

MPI labs

Auteur: Sébastien Bigaret

Date: Octobre 2011

Sommaire

1 Configuration ssh	1
2 Open MPI	3
2.1 Commandes utiles	3
2.2 Logs & trace	3
3 Visualisation: Jumpshot	5

Ce TP se base initialement sur le tutoriel MPI LAM (<http://www.lam-mpi.org/tutorials/mpi/>) et son adaptation par l'Université de Dalhousie (Canada).

Il a été adapté par Télécom Bretagne afin de le rendre utilisable avec l'implémentation open-source «Open MPI» (<http://www.open-mpi.org>).

1 Configuration ssh

Afin de pouvoir lancer de façon transparente des processus sur plusieurs machines, Open MPI doit pouvoir se connecter automatiquement à ces machines distantes sans avoir à demander le mot de passe utilisateur.

Pour faire cela, nous allons mettre en place une authentification ssh. Nous donnons ici les étapes à suivre sur votre compte Télécom Bretagne.

1. Création de la clé. Lancez:

```
ssh-keygen
```

- validez le nom de fichier proposé par défaut (`.ssh/id_rsa`),
- choisissez une *passphrase*. Nous vous déconseillons **fortement** de laisser une passphrase vide: si votre clé était compromise, elle pourrait servir à accéder à votre compte sur toutes les machines de l'école.

2. Autoriser votre clé à accéder à votre compte:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
chmod 600 ~/.ssh/authorized_keys
```

3. Tester l'accès:

```
ssh pc-info-199
```

- Comme c'est la première fois que vous vous connectez à cette machine, ssh vous indique son *fingerprint* et vous demande de confirmer que vous souhaitez vous connecter (*Are you sure you want to continue connecting?*): répondez `yes` (voir plus bas l'URL d'un tutoriel expliquant l'importance de cette fonctionnalité de vérification de l'hôte cible).

- Une fenêtre qui s'intitule «Déverrouiller une clé privée» s'ouvre pour vous demander *passphrase* de la clé (c'est bien d'elle qu'il s'agit même si le texte de la boîte de dialogue demande un «mot de passe»): entrez la passphrase de la clé.
- Vous êtes maintenant connecté sur la machine cible: `ssh` fonctionne correctement. Tapez `exit` pour fermer la session distante.

Note

Vous avez communiqué votre passphrase à un programme spécialement dédié à cet usage, qui est lancé automatiquement lorsque vous ouvrez votre session: `ssh-agent`. Une fois que vous lui avez donné accès à votre clé, il est capable de la fournir à d'autres sessions `ssh`: vous n'avez donc à donner votre passphrase `ssh` qu'une seule fois à chaque session.

Vous pouvez le vérifier en essayant à nouveau de vous connecter sur `pc-info-199`: cette fois, rien ne vous est demandé.

Pour plus de détails sur `ssh`, nous vous invitons à consulter les nombreux tutoriaux présents sur internet, tel que http://support.suso.com/supki/SSH_Tutorial_for_Linux, et à consulter la documentation sur <http://www.openssh.org>. En particulier, le tutoriel vous donnera plus de précisions sur le mécanisme de vérification de l'hôte.

2 Open MPI

Nous utiliserons **Open MPI**, une implémentation open-source de MPI-2. L'environnement et les commandes nécessaires sont pré-installés et configurés.

La documentation se trouve à l'adresse: <http://www.open-mpi.org/doc/v1.4/>

2.1 Commandes utiles

- Compilation d'un fichier C:

```
mpicc -c prog.c
```

Création d'un exécutable:

```
mpicc prog.o -o prog -lm
```

NB: Pour chaque exercice, un fichier `Makefile` est fourni qui permet de compiler les programmes en exécutant la commande `make`.

- Exécution sur la machine locale de n copies du même programme:

```
mpirun --np <n> -- nom_du-programme
```

- Exécution sur une liste de machines de n copies du même programme:

