IFT630 - Projet

# Le webserver

J’ai commencé par faire un webserver en C++. Celui-ci écoute à un port et fait du « load balancing » vers des « backend servers », qui sont des applications web ruby. Les backend servers sont connectés à travers des sockets aussi, ce qui permet d’avoir des applications construite avec n’importe quel langage et ça marche. Je voulais aussi faire que le webserver puisse servir les « static assets » (javascript, css, images, etc..) directement sans passer par les applications.

## Ce qui est fait

J’ai la partie connexion de fait (avec le frontend et les backend servers). J’ai aussi fait des fonctions pour gérer les headers. Les connexions du frontend sont threadées.

## Problèmes

J’ai eu des problèmes avec l’encoding de mes strings ce qui m’a complètement démoralisé, car le projet était inutilisable d’un navigateur web. Par contre, il marche avec un curl en command-line.

## Dossier

/bin Scripts pour compiler et exécuter le projet

/examples Exemple d’utilisation du webserver

 /example\_1 Application Ruby pour la démo

/src Code du webserver

## Exécution

Pour exécuter le projet, il faut Ruby 1.9 installé avec les gems requis (voir Gemfile). Ensuite il faut partir 2 webapps, une sur le port 3000 et une sur le port 3001 et démarrer l’example\_1 du webserver.

# Download accelerator

Comme je n’ai pas réussi à accomplir ce que je voulais avec le webserver j’ai fait un download accelerator pour avoir quelque chose de fonctionnel. Celui-ci est codé en ruby et il est fonctionnel. Il effectue plusieurs connexions vers la même adresse et démarre plusieurs threads. Chaque threads s’occupe d’aller chercher une partie spécifique de l’objet et ensuite l’écrit dans un fichier sur le disque.