

Reponses à l'évaluation de MapReduce

- KV/Format.java et LineFormat : chaque lecture enclenche une nouvelle ouverture de fichier.
 - Problème résolu, nous avons ajouter une méthode open() et une méthode close() pour n'avoir qu'à ouvrir le fichier qu'une seule fois.
- Des méthodes dans KVFormat, LineFormat et Daemon_dataNodes ne levent jamais d'exceptions pour prévenir qu'il y a eu une erreur.
 - Problème résolu, nous avons ajouter des exceptions et des messages d'erreurs clairs là où s'était possible.
- SlaveMap et SlaveEnvoyerVers peuvent être combinés car on effectue systématiquement un EnvoyerVers après un Map.
 - Effectivement les deux peuvent être fusionnés cependant nous n'avons décider de ne pas les fusionner car nous preferons avoir une séparation claire entre les différents appels à distance. De plus dans le cas d'un très grand nombre de fragments le 1er Map pourrait avoir fini et donc envoyer son résultat avant que le Serveur RMI n'est fini d'envoyer les ordres d'execution aux autres machines, le serveur ne recevrais alors jamais le résultat du 1er Map.
- Config.Project.PATH : pourquoi avoir un chemin absolu écrit en dur, il vaudrai mieux avoir un chemin relatif.
 - Ceci a été corrigé, le chemin PATH est maintenant relatif et depend de où on execute l'application.
- Le port de communication de Job (6060) est codé en dur alors que Daemon_dataNode prend un port en paramètre.
 - Effectivement ce n'est pas très malin, on pourrais rajouter un argument à Job pour permettre de changer le port de communication, ceci n'a pas été corrigé.
- Il est difficile de lancer un traitement MapReduce depuis une même session sans passer par le dossier work/ (disque dur de la machine) car les fichiers ont tous le même nom et ils s'ecrasent entre eux.
 - Ce problème a été corrigé, il est possible de lancer un MapReduce depuis une même session utilisateur sans passer par le dossier work/, de plus une console a été ajouté pour rendre l'utilisation de MapReduce plus simple.