École Nationale Supérieure
d'Électrotechnique, d'Électronique, d'Informatique,
d'Hydraulique et des Télécommunications

Institut National Polytechnique

Rapport d'évaluation du Hidoop V0

Chailou Salah Eddine
Chla Aymen

SOMMAIRE

1 Partie Technique

1.1 Résultats des tests

Nous avons effectué plusieurs tests en local sur le fonctionnement du code fourni qui gère la partie map/reduce. On a fait un test avec 3 daemons, ces derniers démarrent correctement, après on a lancé MyMapReduce qui est responsable de créer et lancer le Job qui se charge de communiquer avec les deamons pour récupérer le résultat du runMap executé sur chaque Chunk du fichier. Afin de les rassembler et faire le Reduce pour obtenir le résultat finale.

Nous avons remarqué que certaines commandes faisaient planter le programme, notamment : demande de faire un map sur un fichier qui n'existe pas sur le serveur. Et cela est due au fait que le Job ne communique pas avec un noeud centrale (NameNode) pour récupérer les métadatas ainsi que l'emplacement des fichiers déja enregistrés sur le HDFS.

Hormis ces erreurs, le programme fonctionne correctement et donne le résultat attendue, qui est conforme au résultat du test fourni.

1.2 Validation des performances

Le code fourni fonctionne, bien que ce ne soit pas en réparti et uniquement en local. Ainsi on peut faire le map sur les daemons et enfin le reduce sur le Reducer avec les meilleurs performances, même sur des fichiers assez volumineux. Le système utilise RMI pour assurer la communication entre le Reducer et les daemons, ainsi que les sémaphores dans le Callback pour gérer l'attente des maps ce qui est un choix performant.

1.3 Qualité du code

Le code est bien commenté et très bien organisé ce qui nous a facilité la tâche pour la lecture, la compréhension et par la suite les tests. Ceci est due à l'utilisation des bonnes pratiques de programmation notamment quelques patrons de conception comme le Factory pour créer des instances selon des conditions spécifiques ce qui allège le code.

2 Synthèse

2.1 Correction

Nous avons pu voir dans la partie Validation que le code fonctionnait plutôt correctement sous condition que les chunks existent dans les serveurs où les Daemons sont lancés, cela pourrait engendrer des erreurs dans le cas contraire. La solution consiste donc à soliciter le noeud centrale (NameNode) pour récuperer les informations (Métadata) concernant le fichier enregistré à l'avance sur HDFS et ainsi connaitre l'emplacement des serveurs contenant les différents chunks.

2.2 Complétude

Comme complétude au correction du programme, les dameons peuvent être centralisés et enregistrés directement sur le NameNode au lieu que chaqu'un de ces derniers s'entregistre sur le LocateRegistry.

2.3 Pertinence

Le code implémenté semble pertinent vu l'utilisation des threads pour lancer les maps en mêmes temps sur chaque serveur. Ainsi que les sémaphores dans le Callback pour gérer l'attente de fin des maps ce qui est un choix performant.

3 Pistes d'amélioration

Voici quelques améliorations possibles :

Améliorations	**Gain**
Soliciter le noeud centrale (NameNode) pour récuperer les informations (Métadata) concernant le fichier enregistré à l'avance sur HDFS.	Connaitre l'emplacement des serveurs contenant les différents chunks.
Ajouter des fichiers dans le dossier config contenant les différents informations a propos les adresses et les ports.	Stocker les paramètres de configuration du programme.
Les dameons peuvent être centralisés et enregistrés directement sur le NameNode au lieu que chaqu'un de ces derniers s'entregistre sur le LocateRegistry.	Faciliter l'accès aux serveurs.