

# Hidoop

Bonnet Olivier, Steux Yoann, Xambili Robin

**Abstract**—Explication du fonctionnement et de l'implantation de Hidoop et des technologies utilisées.

## I. INTRODUCTION

**H**IDOOP est une plateforme dont le principe est de traiter une grande quantité de données en même temps en utilisant pleinement toutes les ressources possibles. Le travail est séparé sur plusieurs processeurs en même temps. Le travail réalisé ne concerne que la partie sur la répartition des data.

## II. STRUCTURE GÉNÉRALE

Le client va envoyer la fonction à appliquer sur chaque partie de texte à chaque daemon présent sur des serveurs qui s'occupent d'une partie de texte. On a utilisé pour cela la technologie RMI.

### A. Client

La classe Job implémente JobInterface. Il représente le "client".

La fonction startJob demande à chaque daemon des machines présentes dans le fichier de configuration (pour le moment on utilise une simple liste avec les machines utilisées) de lancer le runMap, en leur donnant le nom du fichier sur lequel lire, celui sur lequel écrire, ainsi que la fonction à exécuter.

Dans cette v0, nous n'avons pas utilisé de callback, puisque les tests ont été effectués sur une seule machine. On applique ensuite la fonction reduce présente dans l'objet MapReduce/Mapper qui recalcule le résultat en utilisant tous les "bouts" de résultats qui ont été récupérés par Hdfs.

### B. Serveurs

La classe DaemonImpl implémente Daemon. Les daemon sont lancés sur les serveurs.

Le main crée un registry sur le serveur. Il y aura donc autant de registry qu'il y a de serveurs.

La fonction runMap lancée par Job ouvre les fichiers présents dans les FileReader et FileWriter passés en arguments. La fonction MapReduce passée en argument est exécutée avec les fichiers précédemment ouverts.

### C. Formats

La classe FormatImpl implémente Format, FileWriter et FileReader.

Les fonctions open() et read() lisent et écrivent dans les fichiers, en respectant les formats KV ou LINE.

## III. COMMENT TESTER

### A. Déployer les daemon

En local :

Dans différents terminaux :

```
java DaemonImpl 1
java DaemonImpl 2
java DaemonImpl 3
java DaemonImpl 4
```

### B. Lancer l'application

Dans le terminal client :

Se placer dans /hidoop

```
java src.application.MyMapReduce
data.filesample.txt
```

## IV. TESTS DE COMPLEXITÉ

Pour tester cette partie de Hidoop, nous avons séparé le fichier texte en 4 parties manuellement.

Exemple du travail réparti en 4 parties et tout sur la même machine :

TABLE I  
TEMPS DE CALCULS

|             |        |
|-------------|--------|
| MyMapReduce | 307 ms |
| Count       | 39 ms  |

## V. CONCLUSION

Pour la prochaine version, l'objectif est d'améliorer les performances (lancer sur différentes machines), implémenter callback pour commencer le reduce quand tous les calculs sont terminés, rendre plus robustes les délégations de calcul en dupliquant les différentes parties de texte ...