configuration of models.
- You need to define **input** and **output files**.
- 2 type of files are important to define all parameters of your simulation:
  - **config.card** (“identity card” of your simulation)
  - **COMP/<model>.card** (specifics information for each component : lmdz.card, orchidee.card, ...)

Second step : what do you need to define YOUR simulation ?

Name, and *type of simulation (Experiment Family, Status of your simulation)*

JobName=MyJobTest
#----- Short Name of Experiment
ExperimentName=clim
#----- TEST PROD DEVT
SpaceName=TEST
LongName="IPSLCM6 phase historical"
TagName=LMDZOR_v6

Config.card

The following output directory will be created,

***$ARCHIVE* / IGCM_OUT / TagName / SpaceName / ExperimentName / JobName**

Note : $ARCHIVE depends on the machine and on SpaceName :

SpaceName=*TEST* → $ARCHIVE=SCRATCHDIR (tgcc) / WORKDIR (IDRIS-ada)

SpaceName=*PROD / DEVT* → $ARCHIVE=STOREDIR + WORKDIR (tgcc) / HOME (IDRIS – ergon)

Second step : what do you need to define YOUR simulation ?

The *date of the beginning* of your simulation, *the date of the end*, with which *calendar* you want to work and the *period Length*

```
#-- leap, noleap, 360d
CalendarType=360d
#-- Experiment dates : Beginning and ending
#-- "YYYY-MM-DD"
DateBegin=2000-01-01
DateEnd=2000-01-01
#========================
#D-- 1Y, 1M, 5D, 1D
PeriodLength=1D
```

Config.card

Calendar leap = real calendar (365/366 days by year)
Calendar noleap = unreal calendar (365 days by year)
Calendar 360d = unreal calendar (360 days by year)

DateEnd = the last day of the simulation
Ex : For 1 simu of 1 day → DateBegin=DateEnd

PeriodLength = One period in your simulation, it can be 1day, 5days (for tests), 1month, 1 year (it's depend of the total length of your simulation)

Test simulation

New simulation → Test simulation → PeriodLength=1D ou 5D

Deactivate archive and post-processing: set in config.card :

```
SpaceName=TEST
TimeSeriesFrequency=NONE
SeasonalFrequency=NONE
```

Run on test queue: modify beginning of main job :

curie :

```
#MSUB -Q test
#MSUB -T 1800
```

ada :

```
@ wall_clock_limit = 01:00:00
```

Second step : what do you need to define YOUR simulation ?

Exercise 2.2 – Define and launch your first simulation of 1 day : setup config.card

- **For 1 day of simulation.**
- This is a first test simulation so keep **SpaceName=TEST**.
- **Deactivate all post-processing**:

Config.card

```
#D-- Post -
[Post]
#D- RebuildFrequency determines the frequency of rebuild submission
RebuildFrequency=NONE
#D- PackFrequency determines the frequency of pack submission
PackFrequency=NONE
#D- TimeSeriesFreqency determines the frequency of post-processing submission
#D- Set NONE to deactivate the creation of all time series
TimeSeriesFrequency=NONE
#D- SeasonalFrequency determines the length for each seasonal average
#D- Set NONE to deactivate the creation of all seasonal avereage
SeasonalFrequency=NONE
#D- Offset for seasonal average first start dates ; same unit as SeasonalFrequency
#D- Usefull if you do not want to consider the first X simulation's years
SeasonalFrequencyOffset=0
#D- If you want to produce compute PCMDI metrics from seasonal average
#D- Set TRUE or FALSE to activate/deactivate the metrics computation.
MetricsPCMDI=FALSE
```

Third step : prepare your simulation Job (Job_JobName)

You need to check your job header:

1) CPU Time
2) Parallelisation options

   → ins_job create the Job header, adapt to your config.card, but it's important to check a last time to be sure that you will launch what you are expecting.

3) Submission group (tgcc)

   → you need to know on which project you will work (if you have hour on several projects).

