Formation Mésocentre de calcul de Franche-Comté

Guillaume LAVILLE

10 janvier 2013

UNIVERSITÉ DE FRANCHE-COMTÉ

mésocentre de calcul de franche-comté

Plan

- Présentation du mésocentre
- Connexion aux clusters
- Environnement de travail
- Utilisation interactive
- Soumissions de jobs

Présentation du Mésocentre

- Service de l'Université mis en place en 2009
- Un directeur Laurent Philippe
- Trois personnels à temps complet
  - Kamel Mazouzi, Ingénieur de Recherche (Conseil, responsable technique)
  - Guillaume Laville, Ingénieur d'Etude (développement, administration technique)
  - Cédric Clerget, Assistant Ingénieur (développement, administration technique)

Présentation du Mésocentre

- Faciliter la réalisation de traitements informatiques HPC
- Encourager la modélisation et la simulation à l'échelle de l'Université
- En fournissant deux types de ressources
  - Matérielles
  - Compétences

Présentation du Mésocentre

- Services accessibles gratuitement à tous les personnels Universitaires et étudiants
- Contrats avec des interventions industriels de la région
- Utilisation libre, sans limite de temps, basée sur un système de priorité
- Nombreux logiciels déjà installés
  - Matlab, compilateurs Intel C et Fortran, Meep...

Moyens de calcul

- Deux clusters de calculs et un noeud interactif
- Mesoshared
  - Exécution de programmes graphiques
- Mesocomte
  - Exécution de jobs parallèles et GPU
- Mesoseq
  - Exécution de tableaux de tâches séquentiels et mémoire partagée

Moyens de calcul : Mesoshared

- Adresse : **mesoshared.univ-fcomte.fr**
- Partagée entre les utilisateurs connectés
  - Pas de quotas mémoire ou processeur
- 32 coeurs d'exécution et 64 Go de mémoire vive
- Exécution de programmes interactifs ou de tests
  - Matlab, Comsol...
- Soumission impossible

Moyens de calcul : Mesocomte

- Adresse : **mesocluster.univ-fcomte.fr**
- Cluster de calcul réseau Infiniband
- 800 coeurs, 11 TFlops de puissance crête
- 74 machines dotées de 12 à 96 Go de mémoire vive
- Exécution de jobs parallèles en mémoire distribuée
- Quotas mémoire et slots

Moyen de calcul : Mesocomte

- Carte graphiques GPGPU :
  - 2 machines dotées de 12 Go de mémoire vive
  - 2 cartes Nvidia Tesla par machine
  - 240 coeurs par carte graphique
- Deux méthodes d'exploitation
  - Programmation directe en CUDA ou OpenCL
  - Utilisation de programmes existants optimisés (NAMD, Vasp...)

Moyens de calcul : Mesoseq

- Adresse : **mesoseq.univ-fcomte.fr**
- Cluster de calcul réseau Ethernet
- 228 coeurs, 4 TFlops de puissance crête
- 16 machines dotées de 48 Go à 64 Go de mémoire vive
- Exécution de tableaux de tâches séquentiels ou parallèles en mémoire partagée
- Quotas mémoire

Logiciels installés

- **Compilateurs :**
  - Intel Cluster toolkit (version 11.0) : 5 jetons
  - Compilateurs GNU (version 4.1.2)
- **Bibliothèques de passage de messages**
  - Intel MPI, Open MPI
- **Bibliothèques scientifiques**
  - Lapack, Scalapack, Blacs, Blas, PETSc, SuperLU, Intel MKL
- **Logiciels scientifiques**
  - Abinit, Meep, Espresso, OpenMX, NAMD, Gaussian 2009, Molpro, Vasp, Siclab

Logiciels installés

- **Logiciels commerciaux :**
  - **Matlab** 10 jetons + les toolbox suivantes :
    - Simulink
    - Image Processing Toolbox
    - Optimization Toolbox
    - Signal Processing Toolbox
    - Symbolic Math Toolbox
    - Statistics Toolbox
    - Compilateur 2 jetons
  - **Comsol**
    - COMSOL Multiphysics 2 jetons
    - Structural Mechanics Module 2 jetons

Deux types d'utilisation

- Lancement en mode interactif :
  - Exécution à distance
  - Interfaces graphiques
- Soumissions de jobs :
  - Désynchronisée
  - Plusieurs instances
  - Exécution parallèle

Plan

- Présentation du mésocentre
- Connexion aux clusters
- Environnement de travail
- Utilisation interactive
- Soumissions de jobs

Connexion aux clusters

- Trois points d'accès :
  - **mesocluster** et **mesoseq** : calcul en mode batch (SGE)
  - **mesoshared** : calcul interactif (matlab, comsol, applications graphique, ...)

utilisateur

mesoshared.univ-fcomte.fr

Calcul interactif

mesocluster.univ-fcomte.fr

Connexion aux clusters

- mesocluster, mesoshared : réseau privé université
- Accès extérieur:
  - VPN
  - Certificat :
    - Correspondant informatique
    - CRI
    - Mésocentre
  - Installation : http://vpn.univ-fcomte.fr/

