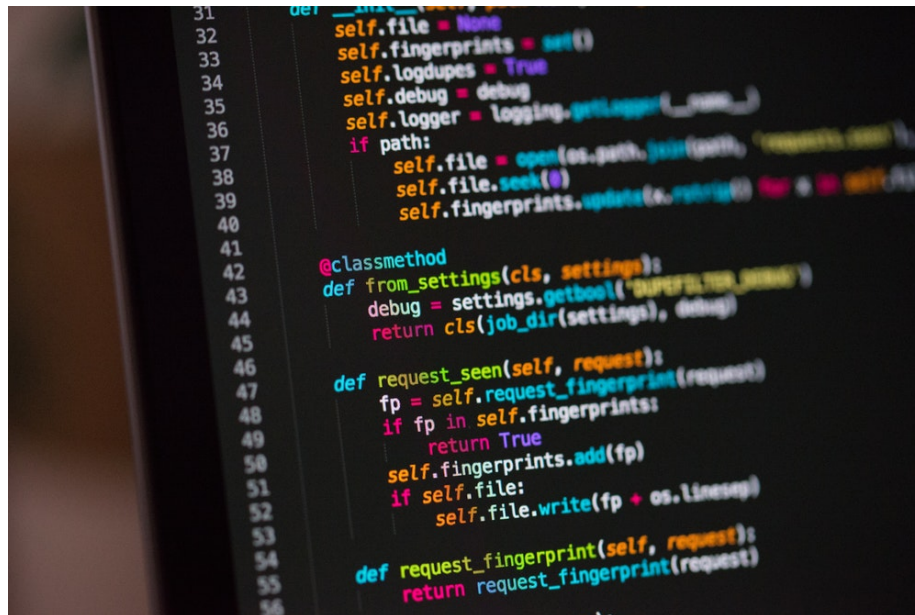


Rapport Final



Hamza Mouddene

Manal Hajji

Benoît Thomas

Bao Long Nguyen

3 mai 2021

Table des matières

1	Une version améliorée de Hadoop	1
1.1	HDFS	1
1.2	Hadoop	1
2	Déploiement	1
3	Une autre application	1
4	Étude	2

1 Une version améliorée de Hidoop

Nous avons décidé d'apporter une amélioration sur la version antérieure de Hidoop selon l'aspect tolérance aux pannes.

1.1 HDFS

1.2 Hidoop

Le but était de résister à différents types de pannes :

- Détection des pannes : Coté Hidoop, le défi était de détecter les pannes aux niveaux des Workers et startJob, ceci a été mis en place via un mécanisme de heart-beat, nous avons profité du fait que la classe Worker est un serveur RMI, ainsi que nous avons transformé la classe Job en un serveur RMI, puis nous avons créé une classe HeartBeatSensor qui est un client RMI et qui appelle la méthode beat de l'interface Worker et Job pour avoir l'état des workers et startJob.
- Panne de processus : Grâce à la mise en place de la détection des pannes, à chaque fois, quand il y a un worker qui tombe en pannes, le HeartBeatSensor récupère son état avant qu'il crash, et en fonction de son état ils relances les maps impactés. de la même manière pour startJob où HeartBeatSensor pourra reprendre l'exécution sans ré-exécution de tous les maps.

2 Déploiement

Un script a été conçu par notre soin pour rendre le déploiement aussi simple que possible, je tiens à préciser qu'il faudra juste installer terminator avec la commande : `sudo apt install terminator`, après vous aurez juste à exécuter le `run.sh`, qui vous proposera de renseigner votre login, puis le nombre de noeud que vous souhaitez utiliser, puis un autre script démarre pour vous demander ce que vous voulez faire, un menu textuel propose de faire un `sois` un `HDFSList` ou une `HDFSRead` ou `HDFSDelete` ou bien faire un `HDFSWrite` qui genere un `data.txt` file, après si vous lancez votre application puis quand c'est fini vous pourrez ré-exécuter encore une fois le script.

3 Une autre application

Nous avons choisi de faire une autre application réalisant l'algorithme X de Knuth. Cet algorithme récursif non déterministe de parcours à profondeur et de retour sur trace permet de trouver des solutions au problème de la couverture exact, représenté sous la forme d'une matrice contenant des 0 et des 1.

La résolution se faisant sous forme d'un arbre de recherche, le travail peut aisément être découpé en plusieurs morceaux indépendants, nécessaire à l'utilisation de MapReduce.

Le découpage en tâches se fait au niveau du non déterminisme de l'algorithme. Après avoir trouvé la première colonne c contenant le minimum de 1 de la matrice M , on doit appliquer l'algorithme à chacune des lignes l tel que $A_{lj} = 1$. Ici, ces applications étant indépendantes les unes des autres, le travail peut être donné à des machines différentes. Il suffit ensuite de concaténer les listes de solutions trouvées par chaque machine pour avoir l'ensemble des solutions du problème initial.

4 Étude

L'évaluation de la tolérance aux pannes de MapReduce requiert plusieurs éléments tels que définir les pannes à étudier, injecter ces pannes dans un système MapReduce en cours d'exécution et analyser l'impact de ces pannes sur la fiabilité, la disponibilité et la performance du système.

Modèles de pannes dans MapReduce

- Panne logicielle d'un processus : un processus peut s'arrêter suite à une erreur ou une exception dans la fonction map ou la fonction reduce. Le processus est considéré alors comme défaillant et son job également.
- Arrêt de processus : le processus exécutant une tâche peut s'arrêter inopinément.
- Blocage de processus : une tâche map ou reduce dont l'exécution reste bloquée sans progrès durant un certain temps est considérée comme défaillante. (le blocage est une pause prolongée lors de l'exécution de map ou de reduce)

Le défi réside alors dans la capacité de job et worker de détecter le problème au cas où un processus est tué et relancer le MapReduce.

Pour tester, on fait un kill à un nombre de workers qui tournent et observe son impact sur le fonctionnement des autres workers et leurs jobs. On constate que startJob est relancé et aucun des workers n'est affecté.