Job_*JobName*

IRENE

```
#!/bin/ksh
#######################
## IRENE   TGCC/CEA ##
#######################
#MSUB -r MyJobTest       # Job Name
#MSUB -o Script_Output_MyJobTest.000001    # standard output
#MSUB -e Script_Output_MyJobTest.000001    # error output
#MSUB -eo
#MSUB -n 143  # Number of MPI tasks : 71*2 + 1 =143
#MSUB -x
#MSUB -T 1800            # Wall clock limit (seconds)
#MSUB -Q test             # Test queue (max: 1800 seconds)
#MSUB -A ::default_project ::     #gen***
#MSUB -q skylake
#MSUB -m store,work,scratch
```

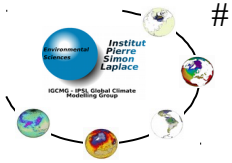
```
#=====================================
#D-- Executable -
[Executable]

ATM= (gcm.e, lmdz.x, 71MPI, 4OMP)
SRF= ("" ,"" )
SBG= ("" ,"" )
IOS= (xios_server.exe, xios.x, 1MPI)
```

Job_*JobName*

ADA

```
#!/bin/ksh
# ######################
# ##  ADA      IDRIS ##
# ######################
# Job name
# @ job_name = MyJobTest
# Standard output file name
# @ output = Script_Output_MyJobTest.000001
# Error output file name
# @ error = Script_Output_MyJobTest.000001
# Job type
# @ job_type = parallel
# Total number of tasks
# @ total_tasks = 72
# Specific option for OpenMP parallelization: Number of OpenMP threads per MPI task
# @ parallel_threads = 2
# Memory : as_limit=3.5gb max per process per core. With 4 threads per process use max as_limit=14gb
# @ as_limit = 7gb
# Maximum CPU time per task hh:mm:ss
# @ wall_clock_limit = 1:00:00
# @ environment = "BATCH_NUM_PROC_TOT=72" ; wall_clock_limit=$(wall_clock_limit)
# End of the header options
# @ queue
```

```
#=====================================
#D-- Executable -
[Executable]

ATM= (gcm.e, lmdz.x, 71MPI, 4OMP)
SRF= ("" ,"" )
SBG= ("" ,"" )
IOS= (xios_server.exe, xios.x, 1MPI)
```



Third step : prepare your simulation Job

The last important parameter : PeriodNb

HEADER
echo
echo "##"
echo "# ANOTHER GREAT SIMULATION #"
echo "##"
Echo
(…)
MODIPSL=(...)/modipsl
libIGCM=${MODIPSL}/libIGCM

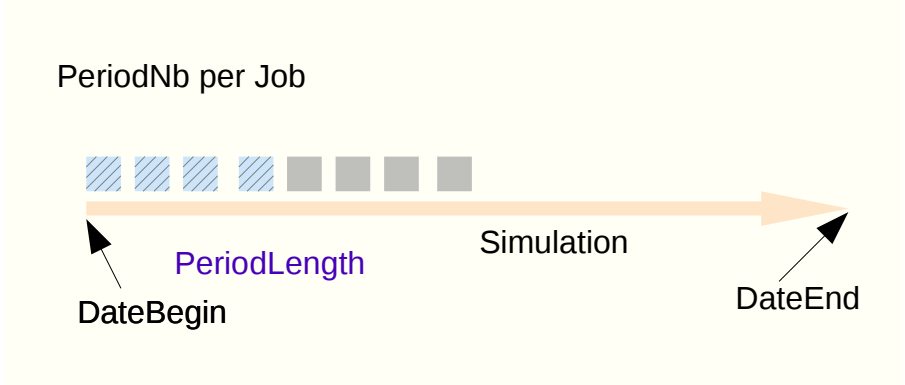
(…)
#D- Task type (computing or post-processing)
TaskType=computing
#D- Increased verbosity (1, 2, 3)
Verbosity=3
#D- Experience type : DEB(ug), DEV(elopment), RUN (default)
JobType=RUN
#D- Number of execution in one job
PeriodNb=1

Job_*JobName*
PeriodNb and cpu time

To avoid to launch a lot of small jobs waiting in queue, the main job can launch n period (parameter *PeriodNb* in Job) of *PeriodLength* (in config.card)

How to set PeriodNb :

CPU Time limit in job header ≥ PeriodNb * max(time exe PeriodLength)



You will receive an email « Simulation Accounting » that will indicate how many periodNb you can use if you are running on 24h.

