

**POLYTECHNIQUE  
MONTRÉAL**

LE GÉNIE  
EN PREMIÈRE CLASSE



## INF8480 Systèmes répartis et infonuagique

### TP2 : Services distribués et gestion de pannes

**Gabriel Côté-Jones : 1771119**

**Aladin Riabi 1717705**

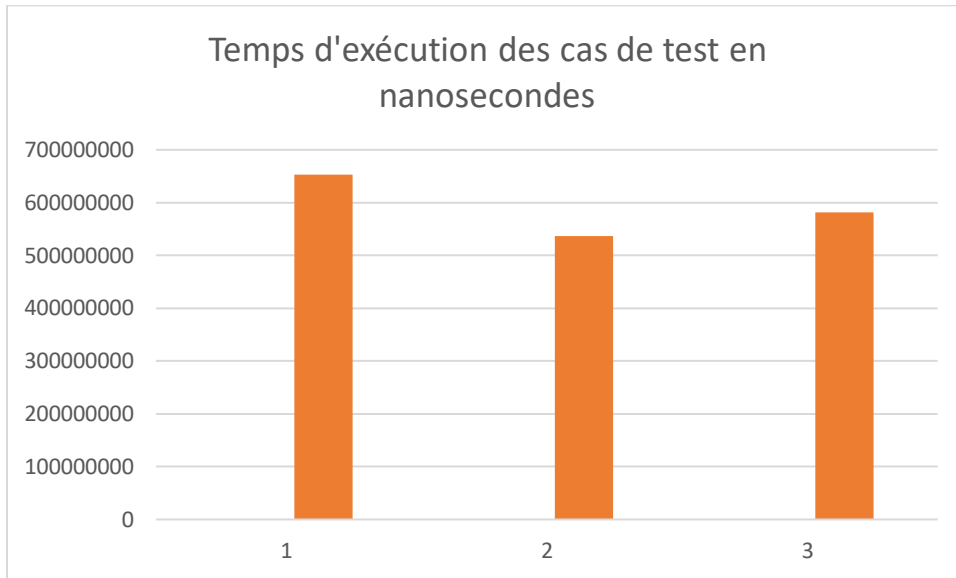
Polytechnique Montréal

18 mars 2019

## **Implémentation**

- **Répartition mode sécurisé :** La répartition en mode sécurisé est plutôt simple. Le répartiteur calcule et envoie une tâche d'une grosseur  $x$  à chaque serveur de calcul de la liste (un à un). La grosseur  $x$  de la tâche est calculée à partir de la formule  $T$  utilisée par les serveurs de calcul, de la quantité maximale d'opérations du serveur et d'une valeur % de refus ( constante THRESHOLD) décidé par l'utilisateur du système. Nous avons considéré de « multithreader » le répartiteur avec un thread par serveur de calcul (plutôt que de traverser la liste séquentiellement). Le nombre de serveurs de calcul disponibles aurait eu un plus gros impact sur le système avec cette dernière stratégie. Cependant, nous partions du principe que les serveurs de calcul étaient utilisés par plus d'un client/répartiteur à la fois et que renvoyer des tâches constamment au même serveur pourrait le surcharger.
- **Répartition mode non-sécurisé :** La répartition pour le mode non-sécurisé est similaire à celle du mode sécurisé. Cependant, dans ce mode, le répartiteur calcule une tâche pour trois serveurs à la fois (la même tâche pour les trois). Ces trois serveurs sont les trois suivant dans la liste de serveurs disponibles. Une fois les résultats obtenus, il y a une comparaison pour déterminer si un résultat est accepté. Il est à noter que si un des trois serveurs choisis refuse la tâche, on recommence avec trois nouveaux serveurs.
- **Gestion de pannes :** Pour détecter les pannes, les envois de tâches aux serveurs sont placés dans un bloc « try-catch » pour détecter un appel à un serveur qui n'est plus disponible. Si l'exception est lancée, le serveur est enlevé de la liste du répartiteur et la tâche est recalculée pour le serveur suivant.
- **Répartiteur :** Il devrait être mentionné que nous avons choisi d'implémenter le répartiteur comme un client. Il prend en entrée (lors de son exécution) le fichier d'opération, demande la liste au répertoire de nom (et s'identifie), répartit les tâches, affiche le résultat final puis se ferme.
- **Paramètres de lancement des serveurs de calcul :** Nous avons longuement considéré ajouter le hostname du répertoire de noms dans les arguments du shell script entrés sur la console lors de l'exécution. Cependant, avec déjà 5 arguments, nous avons opté pour un hostname fixe directement dans le code (changer la valeur de la constante REPERTOIRE\_IP au besoin). Une autre option considérée était un fichier de configuration pour chaque serveur avec son emplacement comme paramètre au shell script.
- **Langue du code :** La majorité du code est en anglais, ainsi que les affichages de consoles. Cependant, les noms de classe (RepertoireNoms, Repartiteur, ServeurCalcul, ...) sont en français pour ne pas qu'il y ait d'ambiguïté sur le rôle de chaque classe par rapport à la description du TP2 dans l'énoncé.

