

Utilisation du cluster

M. Gueguen, A. Morel,

5 février 2015

Objectifs de la présentation

faire le point sur l'utilisation du cluster

- comment accéder au cluster de calcul Thoret utiliser ses ressources ;
- architecture de la machine ;
- présenter les principales commandes et configuration de votre environnement ;
- compiler, analyser et lancer vos codes ;

① Hardware Cluster

Architecture

② Demande de compte

③ Connection

④ Software Cluster

bash

Modules

⑤ Compilation et libs MPI

Compilateurs

utiliser les optimisations/librairies : `mk1`, math kernel library
librairies MPI

⑥ Soumission des jobs : PBSPro

Ressources dans PBSPro

queues définies

- 1 Hardware Cluster
Architecture
- 2 Demande de compte
- 3 Connection
- 4 Software Cluster
- 5 Compilation et libs MPI
- 6 Soumission des jobs : PBSPro

Hardware Cluster

Nom du cluster : `thor.univ-poitiers.fr`

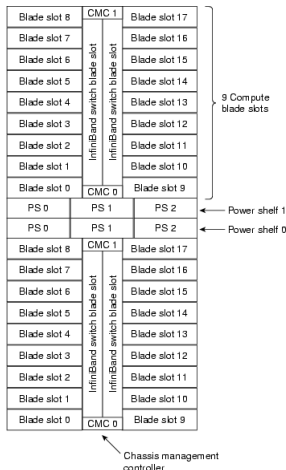
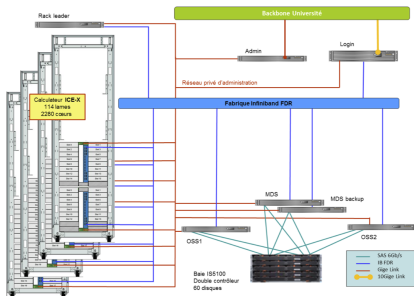
- situé au campus à l'université de Poitiers.
- 115 noeuds de calcul bi-processeurs :
 - processeurs 10 coeurs Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GH,
 - 64 Go de RAM par lame soit 3.2 Go par coeur.
- 2300 coeurs pour une puissance de 55 Tflops ;
- système de fichiers parallèle Lustre de 80 To ;
- réseau de calcul et stockage infiniband FDR 54 Gb/s ;

Architecture

Hardware Cluster

composants de la machine

- noeud de service
thor.univ-poitiers.fr :
40 coeurs *Intel(R) Xeon(R)*
E5-2670 v2 @ 2.50GHz;
- noeuds de calcul : r1iXnY :
X=1..16 et Y=1..18;



Hardware Cluster

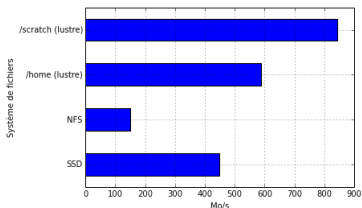
ça ressemble à



Stockage de vos données

Système de fichiers Lustre

- Lustre est dédié au calcul parallèle :
 - accès à des volumes de données de plusieurs centaines de péta-octets ;
 - support des milliers de noeuds de calcul ;
 - permet des E/S de supérieurs à 100 Go/s ;



- 1 Hardware Cluster
- 2 Demande de compte**
- 3 Connection
- 4 Software Cluster
- 5 Compilation et libs MPI
- 6 Soumission des jobs : PBSPro

2 - Demande de compte

- mail `hpc@support.univ-poitiers.fr`

Activation du compte pour une année

- Nécessiter d'effectuer une demande annuelle pour chaque utilisateur ;

- 1 Hardware Cluster
- 2 Demande de compte
- 3 Connection**
- 4 Software Cluster
- 5 Compilation et libs MPI
- 6 Soumission des jobs : PBSPro

3 - Connection- Prérequis pour l'utilisation

sur votre PC

- pour se connecter sur les machines de calcul : `putty`¹, `OpenSSH`² ;
- pour le transfert des fichiers entre votre PC et les cluster : `winscp`, `filezilla` ;

connection à distance via ssh

- création d'un compte sur le cluster
- authentification standard : par mot de passe

