

TP3 noté

Partie 1)

a) Fait, Voir code

b) Suivant la façon utilisée afin d'afficher l'état d'un site il peut y en avoir une ou zéro.

1. Par exemple, dans le cas où l'information est affichée sur la console depuis chaque thread, il n'y en a pas.

Voir [version 1 de la partie 1](#)

2. Par contre si l'on décide de retourner les informations par l'intermédiaire d'une collection au thread principale, il faut la partager. Si cette collection est thread safe, il ne faut rien faire d'autre. En revanche si elle ne l'est pas, on peut utiliser une mutex (sémaphore à un jeton) afin d'y accéder de manière exclusive.

Voir [version 2 de la partie 1](#). J'ai choisi la méthode thread safe avec une Hashtable

Concernant le nombre de threads, une bonne règle est de prendre le nombre de coeurs cpu physiques et de multiplier par deux (grâce au hyperthreading). J'en ai choisi 4 car sur mon pc, j'ai 4 coeurs physique et pas de hyperthreading. (En fait j'ai 5 threads mais bon ... 4 threads + 1 du main)

c) Fait, Voir code

Partie 2)

a) Fait, Voir code

Je vais utiliser une classe thread safe afin de gérer pour moi les accès concurrents. Ici une Hashtable.

Voir [version 1 de la partie 2](#).

b) La première solution est d'attendre que tout les threads du pool se soient exécutés puis de récupérer chaque valeur de retour via la méthode get() et de les additionner ensemble.

L'autre méthode consiste à cumuler toute les valeurs de la collection remplie par chaque singe lorsque tout les threads ont terminé.

J'ai fait les deux méthodes afin de vérifier la cohérence de mes valeurs. Les deux fonctionnent. J'ai également deux façons différentes d'attendre tout les threads et se, juste par curiosité.

Voir [version 1 de la partie 2](#).

c) Si j'utilise une ArrayList, il me faut rajouter une mutex (Sémaphore à un jeton) que je partage entre chaque singe et se afin qu'un seule ajoute dans la collection à un instant t et donc éviter un accès simultané(ou concurrent) à l'ArrayList.

Voir [version 1 bis de la partie 2](#).

d) [Voir version 2 de la partie 2](#)

Pour cela j'ai utilisé une autre collection thread safe qui est vector.
En revanche, cette collection est propre à chaque singe et n'est pas modifiée de façon concurrente, il n'est donc pas nécessaire d'utiliser une collection particulière.