

Projet de programmation concurrente Java

Recherche d'expressions régulières sur le Web

AUGER-DUBOIS Benoît - GUILLET Nathan - MENANTEAU Melvin

Problèmes liés à la concurrence

Plusieurs problèmes liés à la gestion de la concurrence se sont présentés durant le développement de ce projet.

Le premier problème est le fait que plusieurs threads doivent pouvoir ajouter des pages à la liste contenant l'ensemble des pages visitées. Pour résoudre ce problème nous avons utilisé une `ConcurrentHashMap`, un structure de données spécialisée dans l'ajout de valeurs en concurrence.

Pour indiquer les actions restantes à effectuer sur une page ainsi que les données de la pages une fois visitées, nous avons créé une classe `Page` contenant cinq attributs :

```
/* L'url de la page */
final String href;
/* Indique si la page est visité, c'est à dire si un thread à récupérer la page pour la parcourir */
public final AtomicBoolean visited;
/* Indique que la page à été parcourue, on à donc les informations sur les liens et le nombre d'occurences qu'elle contient */
public final AtomicBoolean parsed;
/* Indique que la page à été imprimée */
public final AtomicBoolean printed;
/* Contient les données de la page */
public Tools.ParsedPage parsedPage = null;
```

Nous avons utilisé des `AtomicBoolean` pour nous assurer que les valeurs des attributs ne seraient pas affectées par le cache et qu'il serait uniquement possible de les modifier par un seul thread à la fois. De cette manière, si plusieurs threads accèdent à la même page, un seul pourra continuer son exécution.

Enfin, le plus gros problème se situe au niveau de l'impression des pages étant données qu'il ne doit pas y avoir d'entrelacement entre les sorties de deux pages.

Pour cela, nous avons utilisés un `ReentrantLock` indiquant si un thread était en train d'imprimer. Nous aurions pu ajouter le mot clé `synchronized` à la déclaration de la méthode 'print' de la classe 'Tools' ce qui aurait forcé les threads à attendre que la méthode soit disponible pour imprimer. Dans un but d'optimiser légèrement ce programme, nous avons jugés qu'avec l'implémentation d'un lock, on pourrait continuer l'exécution et donc continuer de visiter des pages. La page sera imprimée par un autre thread, une fois que le verrou aura été libéré.

Générer et exécuter le projet

SDK 17

```
javac -d "build" -cp ".;/lib/*" .\sujet\Main.java
java -classpath ".;/build;/lib/*" sujet.Main [args]
```