Connexion aux clusters

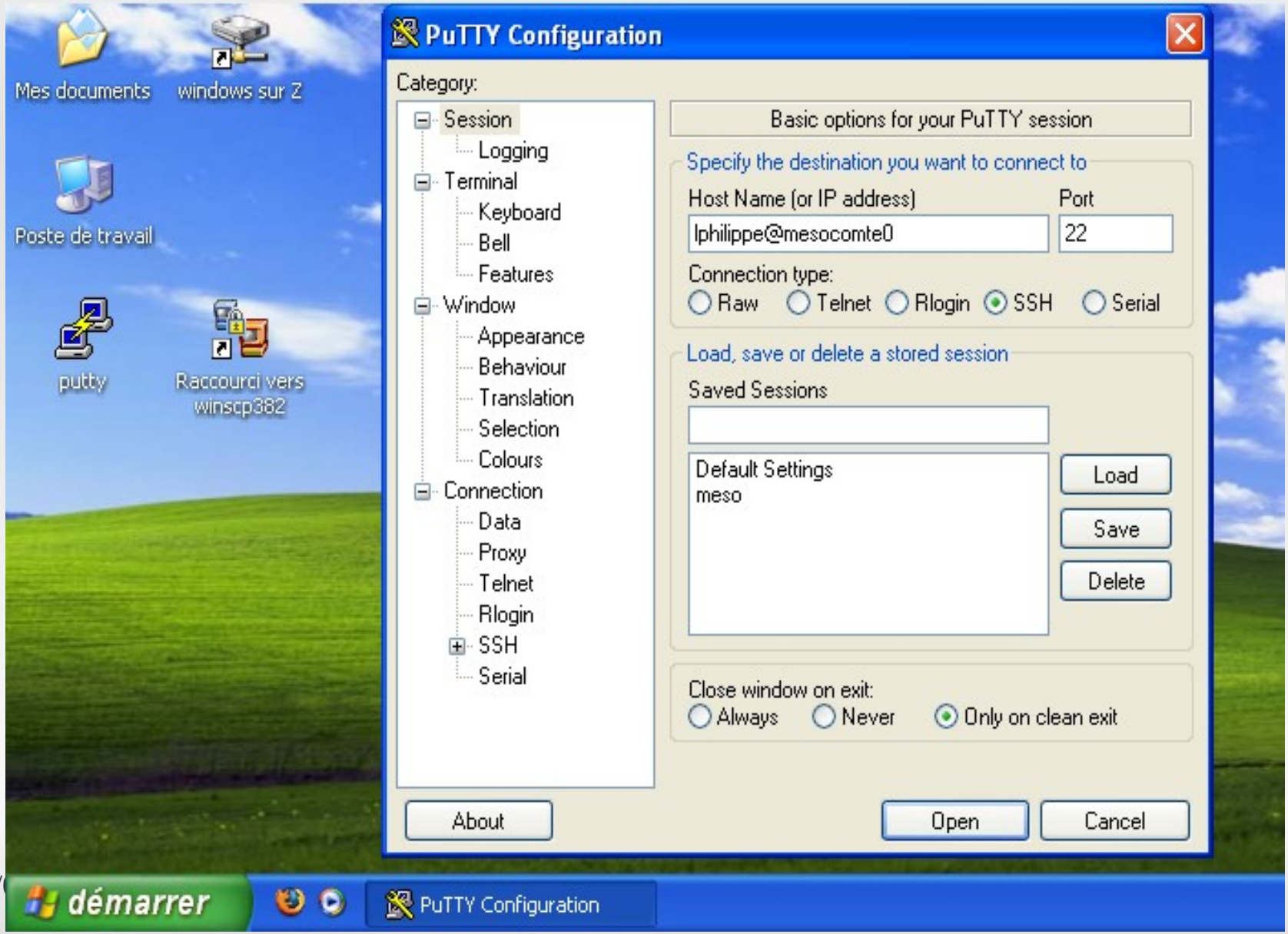
	Windows	Linux ou Mac OS X
Ligne de commande	PuTTY	SSH (inclus)
Copie de fichiers	WinSCP	Client FTP/SCP au choix (FileZilla, ...)
Session graphique	**NXClient**	**NXClient** ou ssh (forwarding X11)

- PuTTY : http://www.putty.org/
- WinSCP : http://winscp.net/
- NXClient (Windows, Linux, Mac OS X) : http://www.nomachine.com/download.php

