

Télécom Saint-Étienne 2A
Année universitaire 2012-2013
Java INFO 4.1
TP n°2 - 3h00

Les enseignants du module : J. Fayolle, C. Gravier, F. Laforest, J. Subercaze

OBJECTIFS DU TP:

L'objectif de ce TP est de vous familiariser avec la manipulation des collections en Java. Vous allez notamment apprendre à :

- Comprendre les différents mécanismes de synchronisation
- Map/reducer un programme

1 Exercice 1 : Différentes méthodes de synchronisation

Récupérez le TP qui est sur le github de l'école. Vous reportez au TP 1 en cas de besoin. Normalement un `git pull` dans le répertoire déjà créé devrait suffire.

Le package `fr.tse.info.tp7.exercice1` contient les fichiers relatifs à cet exercice. Le programme fournit contient un compteur qui est modifié parallèlement par plusieurs threads. Cependant il y a un petit problème ...

Question 1 Exécutez plusieurs fois la méthode `main` de la classe `Exercice1`. La valeur attendue du compteur est de 100 (10 incréments par 10 threads). Qu'observez-vous?

Question 2 Regardez les différentes classes (`Compteur`, `ThreadModifieur`) et identifiez la source du problème.

Question 3 Pour résoudre ce problème vous allez mettre en place 3 solutions différentes :

1. En utilisant le mot-clé `synchronized` sur une méthode
2. Avec l'utilisation d'objets atomiques
3. Finalement en utilisant des locks

2 Exercice 2 : Map/Reduce

Map/Reduce est une approche simple de la programmation multi-thread, elle consiste à effectuer un traitement en parallèle de données qui ne sont pas liées entre elles. Le but étant d'éviter les problèmes de synchronisation entre les threads.

L'approche fonctionne en deux phases :

1. **Map :** Le programme principal sépare les données en blocs distincts puis les envoie sur les différents threads (appelé communément workers) qui effectuent le traitement sur ces blocs.
2. **Reduce :** Le programme principal collecte les résultats des différents processus et les combine pour former le résultat.

Une des applications classiques de Map/Reduce est le comptage de mots dans un très grand volume de données. Dans cet exercice, nous allons compter les mots dans le jeu de données des

citations de film fourni par IMDB.com. Le total fait environ 140Mo, qui vous ont été aimablement coupé en 4 fichiers.

Préparation Des bouts de code vous sont fournis pour mener à bien ce TP. Ils sont situés dans le package `fr.tse.info.tp7.exercice2`.

Compteur.java C'est la version standard (sans MapReduce) du programme de comptage de mots, gracieusement fournie par vos enseignants

MapReduce.java Squelette du code à compléter qui contiendra le programme principal

Worker.java Squelette du worker pour le mapreduce

VolumeReader.java Permet de lire les fichiers contenant les citations

N'oubliez de configurer le projet en tant que projet Maven sous Eclipse !.

Question 1 Exécutez la `main` contenue dans `Compteur.java` et notez le temps d'exécution. Avec `Ctrl` + `Alt` + `Suppr` regardez l'utilisation des processeurs de votre machine. Qu'observez-vous et que pouvez-vous en déduire?

Question 2 : Map Compléter la classe `Worker.java` pour réaliser le comptage des mots. Afficher le temps requis pour la phase de Map.

Question 3 : Reduce La classe `Future` permet de gérer le stockage d'un résultat futur d'un thread qui n'a pas encore été exécuté. En vous inspirant du code disponible sur cette page, récupérez les différentes `HashMap` et fusionnez les pour obtenir le résultat final. Comparer ensuite le temps d'exécution avec la version `monothreadé`.