Outline

1. Introduction
2. High Performance Computing context
3. Which supercomputer(s) for us?
4. modipsl : install and compile
5. LibIGCM : create a simulation
6. **LibIGCM : launch and follow a simulation**

Submission related commands

How to submit your job:

```
> cd modipsl/config/IPSLCM6/MyFirstExp
> ccc_msub Job_JobName (TGCC)
   or
> llsubmit Job_JobName (IDRIS)
   or
> qsub Job_JobName     (obelix)
```

Submission related commands on IDRIS, TGCC and obelix(LSCE)

	TGCC	**IDRIS**	obelix/ciclad
• Submission	**ccc_msub**	**llsubmit**	qsub
• Delete	**ccc_mdel**	**llcancel**	qdel
• Check the queue	**ccc_mpp**	**llq**	qstat
• Informations on queues	**ccc_mpinfo**	**news classes**	*intranet*
• Check quota	**ccc_quota**	**quota_u [-w]**	df -h

Prepare your simulation Job (Job_JobName) and launch it

Exercise 2.2 – Define and launch your first simulation of 1 day : verify Job_JobName

- **Verify the headers in the main job Job_MyJobTest.**
- **Launch the job:**

  llsubmit Job_MyJobTest / ccc_msub Job_MyJobTest

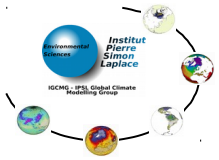
- **Verify that your job is running:**

  `llq -u $USER`

```
Id                       Owner      Submitted   ST PRI Class        Running On
------------------------ ---------- ----------- -- --- ------------ -----------
ada338-ib.1254426.0      UserID     11/23 14:43 R  100 c128t1       ada219-ib

1 job step(s) in query, 0 waiting, 0 pending, 1 running, 0 held, 0 preempted
```

  `ccc_mpp -u $USER`

```
USER   ACCOUNT   BATCHID  NCPU   QUEUE    PRIORITY  STATE  RLIM  RUN/START  SUSP  OLD   NAME        NODES/REASON
UserID gen2201    812513  576     skylake  260122    RUN    1.0h  31.0s      -     31.0s JobName     [1347,1356-1363,1772,1780,1784]
```



How to follow your simulation

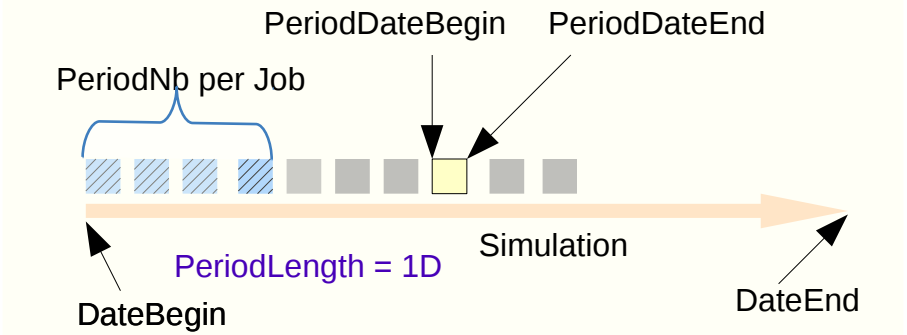
During a simulation, libIGCM create new files in the submission directory :

- **run.card** : information about the evolution of the simulation
- **Script_Output*** : script text output from libIGCM, all transfert of files

And you can also follow yout simulation on **Hermes web-service** (supervision of your jobs).

