

LSINF1252 Projet

Constantin de Schaetzen

Jean Gillain

11 mai 2018

1 Architecture

Au niveau de l'architecture, nous avons choisi de nous calquer sur le problème du producteur/consommateur. Tout d'abord, nos producteurs (un thread par fichier) lisent les fichiers, créent les fractales correspondantes et les insèrent dans un premier buffer. Ensuite, les consommateurs se chargent de récupérer chaque structure et de calculer la fractale avec la formule de récurrence.

2 Choix de programmation

Nous résumons ici les différents choix de programmation que nous avons effectué. Plus particulièrement, nous abordons : les buffers, la lecture des arguments, la création des structures fractales, le calcul de celles-ci, la façon dont nous avons comparé les fractales pour déterminer laquelle a la plus grande valeur moyenne et finalement l'arrêt des threads.

Buffers

Lecture Pour la lecture d'arguments, nous avons utilisé la fonction *getoptlong* qui permet de différencier simplement les options et les arguments. Nous avons utilisé un drapeau pour les options -d et -maxthreads. Si l'option -maxthreads n'est pas spécifiée, la valeur est mise par défaut à 4. Ensuite, nous lançons nos premiers thread. Chaque argument est donné à une fonction *producer* qui commence par déterminer si l'argument est le caractère '-' qui marque la lecture sur l'entrée standard. Si ce n'est pas le cas nous allons ouvrir le fichier, vérifier que l'ouverture s'est bien déroulée et ensuite lire le fichier ligne par ligne avec la fonction *fgets*. S'il faut lire sur l'entrée standard, l'utilisateur peut entrer des lignes une par une et écrire 'Stop' quand il souhaite arrêter.

Création des structures fractales Pour chaque ligne lue, nous avons la fonction *ReadLine* qui fonctionne par *Pattern matching* et qui crée la fractale correspondante si la ligne est correctement écrite. Si ce n'est pas le cas, la fonction retourne NULL. Chaque structure créée est insérée dans un premier buffer.

Calcul des fractales Les threads consommateurs vont récupérer les structures fractales dans le premier buffer et démarrer le calcul par récurrence.

Comparaison