



Wissam Salamé 1488205
Laboratoire 2

Présenté à monsieur
Hussem Daoud
INF4410

École Polytechnique
18 novembre 2015

Explication de la conception

C'est une architecture Client à plusieurs serveurs. Le client va lire un fichier d'opérations, qu'il va ensuite répartir selon le nombre de serveur auquel il a réussi à se connecter.

Si un serveur refuse la liste d'opérations, le client va réessayer avec moins d'opérations, au même serveur. Donc si le client tentait de lui passer l'ordre d'exécuter les opérations 10 à 20, la 2^e fois le serveur recevra l'ordre d'exécuter (par exemple) les opérations 10 à 15. Si le serveur refuse une 2^e fois, les opérations 10 à 15 seront ignorées dans le calcul.

Si un serveur échoue, les autres serveurs ne vont pas accepter plus d'opérations à la fois. Le code va continuer comme si les opérations ont toujours été acceptés, mais va tout de même détecter que l'on a terminé sans toutes les avoir exécuté. Par exemple, si l'on a arrêté à l'opération 85, et qu'il y en a 100 en total, un nouveau array de total 15 opérations (ceux entre 85 et 100) va être créés, et rappelé récursivement.

Il a été décidé que cette logique de créer un nouveau array plus petit s'il y en a qui manque peut être utilisé au plus 1 fois (dans la variable MAX_ATTEMPTS), qui vaut 2 par défaut (car on compte la première itération de base).

Pour détecter les pannes, le code va attraper une exception. Pour distinguer les exceptions dues à une erreur simple versus une exception due à la mort du serveur, un ping va être tenté auprès du serveur. Si le ping échoue, le serveur sera jugé hors-ligne, et le sera marqué comme tel.

Question 1

Les serveurs peuvent garder en cache les résultats des opérations (pour chaque répartiteur) tant que le répartiteur n'a pas envoyé de signaux à chacun des serveurs que les opérations sont terminés. Si le répartiteur tombe en panne, un 2^e répartiteur qui connaît le 1^{er}, ou du moins qui arrive à savoir d'une manière ou d'une autre que le répartiteur #1 est tombé en panne, prend le relai, mais en demandant tout d'abord les résultats en cache de la part de chacun des serveurs.

Évidemment, si tous les répartiteurs tombent en panne, et bien on ne peut pas récupérer le calcul des résultats. Par contre, les serveurs pourraient garder indéfiniment en cache (dans une base de donnée par exemple) les résultats, jusqu'à temps qu'un répartiteur revienne en vie.

Un autre scénario est si tous les serveurs tombent en panne avant que toutes les opérations aient finis.