

# MPI labs

**Auteur:** Sébastien Bigaret

**Date:** Juin 2016

## Sommaire

<b>1</b>	<b>Open MPI</b>	<b>1</b>
1.1	Commandes utiles	1
1.2	Logs & traces	2
<b>2</b>	<b>Visualisation: Jumpshot</b>	<b>3</b>

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 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/>

### 1.1 Commandes utiles

NB: ces commandes ne sont **pas** à exécuter tel quel, mais en fonction du contexte. Ceci est juste une liste de commandes minimales à comprendre dans un environnement openMPI.

- 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`.

## 1.2 Logs & traces

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 `mpecc`, 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).

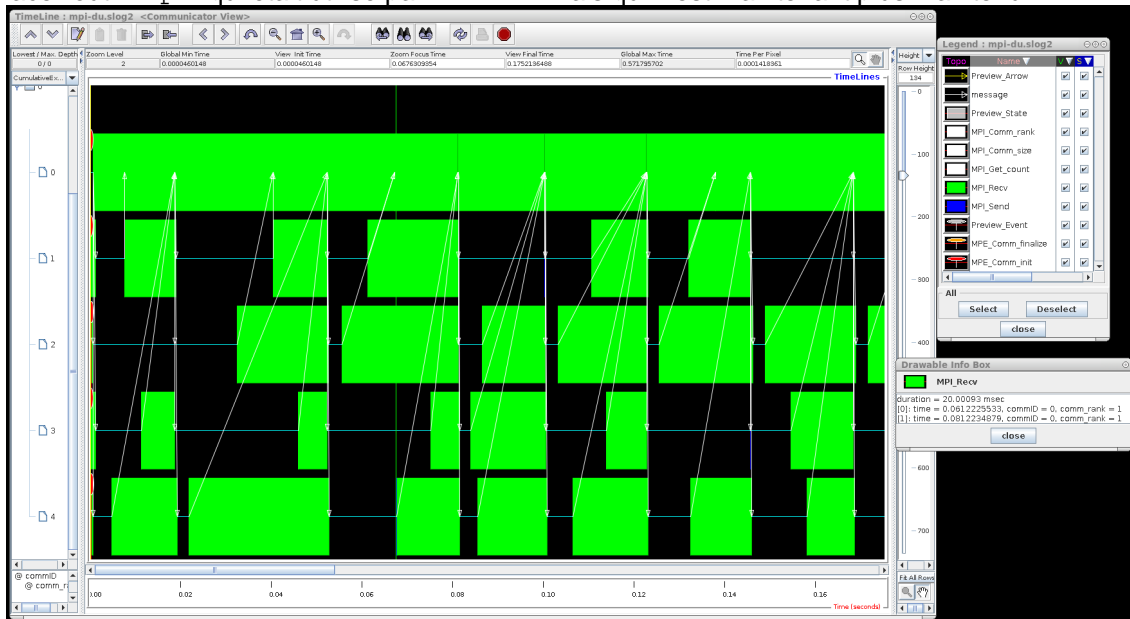
Consultez ces `Makefile` et observez les commandes et leurs paramètres.

### ***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.

## 2 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:

```
/opt/campux/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:

```
/opt/campux/mpe/bin/clog2Toslog2 mpi-du.clog2
```