→ **connection standard** : `ssh -X -p 86 homer@thor.univ-poitiers.fr`

-
1. <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 2. <http://sshhwindows.sourceforge.net/download/>

3 - Connection- à la première utilisation

connection sur à la première utilisation

- activation des clés ssh sur le cluster : permet la connection de vos jobs parallèles ;
- diffusion des paramètres utilisateurs (login, password,...).
 - authentification par clé publique/privé ;

3 - Connection- Espace utilisateurs

connection sur à la première utilisation

- la partition /home défini l'espace utilisateur par défaut sur Thor, Cet espace utilise un système de fichier LUSTRE, et dispose de quota défini par utilisateur (500Go : soft ; 2To : hard ; 1 semaine de grâce).
- partition /scratch défini l'espace de travail du cluster. LUSTRE, non sauvegardé, pas de quota.
- partition /dev/shm emplacement mémoire monté de la même manière qu'un disque dur, les données sont placées directement en mémoire vive.

```
[homer@thor ~]$ lfs quota /home
Disk quotas for user homer (uid 1012):
Filesystem kbytes quota limit grace files quota limit grace
/home 40 524288000 2147483648 - 10 0 0 0 -
[homer@thor ~]$ df -h
Filesystem Size Used Avail Use% Mounted on
/dev/sda31 457G 58G 376G 14% /
tmpfs 32G 0 32G 0% /dev/shm
/dev/sda11 291M 49M 227M 18% /boot
10.148.0.3@o2ib:10.148.0.4@o2ib:/scratch 59T 1.6T 54T 3% /scratch
10.148.0.3@o2ib:10.148.0.4@o2ib:/home 22T 102G 21T 1% /home
```

- 1 Hardware Cluster
- 2 Demande de compte
- 3 Connection
- 4 Software Cluster**
 - bash
 - Modules
- 5 Compilation et libs MPI
- 6 Soumission des jobs : PBSPro

4 - Software Cluster- bash

Par défaut le shell d'un utilisateur est bash. Les fichiers de configuration sont dans le home d'un utilisateur (\$HOME, ie /home/homer) :

- .bashrc
- .bash_profile


Ces fichiers sont configurables en fonction de vos besoins. Vous pouvez ajouter des binaires, librairies, faire des alias... La commande pour connaître les variables d'environnement : `env`³

Les variables d'environnement :

- \$PATH : récupération des binaires, compilateurs, utilitaires
- \$LD_LIBRARY_PATH : récupération des librairies dynamiques

→ penser à surcharger la variable par :
`export PATH=$PATH:/home/homer/bin`⁴

3. pour faire des recherches : `env | grep PATH`

4. export : définition d'une variable d'environnement, propagée aux processus fils. 

bash

4 - Software Cluster- bash

modifier : `vi .bashrc`

```
#!/bin/bash

if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/new/path/to/lib/
export PATH="$HOME/bin:$PATH"
alias 'lt=ls -lhrt '
alias 'rm=rm -i '
alias 'cp=cp -i '
```

toute commande faite dans le terminal peut être intégrée dans le fichier.

4 - Software Cluster- Modules

Modules⁵ est un projet opensource fournissant des commandes pour modifier dynamiquement l'environnement utilisateur.

Modules configure pour les utilisateurs :

- Un code à utiliser pour calculer ;
- Le compilateur à utiliser ;
- La librairie MPI ;
- Un debugueur, un outil de profiling...

5. <http://modules.sourceforge.net>

4 - Software Cluster- Modules

Comment utiliser modules ?

Dans votre terminal ou dans votre script .bashrc voici les commandes à utiliser :

- `module help` : affiche l'aide (`man module`);
- `module avail` : Liste les outils disponibles;
- `module load` : Charge un outil particulier;
- `module unload` : Décharge une configuration;
- `module purge` : Détruit les configurations préalablement chargées;
- `module list` : Affiche la configuration actuelle de la session.