How to follow your simulation - **run.card** : monitor file

```
# $Revision:: 1052
# $Author:: sdipsl
(…)

#=========================================
[Configuration]
#Compute date of loop
PeriodDateBegin= 1980-01-06
PeriodDateEnd= 1980-01-06
CumulPeriod= 6
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState= Completed

SubmitPath= /gpfs5r/workgpfs/rech/dzt/rdzt910/MYFIRSTTEST/modipsl/config/LMDZOR_v6/MyJobTest
Simuid= (...)
#========================================================================
[PostProcessing]

#RebuildRunning=y/n
#RebuildToComplete=number of dir to rebuild
#RebuildDateID=PeriodDateEnd correspondant au dernier flag de RebuildFrequency=true

TimeSeriesRunning=n
TimeSeriesCompleted=

#========================================================================
[Log]
# Executables Size
LastExeSize= ( 91791005, 0, 0, 19145572 )

#-----------------------------------------------------------------------------------------------------------------------
---------------
# CumulPeriod | PeriodDateBegin |   PeriodDateEnd |        RunDateBegin |          RunDateEnd |     RealCpuTime |
#-----------------------------------------------------------------------------------------------------------------------
---------------
#             1 |        19800101 |        19800101 | 2018-11-23T12:20:13 | 2018-11-23T12:25:03 |       290.12000 |
#             2 |        19800102 |        19800102 | 2018-11-23T12:25:22 | 2018-11-23T12:26:47 |        84.94000 |
#             3 |        19800103 |        19800103 | 2018-11-23T12:27:20 | 2018-11-23T12:28:42 |        82.20000 |
#             4 |        19800104 |        19800104 | 2018-11-23T12:29:10 | 2018-11-23T12:30:34 |        84.32000 |
#             5 |        19800105 |        19800105 | 2018-11-23T12:31:06 | 2018-11-23T12:32:28 |        81.58000 |
```



Example : calculation of PeriodNb

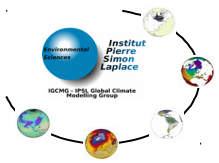
```
# $Revision:: 477                                     $ Revision of last commit
# $Author:: mmaipsl                                   $ Author of last commit
# $Date:: 2011-06-01 09:43:09 +0200 (Wed, 01 Jun 2011) $ Date of last commit
# IPSL (2006)
#  This software is governed by the CeCILL licence see libIGCM/libIGCM_CeCILL.LIC

#=====================================================================
[Configuration]
#last PREFIX
OldPrefix= tLMDZOR01_19800228
#Compute date of loop
PeriodDateBegin= 1980-03-01
PeriodDateEnd= 1980-03-31
CumulPeriod= 3
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState= Completed

SubmitPath= /workgpfs/rech/dzt/rdzt893/LMDZOR/TEST_libIGCMsanspack/LMDZOR_v4/modipsl/config/LMDZOR_v4/LMDZOR

#=====================================================================
[PostProcessing]

#RebuildRunning=y/n
#RebuildToComplete=number of dir to rebuild
#RebuildDateID=PeriodDateEnd correspondant au dernier flag de RebuildFrequency=true

TimeSeriesRunning= n
TimeSeriesCompleted= 19800228

#=====================================================================
[Log]
# Executables Size
LastExeSize= ( 60, 0, 0 )

#-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------
# CumulPeriod | PeriodDateBegin |   PeriodDateEnd |       RunDateBegin |         RunDateEnd |     RealCpuTime |     UserCpuTime |      SysCpuTime | ExeDate
#-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------
#           1 |        19800101 |        19800131 | 2012-03-30T18:04:30 | 2012-03-30T18:18:18 |       828.46000 |         3.48000 |         9.90000 | ATM_Mar_30_17:55
#           2 |        19800201 |        19800228 | 2012-03-30T18:19:58 | 2012-03-30T18:32:16 |       738.38000 |         3.24000 |         9.37000 | ATM_Mar_30_17:55
```

740s ~ 13min per simulated month

For a cpu time of 10H (example) :