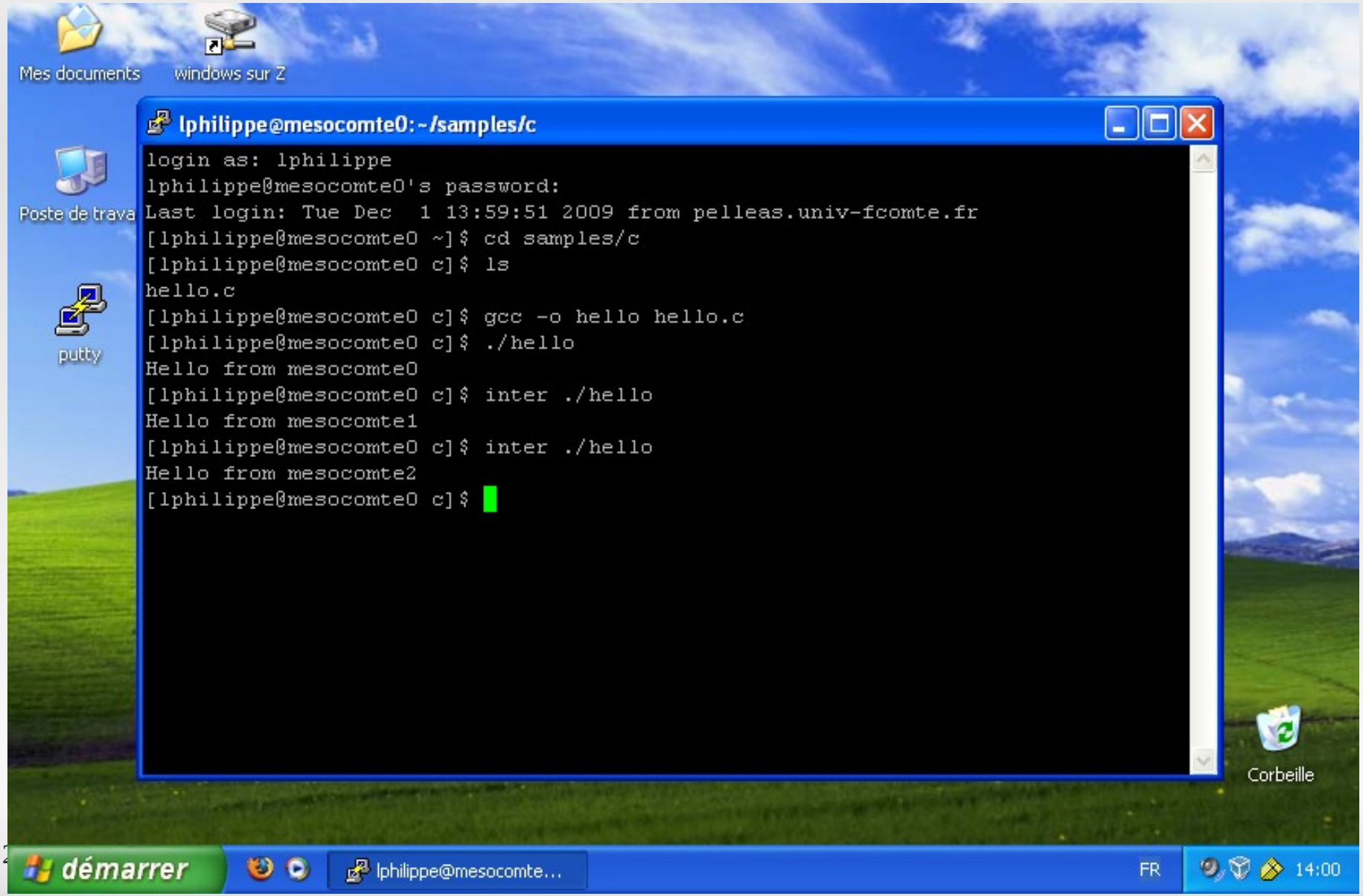
Travail pratique

- Connexion en ligne de commande
  - Au premier cluster de calcul, mesocomte
    - Point d'accès : mesocluster.univ-fcomte.fr
  - Au second cluster de calcul, mesoseq
    - Point d'accès : mesoseq.univ-fcomte.fr
- Connexion graphique à mesoshared
  - Adresse : mesoshared.univ-fcomte.fr