4 - Software Cluster- listes

les modules disponibles : `module avail` (référéncés sous différentes catégories)

- `codes/*` : codes disponibles sur Thor(codes éléments finis ; dynamique moléculaire ; abinitio)
- `intel-compilers-12|13|14` : Compilateurs Intel (icc/ifortran) ;
- `intel-cmkl-12|13|14` : blas/lapack optimisé
- `openmpi/1.8.3` | `mpt/2.10` | `intel-mpi-4` : Librairie MPI
- `lib/petsc`, `lib/mumps/4.10.0`, `lib/fftw3` : libs de calcul parallèle

```
[homer@thor ~]$ module avail
----- /usr/share/Modules/modulefiles -----
blcr          module-cvs  modules      null          perfcatcher
dot           module-info  mpt/2.10     perfboost     use.own
-----
codes/lammps/2013          intel-cmkl-13/13.1.2.183      intel-fc-14/14.0.2.144
codes/vasp/5.3/gamma      intel-cmkl-14/14.0.2.144      intel-mpi-4/4.1.3.048
codes/zebulon/Z8.6        intel-compilers-12/12.1.7.367 intel-tools-14/14.0.2.144
intel-cc-12/12.1.7.367    intel-compilers-13/13.1.2.183 lib/mumps/4.10.0
intel-cc-13/13.1.2.183    intel-compilers-14/14.0.2.144 lib/petsc/3.4.3
intel-cc-14/14.0.2.144    intel-fc-12/12.1.7.367       openmpi/1.8.3
intel-cmkl-12/12.1.7.367  intel-fc-13/13.1.2.183       utils/python
```

4 - Software Cluster- Modules

Exemple d'utilisation

- ❶ `module load intel-compilers-14`
- ❷ `module load intel-mpi-4`
- ❸ `mpif90 -version # verification`
- ❹ `make`

Pour certains modules

On peut utiliser des raccourcis dans les noms de module :

```
[homer@thor ~]$ module load mpt # au lieu de mpt/2.10
[homer@thor ~]$ module load lib/petsc # au lieu de lib/petsc/3.4.3
```

4 - Software Cluster- Modules

Modules, un outil qui vous veut du bien :

- Modules charge en tenant compte des dépendances ;
- Si une librairie, un code ne sont pas *préparés* avec le compilateur souhaité, un erreur est (doit être) affichée ;
- Les administrateurs modifient les modules pour propager un changement de configuration.

Important

Utiliser modules c'est aussi être sur d'avoir la bonne configuration pour utiliser les outils disponibles sur le cluster !

placement des codes

```
[homer@thor ~]$ ls /sw
abaqus  codes  compil  lib  Modules  octave  openmpi  pbs  python  robinhood  sdev
tools
[homer@thor ~]$ ls /sw/codes
abinit  espresso  lammps  vasp  wien2k  zebulon
[homer@thor ~]$ ls /sw/lib/
fftw  hdf5  metis  mumps  petsc
```

1 Hardware Cluster

2 Demande de compte

3 Connection

4 Software Cluster

5 Compilation et libs MPI

Compilateurs

utiliser les optimisations/librairies : `mk1`, math kernel library

librairies MPI

6 Soumission des jobs : PBSPro

5 - Compilation et libs MPI- Compilateurs

2 compilateurs disponibles

`gcc`⁶ et `intel v12,v13,v14`⁷

```
[homer@thor ~]$ gcc --version
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[homer@thor ~]$ ifort --version
ifort (IFORT) 14.0.2 20140120
Copyright (C) 1985-2014 Intel Corporation. All rights reserved.
```

6. sans module associé

7. avec modules associés

5 - Compilation et libs MPI- Compilateurs

2 compilateurs disponibles

Les modules gèrent le compilateur c/c++, fortran, ou les deux suivant l'appel et la version demandée :

```
[homer@thor ~]$ module avail intel
----- /sw/Modules/modulefiles -----
intel-cc-12/12.1.7.367      intel-cmkl-14/14.0.2.144      intel-fc-13/13.1.2.183
intel-cc-13/13.1.2.183      intel-compilers-12/12.1.7.367  intel-fc-14/14.0.2.144
intel-cc-14/14.0.2.144      intel-compilers-13/13.1.2.183  intel-mpi-4/4.1.3.048
intel-cmkl-12/12.1.7.367      intel-compilers-14/14.0.2.144  intel-tools-14/14.0.2.144
intel-cmkl-13/13.1.2.183      intel-fc-12/12.1.7.367
```

```
[homer@thor petsc_test]$ module help intel-compilers-12
----- Module Specific Help for 'intel-compilers-12/12.1.7.367' -----
Sets up the paths you need to use Intel 12.1.xxx C and Fortran compilers.
```

```
[homer@thor petsc_test]$ module help intel-cc-12
----- Module Specific Help for 'intel-cc-12/12.1.7.367' -----
Sets up the paths you need to use Intel 12.1.xxx C.
[homer@thor petsc_test]$ module help intel-fc-12
----- Module Specific Help for 'intel-fc-12/12.1.7.367' -----
Sets up the paths you need to use Intel 12.1.xxx Fortran.
```

```
[homer@thor petsc_test]$ module help intel-tools-14
----- Module Specific Help for 'intel-tools-14/14.0.2.144' -----
```

