Rapport projet AD: Traitement d'images en distribué

Beuque Eric Cornevaux Sébastien

3 février 2009

Table des matières

1 Introduction

Dans le cadre du module d'Algorithmes Distribués, les étudiants devaient réaliser un environnement de calcul distribué permettant d'appliquer des traitements en parallèle sur une image.

De plus, il fallait mettre en place la terminaison des serveurs en applicant l'algorithme de Safra découvert en cours.

Ce rapport revient sur notre travail, principalement en ce qui concerne l'implémentation de la gestion des ressources sur l'anneau de serveurs ainsi que celle de la terminaison.

2 Plate-forme : serveurs en anneau

La plate-forme peut être constituée de plusieurs serveurs qui seront connectés entre eux en formant un anneau. Une fois l'anneau entièrement initialisé, les serveurs peuvent envoyer des jetons de différents types sur l'anneau afin d'effectuer des traitements en fonction des demandes qu'ils reçoivent des clients.

2.1 Initialisation d'un serveur

Au démarrage, un serveur prend d'abord comme paramètres le nom de la machine, le port RMI et le nom du serveur. Il utilise avant tout ces informations pour s'enregistrer dans le RMIRegistry. Il crée ensuite un thread qui fera office de serveur de jeton. Ce serveur de jeton possède une file de jeton et sera chargé de traiter les jetons dans cette file ainsi que de transmettre un jeton à son serveur suivant. Ensuite le serveur instancie aléatoirement entre 2 et 4 threads de traitement qui permette d'effectuer les traitements des morceaux d'images. Enfin, une confirmation est demandée afin d'effectuer la connexion avec le serveur suivant dont les informations de connexion ont été passées en paramètre. Ceci requiert d'avoir lancé au préalable le serveur suivant.

2.2 Type d'action sur le serveur

2.2.1 Demande de traitement

Lorsqu'un client se connecte à un serveur, il doit d'abord effectuer une demande de traitement. Le client lui transmet ainsi ses informations de connexion pour qu'un serveur puisse lui répondre. Le serveur contacté envoi alors un jeton de demande de traitement *JetonDemandeTraitement* sur l'anneau. Quand un serveur accepte la demande de traitement, il en informe le client en transmettant ses informations de connexion. Le serveur se met alors en attente que le client lui transmette son image. Pendant ce temps, le serveur n'accepte plus de demande de traitement.

2.2.2 Envoi d'une image

Lorsqu'un serveur a notifié à un client qu'il acceptait sa demande, ce dernier lui envoi l'image complète du traitement à effectuer, ainsi que le type de traitement qu'il désire sur l'image. Le serveur se charge alors de découper l'image en morceau à sa réception. Chaque morceau est placé dans un jeton *JetonTraitement* qui contient aussi les informations du client pour le retour des morceaux traités, ainsi que le type de filtre voulu. Ces jetons sont alors envoyés sur l'anneau. Une fois que tous les jetons ont été transmis, le serveur est libéré et peut de nouveau accepter une demande de traitement.

2.2.3 Demande de terminaison

Un administrateur peut initier une demande de terminaison de la plate-forme. Lorsqu'un serveur reçoit une telle demande, il envoit un jeton de terminaison *JetonTerminaison* dans l'anneau qui contient les informations du serveur initiateur, et des informations pour l'algorithme de Safra. Ce jeton suit alors l'algorithme de Safra pour détecter la terminaison des serveurs.

2.3 Traitement des jetons

Lorsqu'un serveur reçoit un jeton, il ajoute ce jeton dans la file de traitement et notifie le serveur de jeton qu'un nouveau jeton doit être traité. Lors du traitement d'un jeton, soit il peut être effectivement traité si les ressources du serveur le permettent, sinon le jeton est passé au serveur suivant de l'anneau.

2.3.1 Demande de traitement : *JetonDemandeTraitement*

Un serveur peut traiter un jeton de demande de traitement uniquement s'il n'est pas déjà en train de traiter une demande d'un client. S'il est déjà occupé, le jeton est alors passé au serveur suivant.

2.3.2 Traitement d'un morceau d'image : *JetonTraitement*

Lorsqu'un serveur reçoit un jeton de traitement, il regarde si un de ses threads de traitement est libre. Si c'est le cas, le thread démarre le traitement du morceau de l'image en fonction du filtre demandé. Lorsque le traitement est terminé le thread de traitement envoit le morceau de l'image au client. Le client est alors chargé de recoller les morceaux de l'image dans l'IHM.

2.3.3 Terminaison : *JetonTerminaison*

Ce jeton sert à détecter la terminaison de la plate-forme selon l'algorithme de Safra. Lorsque l'initiateur a détecté la terminaison, il envoit un jeton de stop pour informer les serveurs qu'ils peuvent s'arrêter.

