

Rapport ABDR

BELLACICCO Denis
SIMONET Lucas

5 janvier 2014

1 Transaction et concurrence

1.1 Exercice 1

Dans le tme, il nous est demandé d'écrire un programme, exécutant en parallèle deux threads, qui accèdent en simultané à un même profile. Lors de l'analyse des résultats obtenus après l'exécution de ce programme, on constate que l'on a d'importante perte d'écriture. En effet, le but de l'application étant que chaque thread incrémente la valeur d'un couple clé,valeur de un pendant mille itérations. On s'attend à ce que, lors de la terminaison de l'application, cette valeur soit de égale à sa valeur initiale plus deux milles. On utilise, pour ajouter et modifier les couples clés, valeurs, la fonction proposé par kvstore, KVStore.put() Or, la valeur dans la base de donné après l'exécution de l'application est inférieur à la valeur attendue. Ce problème est du à l'absence de contrôle lors des accès en concurrence de plusieurs applications sur la même base de données. Pour résoudre ce problème, il faut s'assurer que la version de la valeur que l'on veut modifier est la même que celle que l'on a lu.

1.2 Exercice 2

Dans l'exercice précédent, nous avons résolu le problème du à l'incohérence entre la lecture et l'écriture des données. Or lors de l'exécution de deux programmes M1 simultanément ne conserve pas la cohérence des données. Lors d'une itération, les incréments effectués sur les produits sont faites de manière totalement indépendante. Du coup, deux programmes peuvent se chevaucher lors de l'incrément des produits et donner des résultats incohérents. Si les produits on, au départ, tous la même valeur, ils ne l'auront pas forcément à la fin de l'exécution du programme. On décide de résoudre ce problème en incrémentant le max des valeurs de tout les produits et que ces derniers soit alors égale à cette nouvelle valeur. Cela permet d'être sur que les valeurs de tout les produits seront les même. On écrit donc un programme M2 en conséquent.

Lors de l'exécution de deux programme M2 en série, la quantité pour tous les produits vaut 201. Mais lors de l'exécution de deux programmes M2 en parallèle, on observe que le résultat obtenu n'est pas le même que précédemment. Lors d'une itération, certain produit peuvent avoir été mis à jour par un autre programme au même moment.

La solution pour résoudre ce problème, est de rajouter chaque opération à effectuer lors d'une itération dans une liste d'opération et de tout exécuter en une fois. KVStore permet cette solution, du coup toutes les opérations sont effectuées de façon "atomique". On retrouve une cohérence des données avec une exécution en série.