```
mpirun --np <n> --host <host1,host2,...,hostN> -- nom_programme
```

On peut également utiliser un fichier contenant les noms des machines à utiliser avec l'option `--hostfile <hostfile>`.

Rappel: vous devez vous connectez au moins une fois sur chaque machine distante (`ssh machine`)

- À tout moment, pour connaître les processus actuellement gérés par Open MPI, utilisez la commande: `mpi-ps`.
- Si vous pensez que certains processus n'ont pas été correctement fermés, que des fichiers temporaires propres à Open MPI peuvent traîner ici et là, exécutez `mpi-clean` pour «nettoyer».
- Chacune de ces commandes ont un manuel qui peut être affiché avec la commande `man`, exemple: `man mpi-clean`.

2.2 Logs & trace

En plus de OpenMPI, MPE (MPI Parallel Environment) a été installé.

Source: <http://www.mcs.anl.gov/research/projects/perfvis/download/index.htm>

Entre autre, MPE offre des bibliothèques qui permettent de *tracer* automatiquement les appels aux fonctions de la bibliothèque MPI lors de l'exécution de programmes MPI. Selon les options de compilations fournis à `mpicc`, les traces pourront être écrites dans un fichier pour être visualisées plus tard (avec l'outil `Jumpshot` p.ex., voir plus bas) ou directement écrites sur la sortie standard.

Pour les exercices 0 à 3, le `Makefile` fournit prend cela en charge:

- `make log` compile le programme pour qu'il écrive à la fin le fichier de log au format CLOG2, qui peut être visualisé avec Jumpshot;
- `make trace` le compile pour que les traces d'exécution soient écrites sur la sortie standard;
- `make std` le compile sans MPE (compilation Open MPI standard).

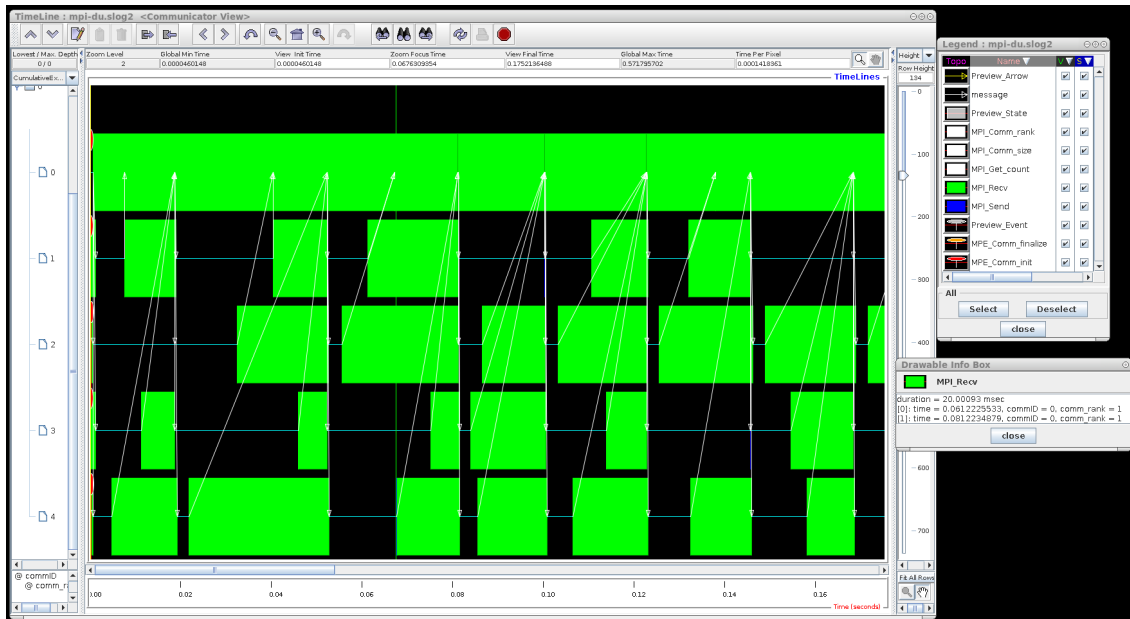
Consultez ces Makefile et observez les commandes et leurs paramètres.

Important

Dans le cas du `make log`: si votre programme ne se termine pas (c'est par exemple le cas qu'on vous demande d'examiner dans l'exercice 2), il ne produira aucun fichier de log --le log n'est écrit qu'après `MPI_Finalize()`, qui n'est pas exécuté si le programme est arrêté brutalement (par un CTRL-C p.ex.). Dans ce cas, vous pouvez utiliser l'option de trace sur la sortie standard (option `-mpitrace` de `mpecc`, ou `make trace`) pour étudier le comportement des processus.

3 Visualisation: Jumpshot

Jumpshot-4 est un utilitaire qui permet de visualiser les traces d'exécution produite par OpenMPI+MPE. Il remplace l'outil `xmpi` qui était utilisé par MPI LAM mais qui n'est maintenant plus maintenu.



Pour visualiser les traces d'un programme:

- compilez-le avec `mpecc` (ou `make log` pour les exercices 0 à 3)
- exécutez-le
- lorsqu'il se termine, il écrit automatiquement les traces dans un fichier `prg.clog2`. S'il ne se termine, vous n'aurez pas de trace: voir la section précédente.
- lancez Jumpshot:

```
/usr/local/mpe/bin/jumpshot file.clog2
```

- répondez oui à la demande de conversion en format slog2,
- la fenetre de conversion s'ouvre: cliquez sur convertir, puis sur Ok
- Jumpshot affiche la

Nous vous invitons à étudier le guide de l'utilisateur Jumpshot, à l'adresse <http://www.mcs.anl.gov/research/projects/perfvis/software/viewers/jumpshot-4/usersguide.html>, pour vous familiariser avec les fonctionnalités de l'outil.

À noter que la conversion du format clog2 vers le format slog2 peut être réalisée en commande en ligne:

```
/usr/local/mpe/bin/clog2TOslog2 mpi-du.clog2
```