5 - Compilation et libs MPI- Compilateurs- optimisation

options de compilation : `ifort -xxx ; icpc -yyy`

- `-O3` optimisation maximale (défaut `-O2`)
- `-xAVX` generating optimized codes for ivy bridge chipset
- `-ipo` interprocedural (IP) optimizations between file
- `-parallel` look for loops to parallelize
- `-fast` correspond à `-ipo -O3 -no-prec-div -static`
- `-xHost`

5 - Compilation et libs MPI- Compilateurs- Debug

options de compilation : ifort -xxx ; icpc -yyy

- O0 optimisation minimale (défaut -O2)
- g generating debug symbol
- traceback traceback information when a severe error occurs at runtime
- check all Checks for all runtime failures. **Fortran only**
- check bounds checks on array subscript and character substring expressions. **Fortran only**
- check uninit Checks for uninitialized scalar variables without the SAVE attribute. Fortran only
- check-uninit Enables runtime checking for uninitialized variables. if a variable (local, scalar) is read before it is written, a runtime error routine will be called. C/C++ only
- ftrapuv Traps uninitialized variables by setting any uninitialized local
- debug all Enables debug information and control output of enhanced debug information (need -g).
- warn interfaces Tells the compiler to generate an interface block for each routine in a source file; the interface block is then checked
- fpe<0|1|3> control over floating-point exception at runtime
 - mp Enables improved floating-point consistency during calculations
 - ftz Flushes denormal results to zero when the application is in the gradual underflow mode

utiliser les optimisations/librairies : `mk1`, math kernel library

5 - Compilation et libs MPI- utiliser les optimisations/librairies : `mk1`, math kernel library

La `mk1` est une librairie de calcul optimisée fourni par intel pour les processeurs intel, concernant les calculs matrices/vecteurs bas niveaux et l'algèbre linéaire (BLAS/LAPACK), et les transformées de fourier. La librairie peut être multithread (via `openmp`) ou séquentiel.

```
[homer@thor ~]$ module load intel-compilers-14/14.0.2.144
[homer@thor ~]$ module load intel-cmk1-14/14.0.2.144
```

Utilisation de l'option de compilation `-mk1=<parallel|sequential|cluster>` : la version est parallèle par défaut :

```
[homer@thor ~]$ module load intel-tools-14/14.0.2.144
[homer@thor ~]$ ifort -o mycode_opt -fast -mk1
[homer@thor ~]$ mpif90 -o mycode_opt_inparallel -fast
-mk1=cluster # seq blas/lapack and blacs/scalapack
```

5 - Compilation et libs MPI- librairies MPI

3 librairies disponibles

- SGI MPT : `module load mpt/2.10`
 - Intel MPI : `module load intel-mpi-4`
 - Open MPI : `module load openmpi/1.8.3`
- par défaut, la librairie MPT est privilégiée pour le portage des codes sur Thor ;

MPT

- The MPI standard supports C and Fortran programs with a library and supporting commands. MPI also supports parallel file I/O and remote memory access (RMA).
- MPT supports the MPI 3.0 standard.
- Multirail InfiniBand support, which takes full advantage of the multiple InfiniBand fabrics available on SGI ICEX systems.
- Optimized MPI remote memory access (RMA) one-sided commands.
- High-performance communication support for partitioned systems.