2.3.4 Stop : *JetonStop*

Ce jeton sert à indiquer à un serveur qu'il peut s'arrêter. Si le serveur qui reçoit ce jeton n'est pas l'initiateur, le jeton est transmit au suivant. Lorsque ce jeton revient à l'initiateur, celui-ci renvoi une notification à l'administrateur pour l'informer que la plate-forme est correctement arrêtée.

FIG. 1 – L'ihm de notre application

3 Ihm

Notre interface est simple, une fois une image ouverte, elle est affichée sur toute la fenêtre.

Les menus de notre application sont les suivants :

- **Fichier** : pour ouvrir une image, l'enregistrer ou quitter l'application.
- **Configuration** : pour les paramètres de rmi.
- **Traitement** : les filtres qu'on peut appliquer sur nos images.
- **Maintenance** : pour arrêter les serveurs lancés.

4 Traitement

4.1 Filtre flou

Pour le filtre flou, nous avons appliqué l'astuce énoncée en tp : on fait simplement un rétrécissement (* 1/4) puis un zoom (* 4) sur les "morceaux" d'images.

4.2 Filtre négatif

Pour le filtre négatif, on applique sur chaque pixel d'un "morceau" d'image la transformation des composantes RGB du pixel de départ en leur inverse.

5 Mode d'emploi de notre application

Pour compiler notre projet, il faut faire **ant** à la racine.

5.1 Lancement des serveurs de l'anneau

Il faut d'abord lancer les serveurs "à la main".
Nous allons lancer ici un anneau de 3 serveurs (sur 3 machines).
Sur la machine 1 :

- **java serveur.ServeurLauncher Serv1 machine1 1099 Serv2 machine2 1099**

Sur la machine 2 :

- **java serveur.ServeurLauncher Serv2 machine2 1099 Serv3 machine3 1099**

Sur la machine 3 :

- **java serveur.ServeurLauncher Serv3 machine3 1099 Serv1 machine1 1099**

Une fois les 3 serveurs lancés, n'oublier pas d'appuyer sur "**Entrée**" dans la console de chaque serveur, de manière à faire l'anneau.

5.2 Lancement d'un client

5.2.1 Mode console

Pour lancer un client en mode console :

- **java client.ClientLauncher Client1 localhost 1099 Serv1 localhost 1099 ../test.jpg**

Il faut aussi configurer la maintenance :

- **java admin.AdminLauncher Admin localhost 1099 Serv1 localhost 1099**

5.2.2 Mode graphique

Pour faire les mêmes configurations qu'en mode console, on doit passer par le menu de configuration de notre ihm.

- **Menu Configuration » Rmi**

5.3 Ouverture d'une image à traiter

- Menu **Fichier** » **Ouvrir une image**

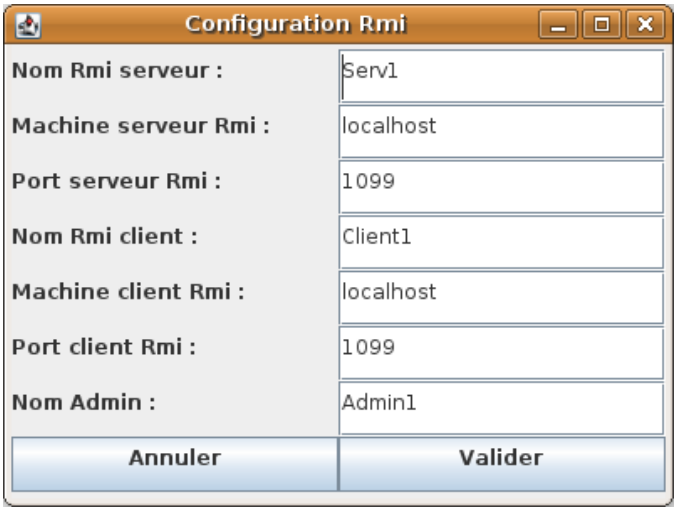

FIG. 2 – Fenêtre de configuration

5.4 Appliquer un traitement

– Menu **Traitement** » **Détection de contour** (ou **Negatif** ou **Noir et blanc** ou **Flou**)

5.5 Arrêt de la plate-forme

Pour demander la terminaison des serveurs, pour éffectuer une opération de maintenance par exemple :

– Menu **Maintenance** » **Arrêt de la plate-forme**

Les serveurs vont terminer les traitements en cours mais ils n'en acceptent plus d'autres.

6 Conclusion

Ce projet nous a permis de mettre en oeuvre l'algorithme de terminaison Safra vu en cours d'AD. De plus, nous avons pu tester nos connaissances quant à l'utilisation des threads et des mécanismes de synchronisation à mettre en oeuvre dans le cadre de la réalisation d'une application de calcul distribué.