

Rapport Projet Multi-Thread

Dans ce rapport, je vais vous parler des problèmes de concurrence rencontrés dans notre projet.

Le premier problème a été la gestion des accès concurrentiels à StartingURL et dejaFait. Pour ce faire, nous avons créé 2 méthodes portant le mot-clé 'synchronized' qui permette de manipuler ces deux listes. synchronized gérant la concurrence dans ces fonctions, chacun de nos appels seraient donc sans concurrence.

Le deuxième problème rencontré concernait aussi la liste StartingURL. Dans WebThread.java, dans la méthode run(), on boucle tant que cette liste n'est pas vide. Autrement dit, chaque thread fait au moins un appel à cette liste. Il nous a donc fallu mettre cette méthode dans une méthode synchronized aussi.

Le troisième problème de concurrence a été la gestion de l'écriture de nos erreurs dans un fichier texte. La méthode utilisée est une méthode qui écrit à la suite. Or si deux threads essayent d'écrire sur le même fichier en même temps,, il peut y avoir concurrence. Nous avons donc utilisé un lock que nous avons passé en attribut à chaque thread par une extension de la classe Thread. Cela nous permet d'utiliser notre lock librement, sans en générer un nouveau. Nous avons

Le dernier problème que nous avons traité a été la mise à jour du nombre de thread pour laquelle nous avons donc réutilisé le lock de la classe WebThread afin que deux thread ne puissent pas avoir accès à cette variable en même temps.

Loisel Emma, Beziau Timothé