## Tests de performance – mode sécurisé

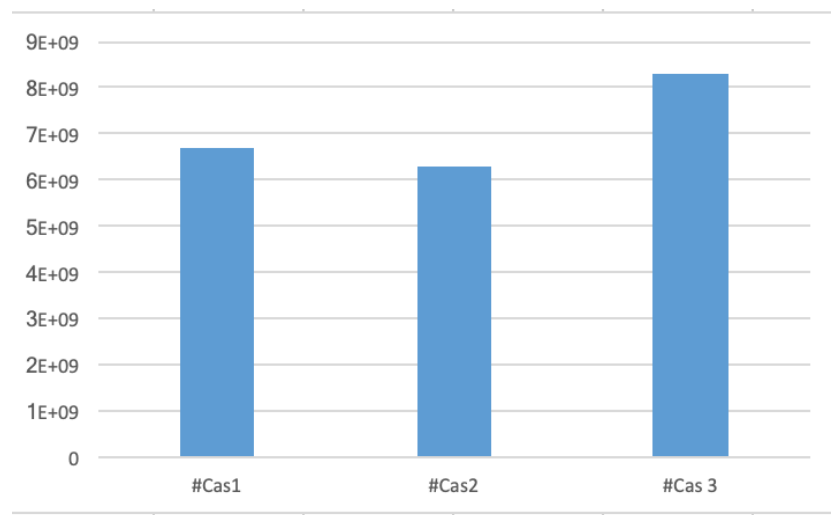


*Fig. 1 : Les temps d'exécution des cas 1, 2 et 3 mode sécurisé de l'énoncé de TP2 en nanosecondes.*

On voit avec le graphique que le nombre de serveurs influence peu la vitesse du système. En fait, elle peut même augmenter selon les refus (aléatoire). Cela, car la stratégie de distribution dépend de la capacité maximale d'opérations des serveurs et non du nombre de serveurs de calcul disponibles. En effet, le répartiteur ne fait qu'alterner entre les serveurs pour envoyer une tâche, puis passe au suivant dans sa liste peu importe si la tâche était refusée ou acceptée.

Une meilleure stratégie aurait été de « multithreader » le répartiteur avec un thread par tâche. Ainsi, la vitesse de calcul aurait été influencée par la capacité maximale des serveurs et le nombre de serveurs de calcul disponibles.

## **Tests de performance – mode non-sécurisé**



*Fig. 2 : Les temps d'exécution des cas 1, 2 et 3 mode non-sécurisé de l'énoncé de TP2 en nanosecondes.*

On voit que les temps obtenus sont beaucoup plus gros que ceux du mode sécurisé. Cependant, ils ne changent pas beaucoup d'un test non-sécurisé à un autre test non-sécurisé. La stratégie pour le mode non-sécurisé calcule une même tâche pour trois serveurs selon leur quantité maximale d'opérations. Le répartiteur envoie ensuite cette tâche aux trois serveurs puis compare les résultats.

Cette stratégie prend beaucoup plus de temps que celle du mode sécurisé, car si un serveur refuse la tâche, une nouvelle tâche doit être reconstruite. Il y a donc 3 fois plus de chance de reconstruire une tâche à cause d'un refus.

Les temps d'exécution entre les différents tests non-sécurisé ne changent presque pas, car il y a toujours au moins deux serveurs de calcul fiables. Ainsi, le répartiteur trouve inévitablement une réponse acceptable lors la phase de comparaison (après avoir reçu les trois réponses). Le temps augmenterait de beaucoup avec plus de serveurs non fiables, car le répartiteur obtiendrait souvent des résultats non satisfaisants et devrait recalculer et renvoyer une tâche.

## **Question de réflexion**

Cette réflexion part du principe que contrairement à notre implémentation, le répartiteur est un serveur et non le client lui-même. Il pourrait y avoir plusieurs répartiteurs. Les autres répartiteurs seraient en « standby » et demanderaient périodiquement à celui qui est actif les résultats calculés jusqu'à présent (ainsi que d'autres informations nécessaires à la répartition). Le client devrait également avoir une liste de ces répartiteurs. Ainsi, à la fermeture du répartiteur utilisé, si les calculs sont incomplets, le client pourrait demander à un autre répartiteur de continuer là où les calculs se sont arrêtés (ou du moins depuis la dernière mise à jour périodique). Quant à la sélection du répartiteur à activer, les répartiteurs peuvent avoir une numérotation et le suivant (selon son numéro) enverrait un message aux autres répartiteurs pour leur signaler son activation.