

Installation d'un cluster à déploiement rapide openMosix-LTSP

Table des matières

Installer LTSP.....	1
Installer les outils nécessaires.....	1
Créer un noyau openMosix.....	2
Télécharger les sources d'un noyau linux.....	2
Télécharger le patch openMosix.....	3
Patcher les sources avec le patch openMosix.....	3
Compiler le noyau avec les options LTSP.....	3
Rendre le noyau bootable par le réseau.....	5
Configurer le serveur.....	6
Copier le noyau et les modules	6
Configurer PXE	6

L'objectif du cluster à déploiement rapide est de créer un cluster auquel on puisse ajouter un nœud de calcul très aisément. Dans notre cas, le but est de pouvoir ajouter une machine au cluster simplement en la faisant démarrer sur le réseau et sans aucune configuration.

Le principe est de combiner les technologies de cluster openMosix (voir la doc "Installer un nœud openMosix sous Debian") à la technologie de boot réseau LTSP de façon à créer un cluster à déploiement rapide openMosix-LTSP.

Pour intégrer une nouvelle machine dans un cluster openMosix, il suffit de la démarrer sur un noyau Linux patché openMosix. Pour faire booter des machines en nœuds openMosix via le réseau, il suffit donc d'installer un serveur LTSP qui propose un noyau patché openMosix.

Installer LTSP

Avant tout, il faut installer LTSP et les différents outils nécessaires à son fonctionnement. (voir Installation d'un serveur LTSP)

Installer les outils nécessaires

Il faut placer dans l'arborescence ltsp (/opt/ltsp/i386) les outils nécessaires au lancement du nœud openMosix. Tout d'abord, pour pouvoir exécuter les outils

nécessaires sur le client, il faut installer le package "ltsp_local_apps". Ce package est téléchargeable sur le site www.ltsp.org. Dans la partie "Downloads" téléchargez le package ltsp_local_apps-3.0.0-i386.tgz. Décompactez le et installez le :

```
cd ~/
tar -zxvf ltsp_local_apps-3.0.0-i386.tgz
cd ./tsp_local_apps
./install.sh
```

Ensuite, il faut installer les outils openMosix. Pour cela, il faut se rendre sur la page de Mr Daniau : <http://www.lpmo.edu/~daniau/ltsp-mosix/> et télécharger les deux packages : ltsp_mosix_core-1.0beta4.tar.gz et ltsp_openmosix_userland_0.3.4.tar.gz.

Les décompresser et les installer dans l'ordre :

```
cd ~/
tar -zxvf ltsp_mosix_core-1.0beta4.tar.gz
cd ./ltsp_mosix_core-1.0beta4
./install.sh
cd ..
tar -zxvf ltsp_openmosix_userland_0.3.4.tar.gz
cd ./ ltsp_openmosix_userland_0.3.4
./install.sh
```

Créer un noyau openMosix

Télécharger les sources d'un noyau linux

Plusieurs versions du noyau linux peuvent être utilisables. Ici nous utiliserons la dernière version stable : la 2.4.22. Les sources du noyau sont téléchargeables sur le site www.kernel.org. Elles se trouvent sous la forme d'une archive compressée tar.gz nommée : linux-2.4.22.tar.gz qu'il faut placer dans le répertoire /usr/src. Sous Debian, les sources sont téléchargeables plus aisément avec apt-get :

```
apt-get install kernel-source-2.4.22
```

Cette commande place directement l'archive dans le bon répertoire : /usr/src/kernel-source-2.4.22.tar.bz2.

Il faut ensuite décompresser cette archive et la nommer linux-2.4.22 :

```
cd /usr/src
tar -zxvf linux-2.4.22.tar.gz
ou
cd /usr/src
bunzip2 kernel-source-2.4.22.tar.bz2
```

```
tar -xvf kernel-source-2.4.22.tar
ln -s kernel-source-2.4.22 linux-2.4.22
```

Télécharger le patch openMosix

Les patches openMosix sont téléchargeables sur le site www.openmosix.org dans la section download (http://sourceforge.net/project/showfiles.php?group_id=46729&release_id=200888). Pour le noyau 2.4.22, le fichier qui nous intéresse se nomme openMosix-2.4.22-1.bz2. Il faut le télécharger, le placer dans le répertoire /usr/src et le décompresser.

```
bunzip2 openMosix-2.4.22-1.bz2
```

Patcher les sources avec le patch openMosix

```
cd /usr/src
patch -p0 < openMosix-2.4.22-1
```

Compiler le noyau avec les options LTSP

Pour rendre le noyau bootable, il faut télécharger un petit utilitaire appelé ltsp_initrd_kit sur le site de LTSP (www.ltsp.org). Dans la section download, téléchargez l'archive : ltsp_initrd_kit-3.0.12-i386.tgz. Puis décompressez la :

```
tar -zxvf ltsp_initrd_kit-3.0.12-i386.tgz
```

Ce package propose un fichier de configurations pour le noyau linux 2.4.22 (config-2.4.22-ltsp-1). Il faut utiliser ce fichier pour construire un nouveau noyau LTSP (préalablement patché openMosix).

```
cp ~/ltsp_initrd_kit/config-2.4.22-ltsp-2 /usr/src/linux-2.4.22/.config
```

```
cd /usr/src/linux-2.4.22
```

```
make xconfig
```

Cette dernière commande fait apparaître une fenêtre de configuration du noyau. Elle nécessite un utilitaire appelé "wish" et vous fournira un message d'erreur : "make: wish: Command not found" si il n'est pas présent sur la machine. Sous debian, cet utilitaire est présent dans un package appelé "tk8.4" et installable par apt-get :

```
apt-get install tk8.4
```

Si toutefois vous n'arrivez toujours pas à afficher la fenêtre de configuration, la commande **make oldconfig** fait la même chose mais en mode texte.

