

## TmeKvstore



# TME Oracle KV Store

## Préparation

- Vérifier qu'il vous reste au moins 50Mo d'espace disque disponible
- Télécharger et décompresser oracle kv store (fichier kv-ce-2.1.54.tar.gz)
- Lire la documentation oracle nosql (en particulier le *quick start*) (PDF)
- Lien vers l'API java
- copier dans le dossier kv-2.1.54, le fichier config-java-classpath-kvlite.txt
  - cd *répertoire kv-2.1.54*
  - source config-java-classpath-kvlite.txt (pour configurer la variable CLASSPATH)
  - java -jar lib/kvstore-2.1.54.jar kvlite -root /tmp/kvroot (pour démarrer le serveur kvlite). L'option -root permet de préciser que le store utilise un support local pour stocker les données.
    - en cas d'erreur au démarrage de kvlite, supprimer entièrement le dossier kvroot (c'est un sous dossier de kv-2.1.54).
- copier dans le dossier exemples, le fichier Init.java (txt )
  - cd exemples
  - lire et comprendre l'exemple Init.java qui ajoute l'objet {P1,0} dans la base.
  - javac Init.java
  - java Init

## Exercice 1

On considère une base de 100 objets dont la clé est P1 à P100 et la valeur représente une quantité de produit.

- Ecrire le programme A1 qui effectue une boucle de 1000 itérations. A chaque itération, ajouter 1 à la quantité du produit P1. Observer les pertes d'écriture lorsque plusieurs programmes A1 s'exécutent simultanément.
- A2: utiliser l'écriture conditionnelle pour contrôler l'exécution simultanée de plusieurs programmes.
  - voir la methode *putIfVersion*. La valeur de retour est nulle lorsque l'écriture est refusée.

## Exercice 2

On considère une base de 200 objets classés en 10 catégories C1 à C10. La clé d'un objet est composée de la catégorie suivie du numéro du produit. La catégorie C1 contient les objets P1 à P20, C2 contient les objets P21 à P40 et ainsi de suite. La valeur d'un objet représente une quantité de produit. Initialement tous les produits ont une quantité égale à 1. On considère une transaction qui modifie la quantité de 5 objets d'une certaine catégorie.

- Ecrire un programme M1 qui effectue 1000 itérations. A chaque itération, on ajoute 1 à la

- quantité des produits P1 à P5 de la catégorie C1.
- L'exécution simultanée de 2 programmes M1 conserve-t-elle la cohérence des données ?
  - Ecrire un programme M2 qui effectue 1000 itérations. A chaque itération, on détermine la valeur *max* qui est la plus grande quantité parmi les produits P1 à P5 de C1. Puis on affecte la nouvelle quantité =  $(1 + \text{max})$  à tous les produits P1 à P5.
  - Si au départ la quantité vaut 1 pour tous les produits, quelle quantité obtient-on après avoir exécuté 2 fois le programme M2, en série (i.e., l'une après l'autre) ?
  - L'exécution simultanée de 2 programmes M2 conserve-t-elle la cohérence des données (la quantité obtenue est-elle la même qu'avec une exécution en série)?
    - Proposer une solution pour que M2 conserve la cohérence des données.
      - définir une transaction comme étant une liste de 5 opérations d'écritures (un écriture conditionnelle pour chaque produit)
      - Structurer la clé des produit pour que le préfixe (major component) soit commun à tous les produits modifiés ensemble
      - Pour cela obtenir une OperationFactory en appelant la méthode `getOperationFactory()` de KVStore
      - Puis créer une liste d'Operation en appelant la méthode `createPutIfVersion` de la classe OperationFactory avec le paramètre boolean `abortIfUnsuccessful`
      - Exécuter la séquence en appelant la méthode `execute` de KVStore
      - Détecter l'abandon éventuel de l'exécution à l'aide de l'exception `OperationExecutionException`

## Questions fréquentes

- Erreur de connexion. Vérifier que le serveur kvlite est bien démarré:
  - `cd kvstore-2.0.23`
  - `java -jar ../lib/kvstore-2.0.23.jar ping -host localhost -port 5000`
  - si le problème persiste: supprimer le dossier kvroot puis relancer le serveur kvlite
  - si nécessaire éteindre le serveur kvlite avec les commandes `ps u` et `kill`
- Erreur avec les lettres accentuées pendant la compilation javac
  - `export LC_CTYPE="fr_FR"`
  - ou `javac -encoding ISO-8859-15 *.java`

---

## LesTme

---

Propriétaire : [abdr](#) Dernière modification le novembre 13, 2013 5:15 par [abdr](#)