PeriodNb = 10*60 / 13 = 46

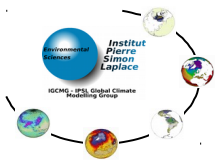
=> set PeriodNb=40



How to follow your simulation - **Script_Output** : text output

```
(load) ferret version 6.85
(load) netcdf seq 4.1.3
(load) nco version 4.4.6
(load) cdo version 1.6.5
Fri Nov 23 12:19:54 CET 2018

########################################
#       ANOTHER GREAT SIMULATION       #
########################################

(...)

########################################
#      DIR BEFORE RUN EXECUTION        #
########################################
(...)
EXECUTION of : /usr/bin/time poe -pgmmodel mpmd -cmdfile ./run_file
================================================================================

########################################
#       DIR AFTER RUN EXECUTION        #
########################################

(...)
```

No Error = good!



How to follow your simulation - **Script_Output** : text output

```
(load) ferret version 6.85
(load) netcdf seq 4.1.3
(load) nco version 4.4.6
(load) cdo version 1.6.5
Fri Nov 23 12:19:54 CET 2018

########################################
#       ANOTHER GREAT SIMULATION       #
########################################
(...)

########################################
#      DIR BEFORE RUN EXECUTION        #
########################################
(...)
EXECUTION of : /usr/bin/time poe -pgmmodel mpmd -cmdfile ./run_file
Return code of executable : 1
IGCM_debug_Exit :  EXECUTABLE

!!!!!!!!!!!!!!!!!!!!!!!!
!!   ERROR TRIGGERED    !!
!!   EXIT FLAG SET      !!
!------------------------!

0 - IGCM_debug_Exit (_0_)
IGCM_sys_Mkdir : /gpfs5r/workgpfs/rech/dzt/rdzt910/MYFIRSTTEST/modipsl/config/LMDZOR_v6/testError/Debug
IGCM_sys_Cp : out_execution
/gpfs5r/workgpfs/rech/dzt/rdzt910/MYFIRSTTEST/modipsl/config/LMDZOR_v6/testError/Debug/testError_19800101_19
800101_out_execution_error

########################################
#      DIR AFTER RUN EXECUTION         #
########################################
(...)
```

