

# Évaluation intermédiaire du projet Hidoop

Voici l'évaluation menée sur le rendu intermédiaire du projet Hidoop développé par Corentin Dominguez et Guillaume Claverie, évalué par Théo Petit et Gautier Ben Aïm.

## Partie technique

Nous avons testé le prototype de la plateforme Hidoop avec le prototype de la plateforme HDFS. Pour l'instant ces deux prototypes ont pour contrainte d'être limités à un unique serveur. Notre protocole de test est le suivant :

- Compiler le projet
- Lancer un serveur RMI
- Lancer un serveur HDFS
- Lancer un serveur MapReduce
- Lancer le main() de MyMapReduce sur l'exemple du poème
- Consulter le fichier de résultat créé

Les lancements sont périlleux à cause du problème du dossier courant : il faut que le dossier courant soit la racine du projet pour certains serveurs, le dossier "src" pour d'autres. Nous avons relevé ce problème dans les pistes d'amélioration.

Le fichier résultat créé contient bien le résultat attendu dans le cas du compte de mot. C'est la seule application fonctionnelle pour le moment.

Par ailleurs, connaissant les limites de la plateforme, nous n'avons pas tenté de la mettre en défaut, seulement de la faire fonctionner dans un cadre normal d'utilisation.

Nous n'avons pas évalué les performances car à ce stade d'avancement cela ne paraît pas pertinent.

Le code est de bonne qualité. Il y a cependant quelques warnings pour des variables et des imports inutiles, mais ils semblent venir en grande partie des fournitures du projet.

## Synthèse

### Correction

Nous avons testé la version intermédiaire du projet Hidoop sur l'exemple de comptage de mot. Bien que le prototype ne fonctionne que sur un seul nœud, il est tout à fait fonctionnel et le décompte des mots est correct.

### Complétude

Il manque un des points clés du projet : le fonctionnement en cluster, et toutes les problématiques liées (la répartition du travail, la collecte ordonnée...).

## Cohérence

La conception proposée colle au squelette fourni pour le projet, et est tout à fait cohérente.

## Pistes d'amélioration

1. Faciliter le lancement du projet avec un script ou un Makefile. En l'état le lancement est très artisanal, avec les lancements séparés du serveur rmi et des workers, ainsi que les problèmes liés au dossier courant.
2. Une meilleure gestion des erreurs. Pour l'instant la gestion des erreurs est toujours "e.printStackTrace();", sans arrêt du programme en cas de problème. Nous souhaiterions des messages d'erreur explicites, ainsi qu'une terminaison du programme en cas d'erreur critique.
3. Commenter le code avec des docblocks et le formater, pour rendre la lecture agréable. Cela n'est actuellement pas problématique, mais cela permettrait d'utiliser des outils comme javadoc.