Amirkaveh Chervin
Laporte Adrien

ENSEEIHT – Toulouse INP
Étudiants 2ème année
Filière SN Spécialité SIL

Rapport Données Réparties

Evaluation HDFS v0

Table des matières

1 Partie technique

1.1 Tests et validation

Tout d'abord nous avons tenté de lancer les serveurs clients, ces derniers semblent se lancer correctement et fonctionner. Nous avons ensuite testé les fonctionnalités du client en exécutant les différentes commandes possibles à partir des exemples fournis. On constate à l'exécution de write que le fichier est bien découpé en plusieurs morceaux. Read retourne bien une concaténation de l'ensemble des fragments et delete supprime bien les fragments. Cependant, nous avons constaté que si on exécute la commande write avec un nom de fichier inexistant la fonction renvoie une exception qui n'est pas rattrapée.

L'utilisation d'une Hashmap dans une Hashmap dans le cadre du stockage des fichiers sur les serveurs paraît être pertinente au niveau de la performance. Au premier abord, on pourrait penser que l'on perd en rapidité d'accès en utilisant deux Hashmap emboîtées. Ce n'est pas le cas car la recherche de fragments est bien plus optimisée et facilitée à l'aide de ce système.

L'ensemble des fonctions liées aux serveurs et clients fonctionnant, on peut supposer que les formats fonctionnent correctement. Nous avons tout de même mis en place quelques tests rapides pour tester qu'il n'y pas d'anomalie. Les tests consistent en des lectures et écritures dans les deux formats, nous n'avons pas constaté d'erreur.

1.2 Propositions de corrections

Le lancement des serveurs se fait sans problème à l'aide de la classe HdfsServer. Cependant, les serveurs doivent être lancés les uns après les autres, en modifiant à chaque lancement la valeur du port. En cas de grand nombre de serveurs, cela semble impossible, une automatisation de ce processus serait bienvenue.

Dans l'implémentation de la classe HdfsServer, des variables globales statiques ont été introduites. Certaines d'entre elles gagneraient à être utilisées dans un fichier de config. Ce fichier de config permetrrait également de faciliter l'automatisation évoquée plus tôt.

1.3 Qualité du code

En ce qui concerne la qualité du code, nous avons trouvé peu de commentaires dans certaines méthodes qui en mériteraient davantage au vu de leur complexité pour quelqu'un d'extérieur au code implémenté.

De plus, les classes n'ont pas été documentées.

Nous avons également pu observer la présence d'une méthode sendFragment qui permet d'éviter une très grande redondance de code, en particulier sur la méthode HdfsWrite du HdfsClient. Cependant, nous pensons que du point de vu de la redondance, le code peut être encore plus optimisé et surtout dans la méthde HdfsRead du HdfsClient

.

2 Synthèse

2.1 Correction

Nous avons pu tester le lancement du plusieurs serveurs ainsi que l'exécution d'un client. Les serveurs semblent se lancer correctement et le client semble remplir sa tâche.

Cependant, les méthodes de la classe HdfsServer ont été spécifiées comme étant statiques. Cela pourrait donner lieu à des situtations qui iraient à l'encontre de norte but. En effet, un client pourrait directement écrire dans HiDFS en excluant la communication TCP et le démarrage des serveurs, ce qui ne nous semble peu robuste.

2.2 Complétude

L'implémentation proposée semble complète. Presque toutes les spécifications semblent avoir été traitées ou abordées. Certaines exceptions ne sont pas attrapées, comme nous l'avons évoqué au 1.1.

2.3 Pertinence

Le travail effectué lors de ce premier rendu semble pertinent et correspond aux attentes d'un premier rendu, à savoir une architecture globalement comprise, et une idée directrice respectée.

2.4 Cohérence

Le lancement des serveurs à la main semble être légèrement incohérent par rapport aux attentes du projet.

2.5 Pistes d'amélioration

Les pistes d'amélioration que nous proposons sont :

AMÉLIORATION	MOTIVATION
Création d'un fichier de config	Permet d'éliminer l'utilisation de variables statiques à l'intérieur de la classe HdfsServer
Automatisation des lancements des serveurs	Perment de lancer un grand nombre de serveurs sans intervention manuelle
Rattraper les excéptions qui ne le sont pas encore	Permet d'améliorer la robustesse du HDFS final