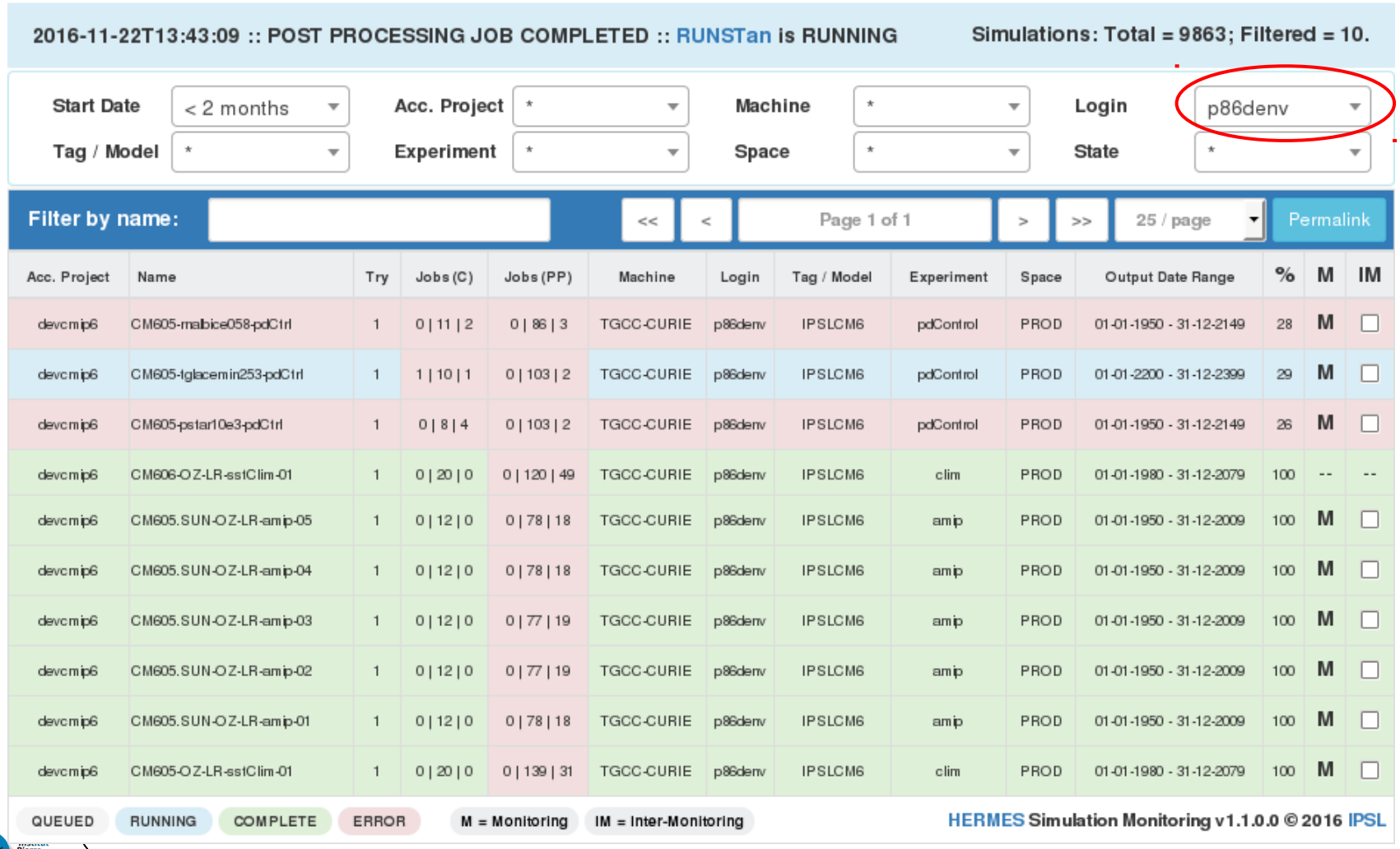
ERROR = oups problem!

=> If error during run execution: new folder **Debug/** in working directory

=> Only look at the **first error** message in this file (the next are only a consequence of the first one)

How to follow your simulation - **Hermes**

https://hermes.ipsl.upmc.fr



2016-11-22T13:43:09 :: POST PROCESSING JOB COMPLETED :: RUNSTan is RUNNING Simulations: Total = 9863; Filtered = 10.

Start Date: < 2 months | Acc. Project: * | Machine: * | Login: p86denv
Tag / Model: * | Experiment: * | Space: * | State: *

