

Utiliser le client classique de BitTorrent avec screen

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 mai 2008

<https://www.bortzmeyer.org/screen-bittorrent.html>

Les téléchargements de fichiers avec le protocole BitTorrent peuvent prendre des jours, voire des semaines. Laisser un logiciel ouvert pendant ce temps n'est pas possible pour une machine de maison ordinaire, qui est souvent éteinte. Même sur un serveur allumé en permanence, cela bloque un terminal par téléchargement, si on utilise le client traditionnel, `btdownloadcurses`. Il existe plusieurs solutions à ce problème, la mienne combine un des logiciels du client traditionnel, `btlaunchmanycurses` et le programme `screen` <<https://www.bortzmeyer.org/screen.html>>.

Une des solutions est évidemment d'utiliser un autre programme, puisqu'il existe beaucoup de clients BitTorrent, tous plus ou moins compatibles (en dépit de l'absence d'une norme écrite du protocole). Je n'ai jamais eu de succès avec Azureus (`core dumped`) qui, contrairement à ce que prétend le discours marketing, comme presque tous les programmes Java, ne fonctionne pas avec n'importe quel environnement Java mais uniquement avec celui de Sun (qui était non-libre jusqu'à une époque très récente). De même, bien que m1Donkey aie, en théorie, un support BitTorrent, les téléchargements ne démarrent jamais, dans mon cas (et ce logiciel semble en voie d'abandon, avec une communauté qui est devenue inactive). Quant à TorrentFlux, il nécessite un serveur Web, PHP, etc.

Retour au client BitTorrent original, écrit en Python. Il comprend de nombreux programmes différents dont un qui fait presque tout ce dont j'ai besoin : `btlaunchmanycurses` prend en argument un répertoire, ouvre tous les fichiers « torrent » de ce répertoire et lance simultanément les opérations. Ajouter un nouveau « torrent » à la liste est aussi simple qu'un `wget` dans ce répertoire (par exemple `wget ftp://ftp.netbsd.org/pub` pour récupérer le CD de NetBSD).

Pour libérer le terminal et pour que `btlaunchmanycurses` s'exécute tranquillement en arrière plan, la meilleure solution est d'utiliser le programme `screen` <<https://www.bortzmeyer.org/screen.html>> qui permet de lancer le programme en arrière-plan, le reprendre à volonté (y compris depuis une autre machine), le détacher à nouveau, etc. Le script shell attaché fait exactement cela :

```
#!/bin/sh

NAME=BitTorrent
DIR=${HOME}/Downloads
BT=btlaunchmanycurses
BT_OPTIONS="--ipv6_enabled 1 --max_uploads 3 --max_upload_rate 8 --max_download_rate 85"

cd ${DIR}

# Starts "screen" detached, with a command to run
screen -S ${NAME} -d -m ${BT} ${BT_OPTIONS} .

# The limit of this approach is that screen disappears when ${BT} is
# over. If ${BT} crashes, you lose the error messages :-))

# The alternative would be to launch screen without a command (just -d
# -m) and then to issue a "screen -X -r ${NAME} exec ${BT} ...". But
# this is ignored, probably because a bug in screen (see
# <https://savannah.gnu.org/bugs/index.php?18882>)

# Another solution (thanks to Thomas Montfort to bring it to me) is to
# use the 'zombie kr' command in ~/.screenrc.
```

Il ne reste plus qu'à faire en sorte que ce script, `start-bittorrent`, soit lancé automatiquement au démarrage de la machine. Le client BitTorrent sait reprendre les transferts interrompus. Encore faut-il le mettre en route. Pour éviter d'utiliser le compte root, on peut exploiter un service des crons modernes comme Vixie Cron qui permet de lancer une commande au moment de certains événements comme le démarrage. Ainsi, j'ai dans ma `crontab` :

```
@reboot bin/start-bittorrent
```

Je dois des remerciements à Erwan Arzur pour m'avoir indiqué `btlaunchmanycurses` et à Erwan David pour son aide sur cron.