R3.09

Intro : Ce document est un document de réponse qui va permettre de répondre à la problématique ainsi que de donner des nouvelles fonctionnalités que l’on pourrait rajouter qui ne sont pas forcément dans ce cahier des charges.

Durant cette SAE, nous devions réaliser un cluster de calcul en python. Pour la réaliser nous avons utilisé les différentes méthodes à la réalisation de ce projet à savoir, les threads, la gestion des exceptions, les interfaces graphiques et les sockets.

Réponse :

Cahier des charges du sujet simples : j’ai d’abord codé un server, celui-ci peut remplir plusieurs fonctionnalités.

Il est capable de recevoir des requêtes clients sur un port spécifié au moment du démarrage. S’il est déjà en train d’exécuter un programme pour un autre client, il envoie un message indiquant qu’il doit se connecter à un autre serveur ou qu’il est en attente.

S’il n’y a pas de programme qui s’exécute alors le serveur accepte de le recevoir, le compiler si nécessaire, l’exécuter et renvoyer les résultats obtenus.

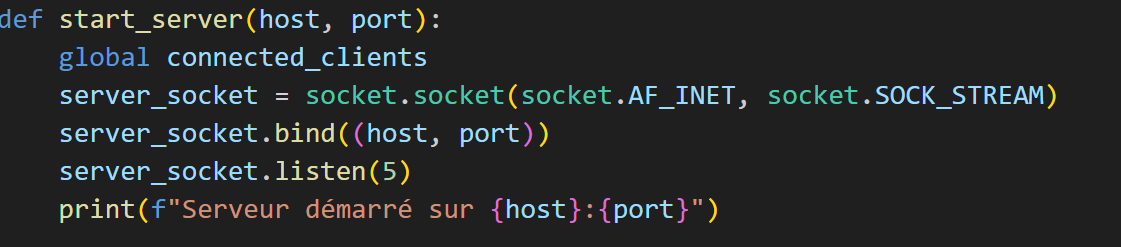
Puis j’ai codé un client, il propose une interface graphique permettant à l’utilisateur de sélectionner un programme à envoyer au serveur. L’interface permet aussi à l’utilisateur de spécifier le nom de la machine, l’adresse IP et le port de la machine visée. Une fois le programme envoyé, si le serveur rejette le programme, un nouveau nom/IP/port devra pouvoir est proposé sans avoir à redémarrer.

Si le serveur accepte, le résultat est affiché sur l’interface.

Architecture de la solution :

L’architecture est simple, le serveur est sur une adresse ip fixe que vous choisissez. Ce serveur ecoute sur un port choisi, le client viens se connecter sur le serveur et viens envoyer les fichiers.

Code du serveur :



Partie connexion et déconnexion des clients.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

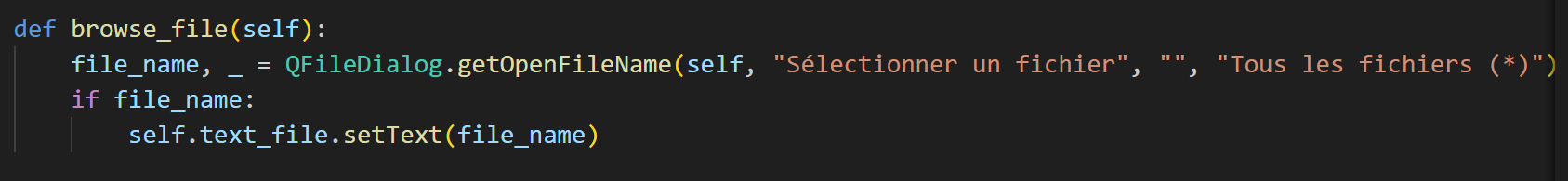
Création d’un fichier temporaire pour stocker les codes.

Une image contenant texte, capture d’écran

Description générée automatiquement

Partie permettant de limiter le nombre de clients connectés en même temps.

Code du client :



Permet de choisir les fichiers à envoyer.

Une image contenant texte, capture d’écran

Description générée automatiquement

Permet d’envoyer des fichiers au serveur.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Partie graphique qui permet ensuite d’être remplie afin de se connecter au serveur.

Conclusion :

La problématique de cette SAE était : Comment mettre en œuvre une solution de cluster de calcul.

Bien évidemment, elle avait pour but de nous faire découvrir et apprendre le code pour des serveur et des clients. Mais si nous nous demandons maintenant à l’echelle professionnel on se rends compte que c’est possible de mettre en œuvre notre cluster de calcul mais certaine question se pose, par exemple coté sécurité : on pourrait rajouter un mot de passe pour se connecter au serveur afin que les personnes malveillantes ne puissent pas s’y connecter. De plus nous pourrions modifier le code afin que le serveur n’exécute pas des codes malveillants.

Une autre fonctionnalité serait d’y ajouter un serveur maitre ainsi que plusieurs serveur esclave afin de rediriger les différents clients qui ne peuvent pas accéder au serveur maitre sur d’autre serveur.

Je pense que cette SAE est importante dans le domaine professionnel car il est toujours bon d’avoir des bases dans le domaine du code dans notre corp de métiers.

Pour finir du coté personnelle, j’ai trouvé cette SAE assez complexe par mes lacunes dans le domaine de code. J’y ai passé entre 30 et 35h. Mais j’ai bien aimé cette SAE malgré tout, j’ai trouvé certain point peu clair mais après avoir discuté avec mes camarades j’ai pu passer au-dessus des doutes.