



INF3405 – Réseaux Informatiques

TP1 : Projet en réseaux informatiques Gestionnaire de fichier

Hiver 2021

1947701 - Simon Kaplo

1956802 - Dawut Esse

1992609 - Edwin Chiem

Groupe 5

Vendredi 26 février 2021

Soumis à : Liliane-Caroline Demers

Introduction

En tant qu'étudiants à Poly, nous utilisons beaucoup les services de stockage d'informations sur les nuages. La majorité de ces service charge de l'argent après une certaine limite d'information stocker. Cela peut être très coûteux pour un étudiant et donc nous avons eu l'idée de développer notre propre serveur de stockage d'informations.

Le développement de notre application client-server a pour objectif de nous permettre de stocker n'importe quel type de fichier sur un serveur de stockage. L'application permet à plusieurs clients de se connecter au serveur en même temps.

Présentation

Pour nous permettre de stocker de l'information sur un serveur, nous avons développé deux applications, un serveur et un client.

La première application le serveur a pour objectif d'établir une communication avec le client qui permet a plusieurs clients de se connecter en même temps, de traiter les commande du client et d'envoyer des informations du serveur au client.

La deuxième application, le client a pour objectif d'établir une communication avec le serveur qui permet l'envoyer au serveur des commande de l'utilisateur et des information du client.

a) Connexion serveur-client en utilisant des socket.

La communication entre le serveur et le client est fait en utilisant des ServeurSocket qui établis un lien de communication entre un client et un serveur. Le socket doit être lié à un port pour identifier la destination de l'envoi au niveau du TCP. Nous avons aussi besoin de fournir une adresse IP, cela est primordial pour une connexion entre plusieurs clients et un serveur. En Java l'adresse IP et le port vont être utilisés pour créer une nouvelle adresse socket qui va être lié à une instance de ServerSocket. Mais avant former cette adresse, il faut s'assurer de la validité de l'adresse IP et du port.

b) Verification de l'adresse IP et du port.

Au lancement de l'application client ou serveur les deux demande à l'utilisateur de premièrement fournir une adresse IP. Tant que l'adresse n'est pas dans le format valide la console affiche un message d'erreur et redemande à l'utilisateur d'entrer une adresse IP. Ensuite, la console demande de fournir le nombre du port qui est entre 5000 et 5050. Tant que le port n'est pas dans cet intervalle la console affiche un message d'erreur et redemande à l'utilisateur d'entrer un port.

Cependant, ces vérifications n'empêchent pas une erreur de connexion avec le socket. Notamment même si l'adresse IP est du bon format elle peut être invalide ou aussi dans le cas ou le port est déjà utiliser la connexion échoue. Tant qu'une connexion n'a pas étai correctement établis la console affiche un message d'erreur et redemande à l'utilisateur d'entrer une adresse IP et le port.

c) Serveur et ClientHandler.

Finalement après une connexion valide sur le serveur, une nouvelle instance de ClientHandler est crée qui sert comme gestionnaire du client. ClientHandler est une classe construite avec le socket passer par serveur et le nombre du client. Une nouvelle instance du gestionnaire de client est créée pour chaque client connecté. ClientHandler hérite de Thread ce qui permet que les gestionnaires de client soient exécutées sur plusieurs threads simultanément et par conséquent permet un accès en parallèle des clients.

d) Client et ClientHandler .

Les get de input et output stream du socket permet une communication entre client et son gestionnaire qui est dans le serveur. Le client et son gestionnaire peu communiquer en imprimant dans l'output stream un message a envoyer. Cela permet au client d'envoyer au gestionnaire les informations necessaire pour le traitement de la commande et permet au gestionnaire d'envoyer une réponse au client après que le traitement de la tache soit fini. Ils peuvent aussi communiquer en lisant dans l'input stream

le message envoyer. Cela permet au client de lire la reponse du gestionnaire et permet au gestionnaire de lire la commande taper dans le client.

Difficultés rencontrées

Il était difficile au début de clairement identifier comment implémenter le serveur de manier que plusieurs clients peuvent accéder simultanément. Mais après qu'ont ces familiariser avec comment fonctionne les threads et sockets dans Java l'implémentation d'un gestionnaire de client est devenu plus claire.

Critiques et améliorations

Les consignes pouvaient être plus précises et il serait intéressant si l'équipe pouvait choisir quelle langue de programmation utiliser.

Conclusion

Le laboratoire nous a aidés à mieux comprendre le concept de socket et la gestion de communication entre un client et un serveur.