Paux Verstraeten

Rapport d'évaluation

---

1. Partie technique

a. Évaluation architecture

i. Rappel architecture - classes principales

➔ Daemon_DataNode.java : Serveur RMI à lancer sur toutes les machines faisant tourner un dataNode, permet d'effectuer à distance un traitement Map ou Reduce sur les chunks enregistrés par le DataNode.

➔ Job.java : Client RMI, recupère les dataNodes géreant un fichier dans le hdfs, appelle les procedure Map des Daemon_DataNodes correspondants sur les chunks du fichier, transfère les résultats sur un Daemon_DataNode chargé du Reduce et appelle le Reduce sur ce Daemon_DataNode.

➔ MyMapReduce.java : implémentation d'une application MapReduce, définie les procédures MAp et Reduce à appliquer aux fichiers.

➔ KVFormat.java et LineFormat.java : Classes permettant écriture et lecture sur fichiers locaux de types KV et Line.

ii. Critique de l'architecture et améliorations possibles

➔ Daemon_DataNode.java : Serveur RMI à lancer sur toutes les machines faisant tourner un dataNode, permet d'effectuer à distance un traitement Map ou Reduce sur les chunks enregistrés par le DataNode.

➔ Job.java : Client RMI, recupère les dataNodes géreant un fichier dans le hdfs, appelle les procedure Map des Daemon_DataNodes correspondants sur les chunks du fichier, transfère les résultats sur un Daemon_DataNode chargé du Reduce et appelle le Reduce sur ce Daemon_DataNode.

➔ MyMapReduce.java : implémentation d'une application MapReduce, définie les procédures MAp et Reduce à appliquer aux fichiers.

➔ KVFormat.java et LineFormat.java : Classes permettant écriture et lecture sur fichiers locaux de types KV et Line.

➔ Job.java utilise le port 6060 codé en dur pour chercher le serveur RMI qui lui est lancé avec comme argument le port d'écoute voulu.

- ➔ KVFormat.java et LineFormat : Chaque lecture enclenche une nouvelle ouverture du fichier en lecture plus un re-parcours complet afin d'arriver là où on avait fini la lecture, pourquoi ne pas juste garder le fichier ouvert en lecture constamment. (corrigé sur les dernières versions du git).

- ➔ Des méthodes dans KVFormat, LineFormat et Daemon_DataNodes qui sont appelées par d'autres classes (des fois à distances) ne lèvent jamais d'exception pour prévenir que quelque chose c'est mal passé mais à la place font juste un printStackTrace -> à corriger.

- ➔ Possible fusion des classes SlaveEnvoyerVers et SlaveMap : SlaveMap est un thread lancé pour envoyer l'ordre à chaque Daemon_DataNodes d'exécuter une fonction Map sur un fichier, et SlaveEnvoyerVers est aussi un thread demandant aux même Daemon_DataNodes de renvoyer leurs résultats vers le Daemon_DataNodes réalisant le reduce. Pourquoi ne pas ajouter l'envoi du résultat à la fin du SlaveMap = moins de code de création et join de threads.

- ➔ KVFormat et LineFormat -> pourquoi utiliser un chemin absolu écrit en dur dans config/Project.java pour préfixer les noms des fichiers et pas juste un chemin relatif. (Les exécutables du projet se lancent et se compilent dans le répertoire src, c'est dans src/ que l'exécutable cherche les fichiers).

b. Évaluation fonctionnalités

Le service d'exécution en parallèle de fonctions Map sur plusieurs chunks du même fichier fonctionne, ainsi que la liaison des différents résultats par le Reduce. Cependant le découpage simuler sans hdfs ne fonctionne pas. Le map s'effectue sur le fichier non découpé et le reduce double ainsi le nombre de mots compter pour l'exemple du calcul du mots d'un fichier.

Il serait bien de pouvoir préciser un dossier racine dédié à la machine à utiliser pour chercher les fichiers pour le Daemon_DataNodes, car si on teste sur plusieurs machines de l'n7 les Daemon_DataNodes écrivent le même fichier résultat au même endroit avec le même nom sur notre /home monté à distance, impossible à tester.

c. Evaluation de performance

L'évaluation de performance ne peut être testée sur cette version. Il faudra attendre la prochaine version.

2. Synthèse

- ➔ Correction : Le produit fonctionne plus ou moins, la communication entre le client RMI et les serveurs RMI fonctionne, de même pour le MapReduce. Il reste la présence de bugs causés par les dossiers partagés des machines de l'N7 lié au compte utilisateur qui empêche de tester les résultats et la performance du MapReduce.
- ➔ Complétude : Toutes les fonctionnalités abordées sauf le RessourceManager, pas de suivie des ressources allouées.
- ➔ Pertinence : L'architecture Hidoop est respectée.
- ➔ Cohérence : Logique claire de l'architecture.