Projet Système Concurent : Partie HIDOOP

JEGO ANTOINE - BROURI SOUFIANE - SAMUEL SQUILLACI

5 décembre 2018

Table des matières

1 Introduction

Ce projet s'inscrit dans l'unité d'enseignement "Systèmes concurents et communicants" et a pour but la conception et la réalisation d'une plateforme nommé HIDOOP capable d'exécuter un MapReduce sur des machines en parallèle à travers une intérface de programmation ou bien par lignes de commandes.

2 Architecture globale du prototype Hidoop

Hidoop est constitué de deux principaux objets :
- Job dont le rôle est de faire l'ordonnancement en commandant et répartissant les tâches à exécuter avant d'agir sur les resultats retournés.
- Daemon, la partie opérative du système et qui exécutent les tâches qu'on leur attribue.

2.1 JOB

L'interface JobInterface constite l'API fournie à l'utilisateur pour lancer un MapReduce sur l'ensemble des machines.

Réalisation de Job Le fonctionnement de startJob est le suivant :
- Mise en relation avec HDFS afin de fragmenter les fichiers d'entrée.
- Mise en relation avec NameNode par RMI pour localiser ces fragments.
- Génération des formats et faire la demande au démon concerné pour exécuter les "Map".
- Attendre la terminaison des tâches.
- Rassembler les résultats via HDFSclient et exécuter le "Reduce".

2.2 Daemon

Réalisation de Daemon Le fonctionnement de Demon est le suivant :
- Lancement de la méthode "runMap" executable par RMI.
- à travers un "main", le lancement des démons sur chaque noeud du cluster et de les enregistrer dans RMI.
- Lecture des résultats du lancement du "Map" et écriture de ces résultats sur les fichiers locaux.

3 Mode de fonctionnement

3.1 L'API

JobInterface constitue l'interface de programmation à travers laquelle l'utilisateur peut exécuter un MapReduce. Pour le faire il faut donner un certain nombre de paramètres (méthodes getFname, setFname par exemple) et faire appel au jobstart() qui démarre les calculs.

3.2 Utilisation en ligne de commande

Lancement de Job La commande suivante permet de lancer un Job :
java ordo.Job (config_path). Un job nécessite un fichier de configuration (qu'on trouve dans un répértoire config/) qui gère les formats et les fichiers en input et en output.

Lancement du Daemon La commande suivante permet de lancer un Daemon :
java ordo.DaemonImpl (id port). Les paramètres "id" et "port" correspondent à :

- id : identifiant du Daemon.
- port : le port sur lequel il doit se connecter en RMI.

Remarque Ces informations doivent etre en adéquation avec ceux fournis par le NAMENODE.

4 Conclusion

Lors de cette première phase, nous avons réalisé le coeur de notre plateforme hidoop à savoir les deux briques essentielles : Job et Daemon. La version actuelle nous permet d'exécuter un code de masse importante sur un cluster de machines en parallèle.