Projet
Programmation Concurrente

Dans ce projet, il nous fallait créer un système d'exploration d'URL permettant de récupérer les pages contenant un mot (ou une expression) à partir d'une page donnée. Ce système doit fonctionner en parallèle avec plusieurs Thread.

J'ai fourni 2 solutions différentes :

1. SharedSet

J'ai réalisé une solution avec 3 listes partagées :

- **toExploreURLs**
- **exploredURLs**
- **validURLs**

toExploreURLs est une **BlockingQueue** de String permettant de récupérer une ou plusieurs URL rapidement et en étant ThreadSafe. On vient récupérer un nombre défini d'URL depuis un des thread afin de les explorer.

exploredURLs et validURLs sont des **SharedSet**, qu'est une classe que j'ai créé d'une liste de String qui fonctionne avec un Lock afin de ne pas avoir de problème de concurrence entre les Thread.
La liste **exploredURLs** est une liste partagée contenant tous les URLs qui ont été exploré (ou qui vont être bientôt exploré par un thread).
La liste **validURLs** est aussi une liste partagée mais qui contient toutes les URLs qui contiennent le mot clé donné.

1.1 Fonctionnement

On vient donc pour chaque Thread, chercher s'il peut récupérer des URLs depuis **toExploreURLs**. Si on peut, on en récupère dans une liste propre au Thread (avec pour maximum le nombre défini) et qu'on ajoute à la liste **exploredURLs** (elles ne sont pas encore explorées mais on va le faire avec le thread qui les contient). Ensuite, on vient chercher chaque URL une par une pour voir si elle est valide. Si oui on l'ajoute à la liste **validURLs** et on récupère tous les liens href qu'elle possède (on enlève les images, .php, gif…) et qu'on ajoute à **toExploreURLs** et puis on répète ça.

2. ConcurrentHashMap

Dans cette solution, on possède seulement 2 “listes” partagées :

- **toExploreURLs**
- **exploredURLs**

toExploreURLs est une **BlockingQueue** de String permettant de récupérer une ou plusieurs URL rapidement et en étant ThreadSafe. On vient récupérer un nombre défini d’URL depuis un des thread afin de les explorer.

exploredURLs est maintenant une **ConcurrentHashMap** d’un **String** vers un **boolean**. Si la valeur est à **false**, cela veut dire que l’URL est soit en cours de recherche ou soit non valide. Si la valeur est à **true**, cela veut dire que l’URL est valide (contient le mot clé).

2.1 Fonctionnement

Il est assez similaire à celui de SharedSet:

On vient donc pour chaque Thread, chercher s’il peut récupérer des URLs depuis **toExploreURLs**. Si on peut, on en récupère dans une liste propre au Thread (avec pour maximum le nombre défini) et qu’on ajoute à la liste **exploredURLs** avec comme valeur false (si l’URL n’est pas déjà présente). Ensuite, on vient chercher chaque URL de la liste propre au Thread une par une pour voir si elle est valide. Si oui on modifie la valeur au sein de **exploredURLs** par **true** et on récupère tous les liens href qu’elle possède (on enlève les images, .php, gif…) et qu’on ajoute à **toExploreURLs** et puis on répète ça.

3. Résultats

Pour les résultats des deux méthodes, j’ai d’abord utilisé le mode offline de **Tools** afin de voir ce que cela donne dans un ensemble fini de donnée et on observe un résultat similaire :

ConcurrentHashMap :

```
toExploreURL : 0
exploredURL : 3518
validURL : 156
```

SharedSet :

```
toExploreURL : 0
exploredURL : 3518
validURL : 156
```

30 secondes avec le vrai web (100 threads, mot clé : Nantes, url de départ : https://www.wikipedia.fr/Nantes) :

ConcurrentHashMap :

```
toExploreURL : 54046
exploredURL : 2791
validURL : 502
```

SharedSet :

```
toExploreURL : 55220
exploredURL : 804
validURL : 479
```

Avis personnel :

Je pense sincèrement que mes résultats sont trop faibles/trop lents par rapport à ce qu'on pourrait faire avec 100 threads (donc environ 100 URLs à la fois). Cela peut très bien venir de ma connexion, du serveur qui refuse la connexion ou autres choses mais je pense qu'au sein de mon code, il y a des endroits possiblement optimisables afin que les Threads fonctionnent mieux. Mes deux solutions ne comportent normalement aucun doublon au sein des validURL ce qui est ce que l'on recherche.