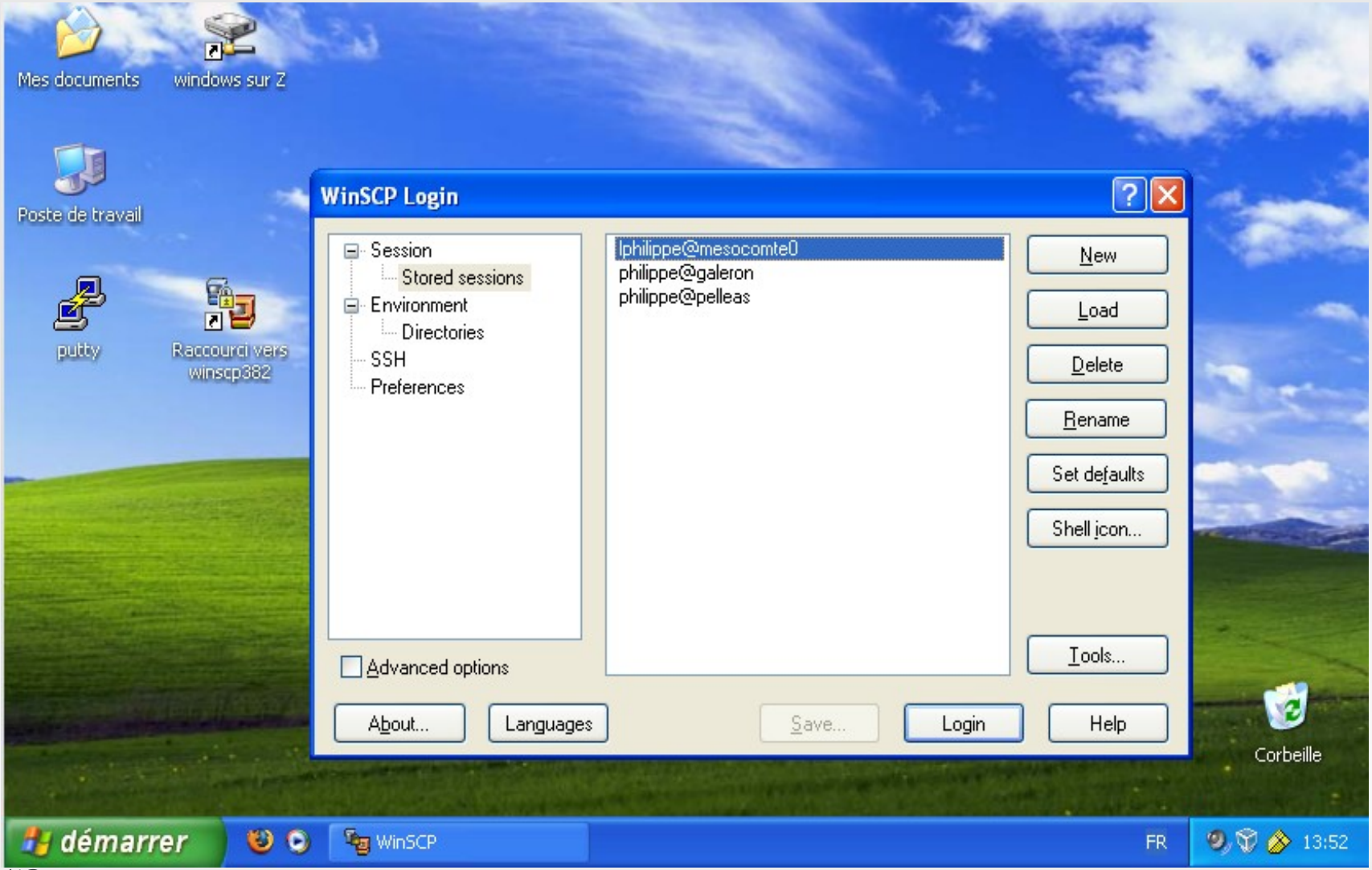
Exemple : PuTTY



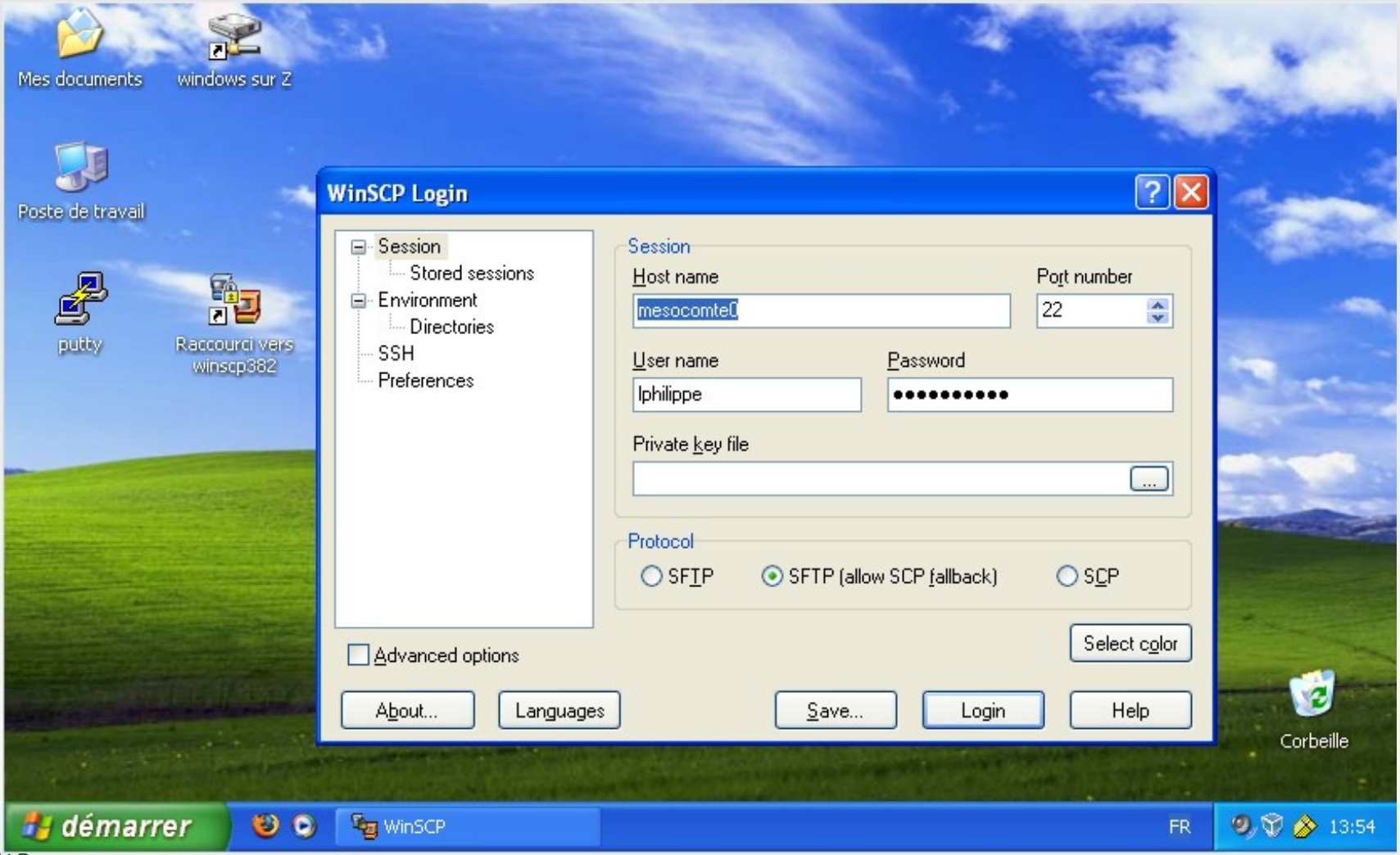
Exemple : PuTTY



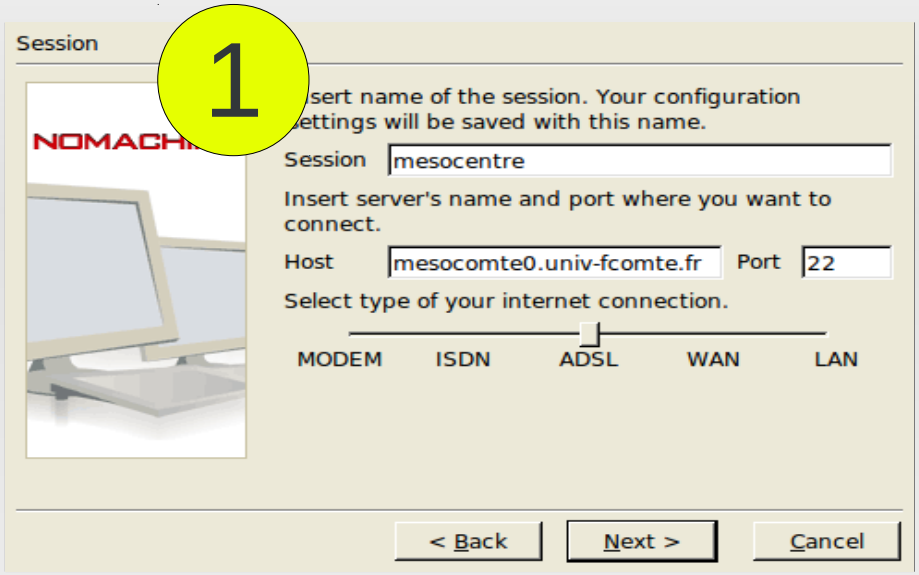
Exemple : PuTTY

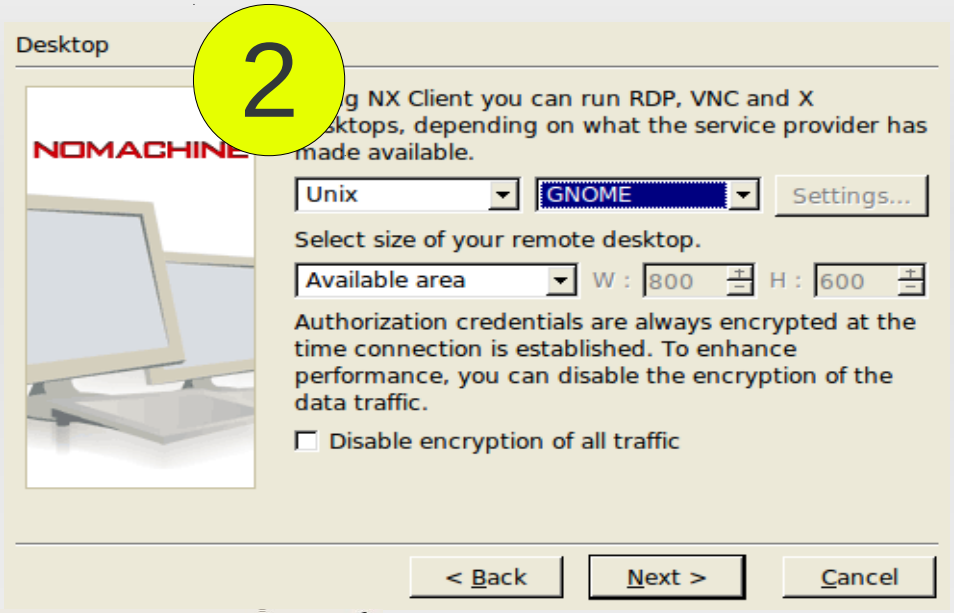


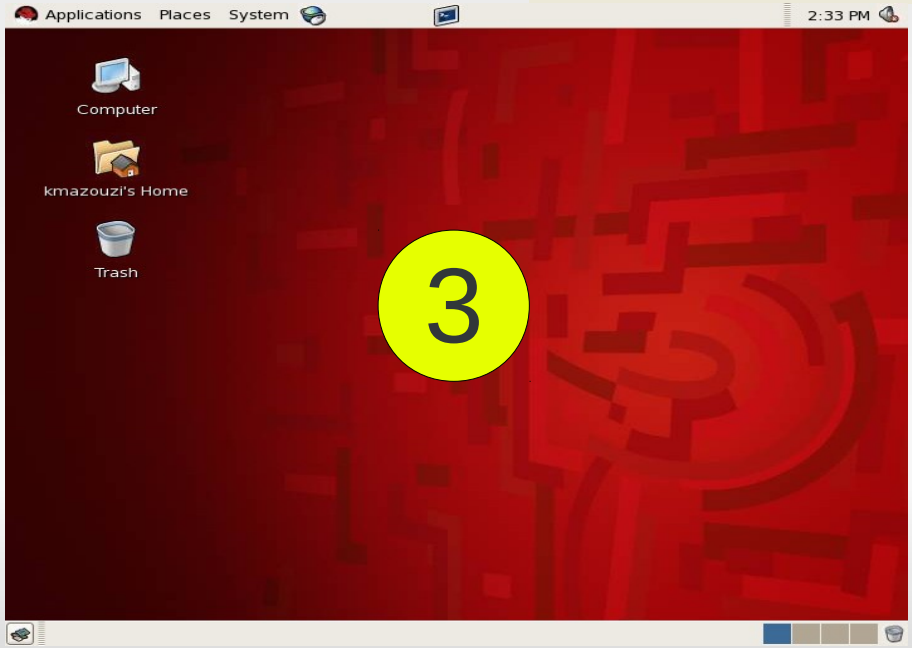
Exemple : PuTTY



Exemple : NXClient







Plan

- Présentation du mésocentre
- Connexion aux clusters
- Environnement de travail
- Soumissions de jobs
- Utilisation interactive
- Introduction au parallélisme

Environnement : Stockage

- **Répertoire personnel**
  - /Data/Users/<login>
  - Capacité de plusieurs Teraoctets
  - Principal emplacement de travail
- **Répertoires de projets**
  - Sur demande aux administrateurs
  - Permet le partage de données entre utilisateurs
- **Sauvegarde**
  - Chaque utilisateur est responsable de la sauvegarde de ses données
  - Utilisation possible du répertoire backup/

