



**POLYTECHNIQUE
MONTRÉAL**

**UNIVERSITÉ
D'INGÉNIERIE**

LABORATOIRE 1
TRAITEMENT D'IMAGES PAR RÉSEAU

INF3405 – RÉSEAUX INFORMATIQUES
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET LOGICIEL - AUTOMNE 2020

Soumis à
Mehdi Kadi
Bilal Itani
Liliane-Caroline Demers

Remis par
Thierry Beiko 1843222
Michael Chehab 1849421
Louis Popovic 1844807

26 octobre 2020
École Polytechnique de Montréal

Introduction

Le modèle client/serveur décrit la manière par laquelle un serveur peut offrir des ressources ou des services à un ou plusieurs clients. La majorité des serveurs sont en mesure de répondre aux besoins de plusieurs clients en même temps. Le présent laboratoire traite des échanges d'un serveur avec ses clients par l'entremise d'un réseau informatique.

Afin de communiquer, les clients envoient une requête de connexion au serveur qui peut être acceptée ou refusée. Si la connexion est acceptée, le serveur établit et maintient la connexion au client en suivant un protocole spécifique, connu des deux partis. Les clients et les serveurs échangent des messages selon un modèle de messagerie requête-réponse. Le client envoie une demande et le serveur renvoie une réponse. Tous ces échanges se font par l'entremise de sockets, soit des ports de communications, propres à une paire client/serveur.

Pour permettre au serveur de servir plusieurs clients en même temps le serveur utilise des threads afin de traiter les images des différents clients en parallèle.

Présentation du travail

La fonction principale du serveur est d'écouter les connexions des clients. Une fois qu'une connexion est établie, le serveur lance un processus sur une de ses threads pour servir ce client. Il peut ensuite continuer à écouter d'autres connexions.

Le processus exécuté est responsable de toute la logique du côté serveur. Dans ce processus, le serveur vérifie si le nom d'utilisateur et le mot de passe du client concordent, il enregistre les nouveaux utilisateurs, il reçoit une image, la traite et la renvoie.

Du côté client, la fonction principale s'assure qu'il est possible de se connecter à l'adresse IP fournie, établit une connexion, demande et envoie au serveur les informations du client, affiche la réponse du serveur, envoie l'image et reçoit finalement l'image traitée.

Le serveur et les clients se communiquent les images (pré et post traitées) à l'aide de *byte array*. La fonction *process()* qui traite l'image utilise des *BufferedImage*. Finalement, l'image d'origine et celle traitée sont enregistrées sur la machine du client. Nous avons créé des fonctions de conversions. Tout d'abord, on convertit l'image originale en *byte array* (*jpegToByte()*) du côté du client. Après avoir été transmis au serveur, le *byte array* est converti en *BufferedImage* (*byteToBufferedImage()*) et elle est traitée par la fonction de Sobel. Ensuite, la *BufferedImage* est re-converti en *byte array* (*bufferedImageToByte()*) pour finalement être renvoyé au client. Après avoir reçu le *byte array*, le client le re-converti en *jpeg* (*byteToJpeg()*) avant de la sauvegarder sur la machine.

Difficultés rencontrées et solutions apportées

Au début, nous essayions de communiquer les images directement en format *jpg* ou *BufferedImage* et nous avons beaucoup de misère. Après avoir compris grâce à des recherches internet que la communication entre socket était beaucoup plus facile avec des *byte array* nous avons été en mesure de finir le code pour la communication. Nous avons aussi dû implémenter une logique pour envoyer la taille du *array* avant de communiquer l'*array* en tant que tel. En effet, lorsqu'on envoie des *byte array* entre socket, il faut réserver l'espace mémoire.

Critiques et Améliorations

En termes de réseautique, nous croyons que ce laboratoire ne permet pas énormément de mettre en application les connaissances théoriques vues en cours. En effet, le seul parallèle entre le cours théorique et ce laboratoire sont les adresses IPs et les ports. De plus, tout ce que nous avions à faire était d'appeler une fonction *connect()* pour établir la connexion.

Aussi, selon nous, le laboratoire aurait pu être un peu plus demandant. En effet, avec deux mois pour faire ce laboratoire, il aurait pu y avoir plus de tâches (ex. : avoir à faire des fonctions de Sobel pour le traitement d'images).

Conclusion

Même si nous n'avons pas appliqué énormément de connaissances vues en cours, le laboratoire nous a permis d'approfondir nos connaissances sur java et sur l'utilisation des sockets.