

Laboratoire 2 - Bogosort

Calum Quinn - Urs Behrmann

Table des matières

- [Table des matières](#)
- [Introduction au problème](#)
- [Choix d'implémentation](#)
 - [Gestion des threads](#)
 - [Calcul du pourcentage effectué](#)
 - [Mise à jour de la barre de progression](#)
- [Tests effectués](#)
 - [Tri](#)
 - [Affichage](#)
 - [Reset](#)

Introduction au problème

Le problème premier de ce laboratoire est d'ajouter la partie multi-thread au Bogosort.

Ceci est nécessaire pour avoir des temps de calcul raisonnable car le Bogosort est très peu efficace.

Choix d'implémentation

Nos choix d'implémentation se séparent en plusieurs catégories.

- Gestion des threads
- Calcul du pourcentage effectué
- Mise à jour de la barre de progression

Gestion des threads

Pour la gestion des threads le format est assez basique, créer tous les threads demandés, laisser effectué le bogosort passé en paramètre et supprimer tous les threads.

Ce qui a requis plus de réflexion est la récupération du tableau trié ainsi que l'arrêt des autres threads n'ayant pas encore trouvé la solution.

Pour ça nous avons passé en paramètre du bogosort, à la création de chaque thread, une variable partagée pour stocker le résultat.

Lorsqu'un thread trouve la séquence triée, il "l'annonce au thread manager" en changeant la valeur d'une variable partagée et ceci arrête les autres threads qui obtiennent l'information que la solution a été trouvée. Cette variable est un `atomic boolean` pour éviter les problèmes de concurrence.

Le thread ayant trouvé la solution la met dans la variable partagée pour récupération par le thread manager.

Pour ne pas avoir de problème lorsque deux ou plus de thread ont trouvé la solution en même temps, nous avons mis un `qmutex` pour protéger la variable partagée.

Calcul du pourcentage effectué

Le calcul du pourcentage se fait environ à chaque pourcent d'avancement d'un thread. C'est-à-dire qu'il envoie un incrément de 0.1 à chaque fois qu'un pourcent des permutations a été vérifié.

Mise à jour de la barre de progression

La barre de progression est mise à jour à chaque fois qu'un incrément est envoyé par un thread. Le thread envoie son incrément de 0.1 au ThreadManager qui lui le relaie à l'UI.

Avant de faire ça le manager doit adapter le pourcentage pour qu'il soit en rapport avec le pourcentage total. Il doit donc diviser l'incrément par le nombre de threads car ils ont chacun le même nombre de permutations à vérifier.

Tests effectués

Tri

Il a fallu contrôler que l'algorithme de tri implémenté fonctionne réellement.

Pour ceci nous avons effectué de nombreux tri en contrôlant le résultat final.

Affichage

Lors du tri nous avons vérifié que la barre de progression s'incrémente petit à petit et que les valeurs correspondent réellement à l'avance dans le contrôle des différentes permutations.

Reset

Lorsqu'un tri est fini et que la solution a été donnée il faut pouvoir démarrer un autre tri sans fermer l'application.

Pour ceci nous avons contrôlé que les variables contrôlant le pourcentage calculé sont bien remises à zero pour que l'affichage soit correct pour chaque tri.