5 - Compilation et libs MPI- librairies MPI

1ère utilisation : wrapper mpi

```
[homer@thor ~]$ module list
Currently Loaded Modulefiles:
  1) mpt/2.10
  2) intel-cc-14/14.0.2.144
  3) intel-fc-14/14.0.2.144
  4) intel-compilers-14/14.0.2.144
[homer@thor ~]$ mpif90 --version
ifort (IFORT) 14.0.2 20140120
Copyright (C) 1985-2014 Intel Corporation. All rights reserved.
[homer@thor ~]$ module purge
[homer@thor ~]$ module load mpt/2.10
[homer@thor ~]$ mpif90 --version
GNU Fortran (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)
Copyright (C) 2010 Free Software Foundation, Inc.
[homer@thor ~]$ module list
Currently Loaded Modulefiles:
  1) mpt/2.10
```

Wrapper vers ifort mais pas icc/icpc

```
[homer@thor ~]$ mpicc --version
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

5 - Compilation et libs MPI- bibliothèques MPI

sans les wrappers

```
[homer@thor ~]$ gcc -o myprog myprog.c -lmpi  
[homer@thor ~]$ icc -o myprog myprog.c -lmpi  
[homer@thor ~]$ g++ -o myprog myprog.C -lmpi++ -lmpi  
[homer@thor ~]$ ifort -o myprog myprog.f -lmpi
```


5 - Compilation et libs MPI- librairies MPI

L'appel des programmes se fait via

- `mpirun` : en local pour phase de tests,
 - `mpiexec` : interface de PBSPro vers MPT
 - `mpiexec_mpt` : interface de MPT vers PBSPro
- **utiliser `mpiexec_mpt` ou `mpiexec` obligatoirement pour la soumission dans les files**
- option `-n nb_proc` avec `mpiexec_mpt` optionnelle

- 1 Hardware Cluster
- 2 Demande de compte
- 3 Connection
- 4 Software Cluster
- 5 Compilation et libs MPI
- 6 Soumission des jobs : PBSPro**
Ressources dans PBSPro
queues définies

6 - Soumission des jobs : PBSPro- librairies MPI

Utilisation de PBSPro pour la soumission des jobs et la gestion des files d'attente.

Un premier script : vim myscript.pbs

```
#!/bin/sh
#PBS -l walltime=1:00:00
#PBS -N testjob
#PBS -l select=3:ncpus=20:mpiprocs=20:mem=200gb
#PBS -j oe
date
module load mpt/2.10
cd ${PBS_O_WORKDIR}
mpiexec_mpt ./my_application -myparams
echo "DONE"
```

soumettre le job

```
[homer@thor ~]$ qsub myscript.sh
```

6 - Soumission des jobs : PBSPro- Ressources dans PBSPro

explication

PBS fonctionne pour l'attribution des ressources sur la notion de *chunk*, correspondant à un ensemble de ressources défini comme une unité. Dans le cas de Thor, la première unité de calcul peut être vue comme un noeud de calcul. Les demandes d'unité de calcul se font par l'option : `-l select=` :