Environnement : Logiciels

- Disponibilité de nombreux logiciels
  - Versions parallèles/séquentielles, plus ou moins anciennes...
- Plupart demandent un chargement explicite
  - Exceptions : Compilateur Intel C/Fortran, GCC, Gaussian
- Utilisation de la commande **module**
  - module <action> <arguments>

Environnement : Logiciels

Vérification des logiciels disponibles : module avail

```
$ module avail


---------------------------------------- /opt/cuda/modulefiles ----------------------------------------
cuda/2.10

------------------------------------------ /Softs/modulefiles ------------------------------------------
abinit/5.8.4      fftw/2.1.5        graphviz/2.26.3   molpro/2009.1     scons/1.3.0
abinit/6.0.2      fftw/3.2.2        hdf5/1.9.63       tcl/8.5.8
cmake/2.8.1       fftw2/2.1.5       impi/4.0          ompi/1.3.4
comsol/3.5a       fftw3/3.2.2       java/1.6.0        petsc/3.1
espresso/4.1.2    ga/4.3            jcuda/0.3.1       python/2.6.5
espresso/4.2.1    grace/5.1.22      meep/1.1.1        ruby/1.9.1
```

Chargement d'un module : module load

```
$ module load comsol/3.5a
$ comsol
```

Environnement : Logiciels

Vérification des modules chargés : module list

```
$ module list

Currently Loaded Modulefiles:
  1) matlab/r2012b   2) cuda/4.2
```

Déchargement d'un module : module rm

```
$ module rm matlab
$ matlab
```

Chargementautomatiquement à la connexion

```
$ nano ~/.bashrc

... reste du fichier …

module load matlab
```

Travail pratique

- Vérification de l'existence du répertoire backup
- Création d'un nouveau fichier texte à sauvegarder
- Quels sont les modules actuellement chargés ?
- Comment lancer la version 2012 de Matlab ?

Plan

- Présentation du mésocentre
- Connexion aux clusters
- Environnement de travail
- Utilisation interactive
- Soumissions de jobs

Travail pratique

- Lancement d'une session graphique
- Lancement de Matlab
  - Fichier magicsquare.m
- Lancement de la fonction

Plan

- Présentation du mésocentre
- Connexion aux clusters
- Environnement de travail
- Utilisation interactive
- Soumission de jobs

Soumission de jobs

- SGE est un gestionnaire de batch
  - Équilibrage de charge
  - Partage des ressources
  - Gestion des jobs soumis en files d'attente (queues)
  - Système de priorité
  - Passage des jobs en fonction des ressources demandées par rapport à celles disponibles

Jobs : Terminologie

Q : Qu'appelle-t-on une *queue* ?

R : Une *queue* est une file d'attente
dans laquelle s'accumulent les jobs en
attente de traitement par le serveur.

Q : Qu'est-ce qu'un *job* ?

R : Une tâche ou *job* est un petit programme (shell)
contenant la définition de l'environnement
dans lequel il doit être exécuté

Jobs : Files d'attentes

Q : Combien y a-t-il de files d'attente sur le cluster ?
R : `qconf -sql`

```
[mesocomte0 ~]$ qconf -sql
bigmem2m
normal15d
normal2h
normal2m
tesla
```

Jobs : Les files d'attente

Organisation des file d'attente au mésocentre

Nom	Description
normal15d	Travaux parallèles en mémoire distribuée (durée inférieure à deux semaines, 650 coeurs)
normal2h	Travaux rapides (moins de deux heures) sur un nombre de nœuds limité (16 cœurs)
all.q	Travaux parallèles en mémoire partagée (durée inférieure à deux semaines, 228 coeurs)
tesla	Jobs GPGPU utilisant les technologies Tesla (16 cœurs)
bigmem2m	Jobs gourmands en mémoire (jusqu'à 96 Go, 8 cœurs)
interactive	Travail interactif : compilation, post-traitement ...

Soumission de jobs : qsub

```
qsub [options] [scriptfile]
```

```
man qsub
qsub -help
```

- **Options :** ressources
  - file d'attente
  - mémoire
  - nombre de nœuds
  - temps d'exécution
  - ...
- **Scriptfile :** script de lancement de l'application

En pratique, ces options sont Souvent intégrées dans le script

```
qsub -q tesla matlab.sge
```

Soumettre un job : les options

Les options principales SGE

Option	Explication
`#$ -q [queue]`	Queue à utiliser
`#$ -N [job]`	Nom du job
`#$ -V`	Exporter toutes variables d'environnement
`#$ -o [outfile]`	Nom du fichier où stocker la sortie standard
`#$ -e [errfile]`	Nom du fichier où stocker les erreurs
`#$ -pe [par]`	Environnement parallèle à utiliser

