TP 6 : Découverte du réseaux *Myrinet*
Programmation avec MX

Alexandre DENIS
denis@labri.fr

Systèmes et supports d'exécution — 2005–2006

L'objectif de ce TP est de découvrir le réseau haute performance *Myrinet* à travers son interface de programmation MX.

1 Infrastructures

Le TP a lieu en salle 007 — seule salle équipée en *Myrinet*. Les machines n'ont pas de système d'exploitation sur le disque dur local et sont isolées du réseau du CREMI. Pour les démarrer, nous utilisons un CD *bootable* Knoppix et nous sauvegardons nos données sur une clef USB. Vous êtes automatiquement connecté avec le compte utilisateur `knoppix` et vous avez les droits nécessaires pour `sudo`.

En fin de séance, **n'oubliez pas de sauvegarder votre travail sur la clef USB qui vous a été fournie**. La clef USB est le périphérique `/dev/uba1`; son point de montage par défaut est `/mnt/uba1`. Vous pouvez également, par précaution, recopier vos données par `scp` vers votre compte au CREMI. N'oubliez pas qu'à chaque reboot, tout ce qui n'a pas été sauvegardé est perdu.

En standard, les machines de la salle 007 n'ont pas de connexion inernet vers l'extérieur, ni même vers les autres machines du CREMI. Pour autoriser la connexion, ouvrez un navigateur vers n'importe quelle adresse web. La fenêtre d'authentification s'affiche. Répondez par votre login et mot de passe CREMI. **Gardez cette fenêtre ouverte** pour garder la connexion internet active.

2 Présentation de *Myrinet*

Myrinet est un réseau haute performance basé sur des cartes de communication équipées d'un processeur programmable (le *LANai*) et de mémoire SRAM embarquée. Le réseau est entièrement commuté, *full duplex*, et les communications sont fiables (sans perte). Plutôt que de programmer directement la carte *Myrinet* et son *LANai*, nous l'utiliserons au travers de son pilote MX qui contient un *firmware* complet pour le *LANai*, un module noyau, et une bibliothèque utilisateur appelée `myriexpress`.

La documentation complète de MX est disponible en ligne sur `http://www.myri.com/scs/MX/doc/`

Une copie de PM2 et de la documentation MX est préchargée dans le répertoire `Ressources/` sur la clef USB qui vous été remise.

Myrinet en salle 007

Le pilote MX n'est pas chargé par défaut. Il dépend d'une configuration dynamique qui ne peut pas être présente sur un CD Knoppix *gravé* (en lecteure seule, donc). Il faut donc initialiser la configuration de MX, puis le charger à la main.

1. Commencez par passer super-utilisateur. Toutes les commandes suivantes doivent être exécutées en tant que `root`.
2. Par défaut, une machine démarré avec Knoppix s'appelle `knoppix`. Pour éviter que toute la salle 007 porte le même nom, changez le nom de votre machine avec la commande `hostname` (donnez-lui par exemple votre nom, pour s'y retrouver...)
3. Créez une configuration MX en lançant la commande :

   ```
   /opt/mx/sbin/mx_local_install
   ```
4. Chargez le pilote MX en lançant la commande :

   ```
   /etc/init.d/mx start
   ```
5. Contrôlez la configuration de MX avec la commande :

   ```
   /opt/mx/bin/mx_info -v
   ```

   Votre machine ainsi que toutes celles qui ont déjà chargé le pilote MX doivent apparaître, avec leur nom correct (qui est de la forme « *nom de machine* :0 », le « :0 » signifiant qu'il s'agit de la carte Myrinet numéro 0 de la machine donnée.
6. En cas de problème de configuration, vous pouvez :
   – Corriger le nom de votre machine dans un pilote MX déjà chargé. Utilisez pour celà la commande `/opt/mx/bin/mx_hostname -n` *nom*
   – Mettre à jour le cache des tables de routage (s'il vous manque des machines dans `mx_info` ou si les noms ne sont pas à jour). Utilisez la commande `/opt/mx/bin/mx_hostname -c`
   – En cas de problème insoluble, rechargez le pilote MX : `/etc/init.d/mx restart`
7. Lancez le serveur `ssh` (si ce n'est pas déjà fait) :

   ```
   /etc/init.d/ssh start
   ```

Vous n'avez ensuite plus besoin de la console `root`. Tous vos travaux se font sous le nom d'utilisateur `knoppix`.

3 Travail à faire

Commencer par lire attentivement au moins les sections 1 (*overview*) et 2 (*Concepts*) de la documentation de MX 1.0.

Note :

– pour compiler un code qui utilise MX, n'oubliez l'argument `-I/opt/mx/include/` sur la ligne de commande de compilation. L'en-tête à inclure dans vos programmes est `myriexpress.h`.
– pour l'édition de lien, n'oubliez pas les arguments `-L/opt/mx/lib -lmyriexpress` sur la ligne de commande d'édition de liens.

3.1 Envoi simple

Ecrire un programme émetteur et un programme récepteur s'échangeant un message de 10 octets (taille fixe et connue par les deux parties).

3.2 Ping-pong

Étendre le programme précédent pour réaliser un ping-pong, de façon à pouvoir mesurer les performances.

3.3 Benchmark

– Modifier le programme de façon à envoyer des messages de taille croissante (les puissances de 2, de 4 octets à 4 Mo par exemple), et mesurer le temps de transfert pour chacune des tailles. Tracer ensuite la courbe de débit en fonction de la taille des messages à l'aide de `gnuplot`.
– Comparer les performances obtenues en utilisant les primitives de communication synchrones et celles de communication asynchrones.