```
qsub -l select=[N:][chunk specification][+[N:]chunk specification] my_script.pbs
#PBS -l select=[N:][chunk specification][+[N:]chunk specification]
qsub -l select=20:ncpus=20:mpiprocs=20 my_script.pbs # demander 20 noeuds
# de calcul avec 20 coeurs et 20 processus MPI
```

attribution des ressources (calcul MPI standard)

Utilisez le triplet `select=X:ncpus=Y:mpiprocs=Y` par défaut pour l'attribution correcte des ressources

6 : queues définies- queues définies

files définies

Queue	Max core	Min core	MaxWalltime	DefaultWalltime	Max Job/user
small	20	1	600 : 00 : 00	24 : 00 : 00	10
normal	200	≫ 20	24 : 00 : 00	12 : 00 : 00	5
medium	400	≫ 200	12 : 00 : 00	06 : 00 : 00	2
large	1200	≫ 400	10 : 00 : 00	02 : 00 : 00	1

pour lancer un calcul

queue default routant vers les queues d'exécution en fonction des ressources demandées :

mettre `-q default` dans vos scripts.

recommandations

- walltime default sont ceux imposés par défaut à votre job (pour `small` : 24 :00 :00).

→ indiquer le walltime, à chaque soumission, le plus précisément.

6 : Soumission des jobs : PBSPro- options

Options	Use	Description
-P	#PBS -P <project>	Causes the job time to be charged to project. Useful for accounting
-l	#PBS -l select=1 #PBS -l walltime=00:10:00 #PBS -l mem=100GB #PBS -l place=	Number of chunk requested. It can correspond to compute nodes Maximum wall-clock time, in the format HH :MM :SS Maximum memory consumption placement type : the good way is to use default
-j	#PBS -j oe	join standard and error output in a file
-o -e	#PBS -o Nom_Sortie #PBS -e Nom_Sortie	sortie standard vers le fichier erreur standard vers le fichier
-M mail	#PBS -M homer@pp.fr	usermail
-N name	#PBS -N job_name	set job name
-m abe	#PBS -m abe	options de mail
-q queue	#PBS -q default	launch job in specified queue
-W depend =afterok:JOBID		création de dépendances entre job

6 - Soumission des jobs : PBSPro- Connaître l'état des jobs

`qstat` → donne la liste des tâches et leurs états

- ★ `qstat`, `qstat -a`, `qstat -p` # liste l'ensemble des jobs
- ★ `qstat -rn JOBID` # connaître les noeuds d'un job
- ★ `qstat -f JOBID` # tous les détails d'un job
- ★ `qstat -q`, `qstat -Q`, `qstat -Qf` # détail des files
- ★ `qstat -T` show estimated start
- ★ `qstat -s` show additional scheduler information
- ★ `qstat -H -u username` show history for user

Etat	signification	commentaire
Q	"en queue"	attente de ressource
R	running	jobs en cours
S	suspended	jobs suspendus
E	Exiting	sortie du calcul
H	Hold	action de l'utilisateur/admin

6 - Soumission des jobs : PBSPro- Connaître l'état des jobs

`qstat` → donne la liste des tâches et leurs états

- ★ `tracejob JID` Print log messages for a PBS job
- ★ `pbs_rdel` Delete a reservation
- ★ `pbs_rstat` Status a reservation
- ★ `pbs_rsub` Submit a reservation
- ★ `qalter` Alter job
- ★ `qdel` Delete job
- ★ `qhold` Hold a job
- ★ `qmove` Move job
- ★ `qorder` Order a job
- ★ `qrls` Release a hold job
- ★ `qselect` Select jobs by criteria
- ★ `pbsnodes rliXnY` Query PBS host

6 - Soumission des jobs : PBSPro- pour l'écriture de vos scripts

recommandations

- renseigner au plus juste le walltime
- si possible renseigner la mémoire occupée
- laisser `ncpus=20` permet d'avoir 3.2GB par coeurs
- toujours ajouter : `cd $PBS_O_WORKDIR` (cf option `-d path`)
- toujours travailler dans `/scratch`
- charger vos modules (cf §2)



coder le calcul du nombre de cpu : `NCPU=$(wc -l < $PBS_NODEFILE)` ou
utiliser directement `mpiexec_mpt`

- utiliser `mpiexec_mpt` si utilisation de la lib `mpi sgi mpt`

6 - Soumission des jobs : PBSPro- Monitoring

qq outils pour visualiser la charge machine

- ganglia (via firefox sur Thor→ <http://thor-admin/ganglia>)
- `pmice`
- `pmgcluster`
- scripts sous `/sw/tools/bin8` : `qmem`, `gload`, `pbsn`

gload

```
[homer@thor ~]$ gload
total cpus available   : 2460
total used nodes      : 64
total used process     : 1265
1mn load on thor      : 51.3743902439024 %
idle load on thor     : 48.7333333333333 %
```

8. à ajouter au `$PATH`

6 - Soumission des jobs : PBSPro- Documentation/support

On y travaille !

- Site internet <https://forge.univ-poitiers.fr/projects/mesocentre-spin-git/wiki>
- Site pour le support/demande
<https://forge.univ-poitiers.fr/projects/mesocentre-spin-git>
- mail hpc@support.univ-poitiers.fr
- publications et communications le méso-centre de calcul de l'Université de Poitiers par une phrase du style :
 - en français : Les calculs ont été effectués sur le calculateur du Méso-centre de calcul de Poitou Charentes.
 - en anglais : Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Poitou Charentes.