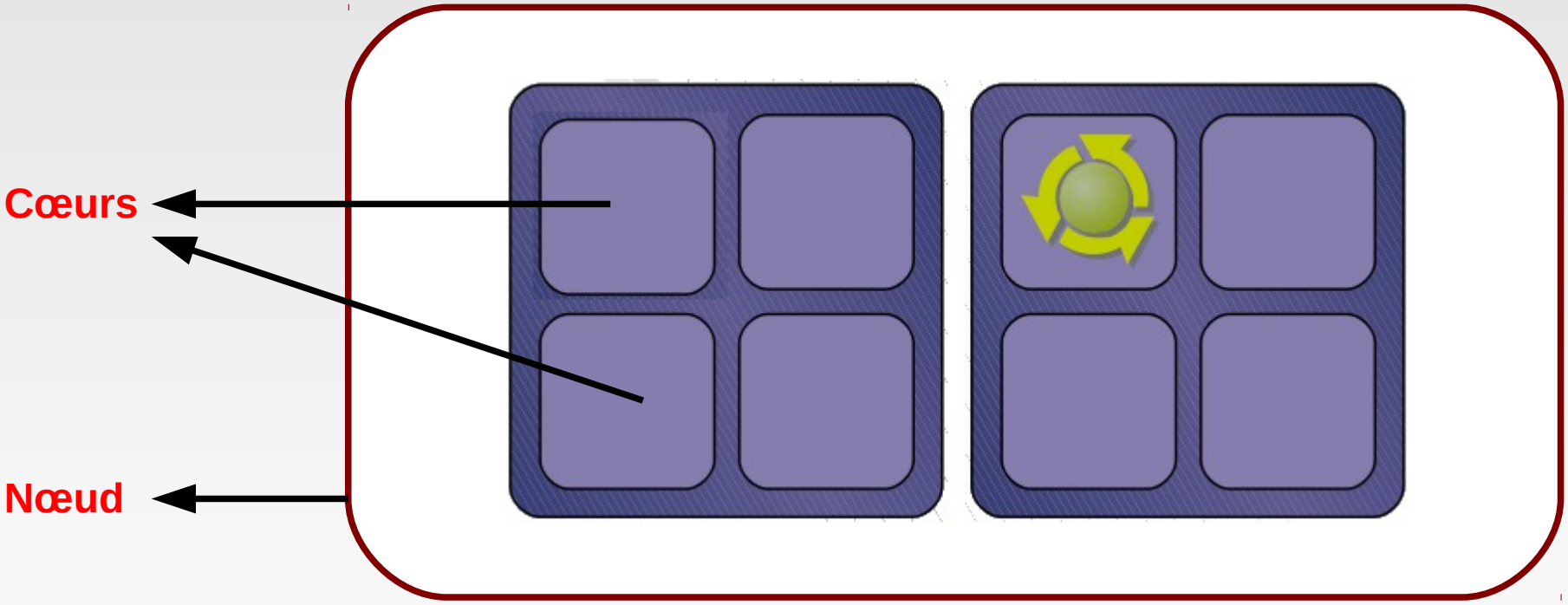
Soumettre un job : les options

Variables d'environnement SGE

Variable	Signification
`$JOB_ID`	L'identificateur unique propre au job.
`$JOB_NAME`	Nom du job défini par l'utilisateur (par défaut, correspond au nom du script)
`$TMPDIR`	Répertoire temporaire SGE, supprimé à la fin de l'exécution du job
`$TASK_ID`	L'identificateur du programme dans un tableau de tâche (**CF. Job parallèle**)
`$NSLOTS`	Nombre de cœurs demandés par l'utilisateur (**CF. Job parallèle**)

Soumettre un job séquentiel

Un job séquentiel s'exécute sur
un et un seul **cœur** d'un nœud particulier



Exemple : job séquentiel

```
qsub script.sge
```

script.sge

```
#!/bin/bash -l
#$ -q normal2h
#$ -V
#$ -N test_sge
#$ -cwd
#$ -o $JOB_NAME.$JOB_ID.out
#$ -e $JOB_NAME.$JOB_ID.err

cd $TMPDIR
pwd
sleep 30
exit 0
```

qstat

```
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
  90024 0.75000 test_sge   kmazouzi     r     06/29/2010 10:22:54 normal2h@mesocomte34           1
```

Mode interactif

Besoin : Se connecter sur un nœud de calcul
en mode interactif, utile pour la compilation
et le post-traitement.

Solution : qlogin

script.sge

```
[kmazouzi@mesocomte0 Exemples SGE]$ qlogin
Your job 90026 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 90026 has been successfully scheduled.
Establishing builtin session to host mesocomte36 ...
[kmazouzi@mesocomte36 ~]$
```

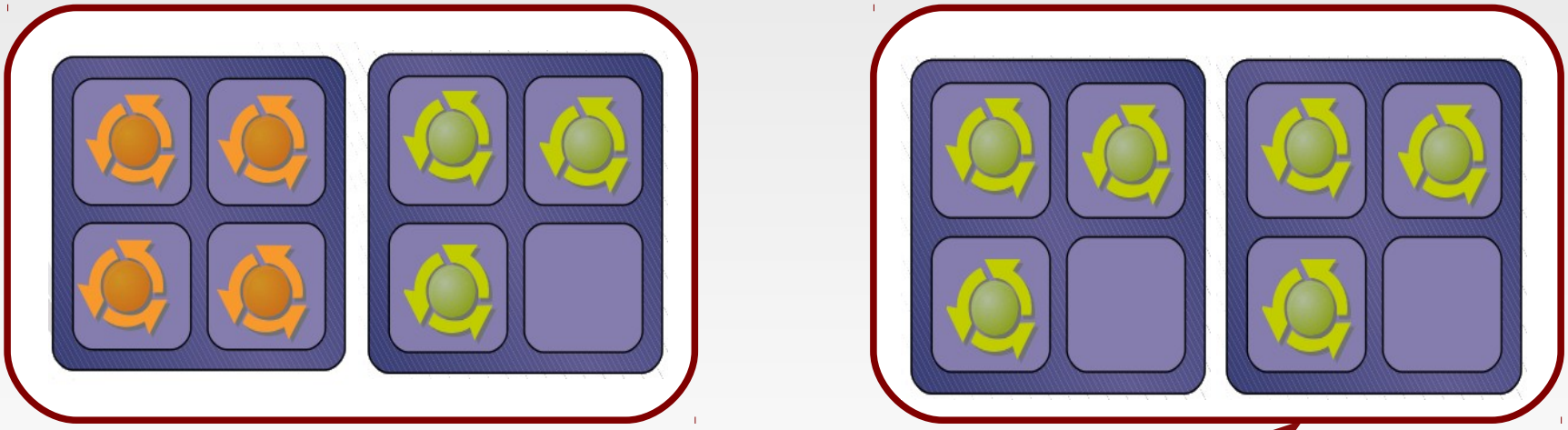
Taper **exit** pour quitter le mode interactif