Dans cette phase, il faut sélectionner les options concernant openMosix. Voici une brève description de ces options :

CONFIG_MOSIX	openMosix process migration support	C'est l'option qui active le cluster openMosix.
CONFIG_MOSIX_TOPOLOGY	Support clusters with a complex network topology	Permet de créer un cluster dont les nœuds sont sur un réseau complexe à travers des routeurs .
CONFIG_MOSIX_MAXTOPOLOGY	Maximum network-topology complexity to support (2-10)	Nombre maximum de routeurs à traverser dans un réseau complexe.
CONFIG_MOSIX_SECUREPORTS	Stricter security on openMosix ports	
CONFIG_MOSIX_DISCLOSURE	Level of process-identity disclosure (0-3)	
CONFIG_MOSIX_FS CONFIG_MOSIX_DFSA	openMosix File-System	Utilisation de OMFS et DFSA, système de fichier spécifique à openMosix destiné à donner une vision du même système de fichier sur chaque nœud du cluster.
CONFIG_MOSIX_PIPE_EXCEPTIONS	Poll/Select exceptions on pipes	
CONFIG_openMosix_NO_OOM	Disable OOM Killer	Cette option désactive la destruction d'applications qui s'exécutent hors de leur espace de mémoire (Out Of Memory). Les applications tournant hors de leur espace mémoire ne seront pas détruites.
CONFIG_MOSIX_LOADLIMIT	Load Limit	Option récente qui permet de définir une charge CPU limite pour un nœud particulier. Quand la limite est atteinte, le nœud n'accepte plus de migrations de processus d'autres nœuds.

Le plus simple est de répondre oui pour toutes et de laisser les valeurs par défaut.

Il faut ensuite lancer la compilation :

```
make clean && make oldconfig && make dep && make bzImage  
&& make modules && make modules_install
```

Cette phase est assez longue. Patience ...

Rendre le noyau bootable par le réseau

Pour rendre le noyau bootable, il faut le modifier grâce à un utilitaire nommé "mknbi" ce qui signifie "MaKe Net Bootable Image". Cet utilitaire est disponible sur les miroirs Debian :

apt-get install mknbi

Dans le package ltsp_initrd_kit, on trouve un utilitaire nommé "buildk" qui effectue automatiquement le travail et "taggue" le noyau avec mknbi.

cd ~/ltsp_initrd_kit/

./buildk

Buildk va vous demander si vous utilisez des cartes réseau PCI/ISA ou PCMCIA (pour les ordinateurs portables). Dans notre cas nous utilisons bien entendu des cartes PCI. Il va aussi vous demander l'emplacement de vos sources. Enfin ,il va vous demander la taille et le nombre de nodes pour le système de fichier initrd. Il suffit de laisser les valeurs par défaut :

Labo2:/home/olivier/ltsp_initrd_kit# **./buildk** ↵

You can prepare a kernel for the following type of workstation:

- 1 - PCI/ISA network card
- 2 - PCMCIA (laptop) network card

Enter option (1-2) [1]: ↵

Kernel source directory [/usr/src/linux-2.4.22-ltsp-2]: **/usr/src/linux-2.4.22** ↵

Size of initrd filesystem [2500]: ↵

Number of inodes to allocate in initrd filesystem [300]: ↵

#####

#####

/usr/src/linux-2.4.22/arch/i386/boot/bzImage

2500+0 records in

2500+0 records out

mke2fs -F -q -m 0 -N 300 initrd.ltsp 2500

mke2fs 1.27 (8-Mar-2002)

Copying the root tree

1038 blocks

2524 blocks

88 blocks

Checking size

Filesystem 1k-blocks Used Available Use% Mounted on

/home/olivier/ltsp_initrd_kit/initrd.ltsp

```
2457    1982    475 81% /mnt/initrd
```

```
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/home/olivier/ltsp_initrd_kit/initrd.ltsp
                304    147    157  49% /mnt/initrd
```

```
gziping initrd
running mknbi-linux
-rw-r--r--  1 root   root   1811968 Jan  7 12:53 /
home/olivier/ltsp_initrd_kit/vmlinuz-2.4.22-openmosix-2
```

```
Labo2:/home/olivier/ltsp_initrd_kit#
```

Une fois le script terminé avec succès, il aura créé deux fichiers dans le répertoire `ltsp_initrd_kit` : **bzImage-2.4.22-openmosix-2** et **initrd-2.4.22-openmosix-2.gz**

Configurer le serveur

Copier le noyau et les modules

Il faut copier le nouveau noyau `ltsp+openMosix` "taggué" avec `mknbi` dans le répertoire `tftpboot` afin qu'il soit téléchargeable par *tftp* :

```
cp ./bzImage-2.4.22-openmosix-2 /tftpboot
cp ./initrd-2.4.22-openmosix-2.gz /tftpboot
```

Et copier les modules dont le noyau aura besoin dans le système de fichier LTSP :

```
cp -r /lib/modules/2.4.22 /opt/ltsp/i386/lib/modules
```

Configurer PXE

Il faut configurer PXE dans le répertoire `tftpboot` pour qu'il prenne en compte le nouveau noyau. Pour cela, il faut modifier le fichier `/tftpboot/pxelinux.cfg/default` de manière à ce qu'il ressemble à cela :

```
prompt=0
label linux
    kernel bzImage-2.4.22-openmosix-2
    append init=/linuxrc rw root=/dev/ram0
    initrd=initrd-2.4.22-openmosix-2.gz
```