Filter by name: | << | < | Page 1 of 1 | > | >> | 25 / page | Permalink

Acc. Project	Name	Try	Jobs (C)	Jobs (PP)	Machine	Login	Tag / Model	Experiment	Space	Output Date Range	%	M	IM
devcmip6	CM605-malbice058-pdCtrl	1	0 | 11 | 2	0 | 86 | 3	TGCC-CURIE	p86denv	IPSLCM6	pdControl	PROD	01-01-1950 - 31-12-2149	28	M	
devcmip6	CM605-tglacemin253-pdCtrl	1	1 | 10 | 1	0 | 103 | 2	TGCC-CURIE	p86denv	IPSLCM6	pdControl	PROD	01-01-2200 - 31-12-2399	29	M	
devcmip6	CM605-pstar10e3-pdCtrl	1	0 | 8 | 4	0 | 103 | 2	TGCC-CURIE	p86denv	IPSLCM6	pdControl	PROD	01-01-1950 - 31-12-2149	26	M	
devcmip6	CM606-OZ-LR-sstClim-01	1	0 | 20 | 0	0 | 120 | 49	TGCC-CURIE	p86denv	IPSLCM6	clim	PROD	01-01-1980 - 31-12-2079	100	--	--
devcmip6	CM605.SUN-OZ-LR-amip-05	1	0 | 12 | 0	0 | 78 | 18	TGCC-CURIE	p86denv	IPSLCM6	amip	PROD	01-01-1950 - 31-12-2009	100	M	
devcmip6	CM605.SUN-OZ-LR-amip-04	1	0 | 12 | 0	0 | 78 | 18	TGCC-CURIE	p86denv	IPSLCM6	amip	PROD	01-01-1950 - 31-12-2009	100	M	
devcmip6	CM605.SUN-OZ-LR-amip-03	1	0 | 12 | 0	0 | 77 | 19	TGCC-CURIE	p86denv	IPSLCM6	amip	PROD	01-01-1950 - 31-12-2009	100	M	
devcmip6	CM605.SUN-OZ-LR-amip-02	1	0 | 12 | 0	0 | 77 | 19	TGCC-CURIE	p86denv	IPSLCM6	amip	PROD	01-01-1950 - 31-12-2009	100	M	
devcmip6	CM605.SUN-OZ-LR-amip-01	1	0 | 12 | 0	0 | 78 | 18	TGCC-CURIE	p86denv	IPSLCM6	amip	PROD	01-01-1950 - 31-12-2009	100	M	
devcmip6	CM605-OZ-LR-sstClim-01	1	0 | 20 | 0	0 | 139 | 31	TGCC-CURIE	p86denv	IPSLCM6	clim	PROD	01-01-1980 - 31-12-2079	100	M	

QUEUED | RUNNING | COMPLETE | ERROR | M = Monitoring | IM = Inter-Monitoring

HERMES Simulation Monitoring v1.1.0.0 © 2016 IPSL

How to follow your simulation - **Hermes**

https://hermes.ipsl.upmc.fr

DEVCMIP6 -> DEVT -> CM606.GUST [2]

2016-11-22T10:29:40 :: POST PROCESSING POST PROCESSING JOB COMPLETED :: CM606.GUST is RUNNING

OVERVIEW | CONFIG CARD | COMPUTE JOBS 1 3 0 | POST PROCESSING JOBS 7 252 60

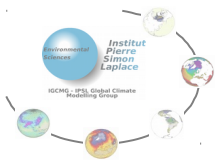
Acc. Project	devcmip6	**Output Start Date**	01-01-1950
Name	CM606.GUST	**Output End Date**	31-12-1999
Machine	TGCC-CURIE	**Output Progress**	36 %
Login	p529tra	**Compute Start Date**	19-11-2016 22:22:43
Experiment	pdControl	**Compute End Date**	22-11-2016 01:01:51
Model	IPSLCM6	**Compute Status**	RUNNING
Space	DEVT	**Try [Previous Tries]**	2 [1]
Submission Path	/ccc/work/cont003/gencmip6/p529tra/COUPLE/IPSLCM6.0.6/config/IPSLCM6/CM606.GUST		
Archive Path	/ccc/store/cont003/gencmip6/p529tra/IGCM_OUT/IPSLCM6/DEVT/pdControl/CM606.GUST		
Storage Path	/ccc/scratch/cont003/gencmip6/p529tra/IGCM_OUT/IPSLCM6/DEVT/pdControl/CM606.GUST		
Storage Path (Small)	/ccc/work/cont003/gencmip6/p529tra/IGCM_OUT/IPSLCM6/DEVT/pdControl/CM606.GUST		

Output directories

Basedir/IGCM_OUT/TagName/SpaceName/ExperimentName/JobName

Differents base directories :

	IDRIS	TGCC
Temporary ouput directory (before post-processing)	$WORKDIR (ada)	**$SCRATCHDIR**
Permanent archive directory (after post-processing)	$HOME (Ergon)	**$STOREDIR**
Permanent archive directory for small files (after post-processing)	Same as perm.	**$WORKDIR**



How to check that the simulation finished ?

- **run.card** : PeriodState=Completed
- Mail at the end of simulation
- Files on archive directory
- Post-treatment launched and finished
- Supervisor : Hermes

End of this part!

Now you can continue the exercices until 2.4 included.