Soumission de jobs parallèles

Soumettre un job parallèle

Un job parallèle s'exécute sur
plusieurs **cœurs**

1 nœud
Mémoire partagée



Plusieurs nœuds
Mémoire distribuée

Tableaux de tâches

Besoin : lancer plusieurs instances du même programme en parallèle sur des données différentes
Solution : Tableau de tâches

Exemple : lancer une application **`appli`** 100 fois en parallèle avec des données différentes :
`input1,...input100`

```
#!/bin/bash -l
#$ -q normal2h
#$ -V
#$ -N test_sge
#$ -t1-100
#$ -o $JOB_NAME.$JOB_ID.out
#$ -e $JOB_NAME.$JOB_ID.err

./appli input$TASK_ID
```

Application parallèle : openMP

Contexte : l'application s'exécute sur un nœud sur plusieurs cœurs

Solution : utilisation l'environnement parallèle

-pe openmp <nbSlots>

export OMP_NUM_THREADS=$NSLOTS

```
#!/bin/bash -l
#$ -q normal15d@@nonsusp15d
#$ -V
#$ -N test_sge
#$ -openmp 7
#$ -o $JOB_NAME.$JOB_ID.out
#$ -e $JOB_NAME.$JOB_ID.err

export OMP_NUM_THREAD=$NSLOTS

./appliopenmp input
```

Application parallèle : MPI

Contexte : l'application s'exécute sur plusieurs nœuds
Solution : utilisation l'environnement parallèle
-pe impi_tight <nbSlots>
-pe impi_robin <nbSlots>

impi_tight : rassemble au mieux les tâches sur un même nœud

impi_robin : distribue au mieux les tâches sur les nœuds

```
#!/bin/bash -l
#$ -q normal15d@@nonsusp15d
#$ -V
#$ -N test_sge
#$ -pe impi_tight 80
#$ -o $JOB_NAME.$JOB_ID.out
#$ -e $JOB_NAME.$JOB_ID.err

module load ompi

mpirun -machinefile $TMPDIR/machines
       -np $NSLOTS ./appli_mpi
```

Suivi des jobs

Suivi des jobs

Besoin : lister les jobs et les ressources disponibles
Solution : qstat, qhost

État du job	Explication
`d`	deleting
`t , r`	transfering, running
`s , S , T`	suspending , threshold
`R`	restarted
`w`	waiting
`h`	hold
`E`	error
`q`	queuing

Suivi des jobs

Quelques commandes utiles

Commande	Explication
`qstat -t`	Affichage complet des jobs sur les nœuds
`qstat -j Job Id`	Affichage complet d'un job en particulier
`qstat -u`	Affichage des jobs et des files associés à un utilisateur
`qstat -s z`	Affichage de jobs terminés (historique)
`qhost -q`	Affichage des ressources (nœuds, mémoire, swap,queue)
`qdel id`	Suppression du job id
`qdel -u toto`	Suppression de tous les jobs de l'utilisateur toto

FAQ

FAQ

Q : pourquoi mon job est-il en "Eqw" ?
R : Suite à une erreur dans votre script. Vérifier notamment que `#!/bin/bash` est en première ligne du shell, penser à passer **`dos2unix`** sur les fichiers windows

Q : pourquoi mon job reste-t-il bloqué (en "qw") ?
R : SGE n'arrive pas à trouver les ressources demandées,
pour plus d'informations, il est possible d'utiliser **`qstat -j idJob`**

Q : Comment indiquer le temps d'exécution du job?
R : En utilisant l'option -l h_rt,

- en ligne de commande :

```
qsub -l h_rt=02:30:00 job.sge
```

- dans le script : `#$ -l h_rt=02:30:00`

FAQ (2)

Q : pourquoi indiquer le temps d'exécution ?
R : pour faciliter la répartition des ressources par GE

Q : Que se passe-t-il si le temps indiqué est trop court ou trop long ?
R : Si le temps indiqué est trop court, votre job sera interrompu.
S'il est trop long, vous serez pénalisé par GE qui utilise cette Information pour déterminer la priorité des jobs.

Q : comment demander une certaine quantité de mémoire?
R : en utilisant l'option -l h_vmem,

```
#$ -l h_vmem=10g
```

Mémoire nécessaire à l'exécution (par slot)
Attention : le job sera tué en cas de dépassement !

Travail pratique

- Soumission d'un job batch Matlab sur le mésocentre

- Suivi et vérification du bon fonctionnement de l'exécution

Contacts et informations

- Site WEB : http://meso.univ-fcomte.fr
- Site intranet : http://mesoserver.univ-fcomte.fr
- Liste de diffusion : meso-utilisateurs@univ-fcomte.fr
  - Adhésion automatique à l'ouverture du compte
- En cas de problème : svpmeso@univ-fcomte.fr
- Citer l'utilisation / Reporter vos publications au mésocentre

Question ?