Rapport
Données répartiers

AUGEREAU Robin - AUBRY Jules

Département Sciences du Numérique - Deuxième année
2023-2024

1 Contributions

Nous avons réparti le travail en 2 équitablement :

— Jules a codé la partie Map-Reduce-Job Launcher

— Robin a codé la partie HDFS ainsi que les scripts permettant de lancer et utiliser Hagidoop

2 Manuel d'utilisation

On peut utiliser les codes *HdfsClient*, *HdfsServer*, *JobLauncher*, *Worker* indépendemment. A des fins de test, le dossier *scripts* contient 4 codes :

— *aio.sh* qui permet de lancer une instance de *HdfsServer* et *WorkerImpl* sur chacune des machines décrites dans le fichier de configuration situe dans *src/config/main.cfg*, apres avoir copié collé le projet Java et compilé à distance sur l'une des machines

— *down.sh* qui permet de stopper les instances de *WorkerImpl* et *HdfsServer* sur chacune des machines, ainsi que de nettoyer le dossier */tmp/data*

NB : ces scripts ne marchent qu'avec les machines de l'ENSEEIHT

Il est possible de modifier le fichier de configuration :

— la première ligne contient le nom des machines

— la seconde ligne contient les ports HDFS des serveurs

— la troisième ligne contient les ports RMI des workers

— la quatrième ligne contient la taille des fragments en caractères

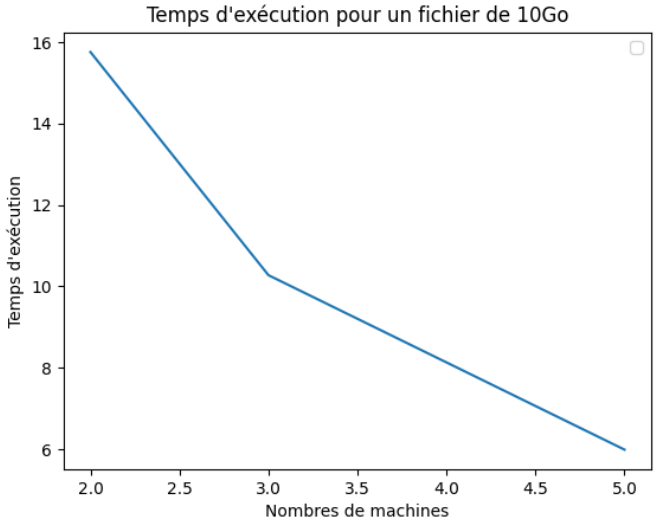
Sur chacune des lignes, les valeurs doivent êtres séparées par des virgules et sans espaces.

Il est nécéssaire de modifier le nom de la machine principale dans *JobLauncher.java* ligne 25 avec celle à partir de laquelle vous lancez les srcipts d'utilisation.

3 Évaluations de performances

Afin d'évaluer les performances de ce système de répartition de données, nous avons effectué une batterie de tests :

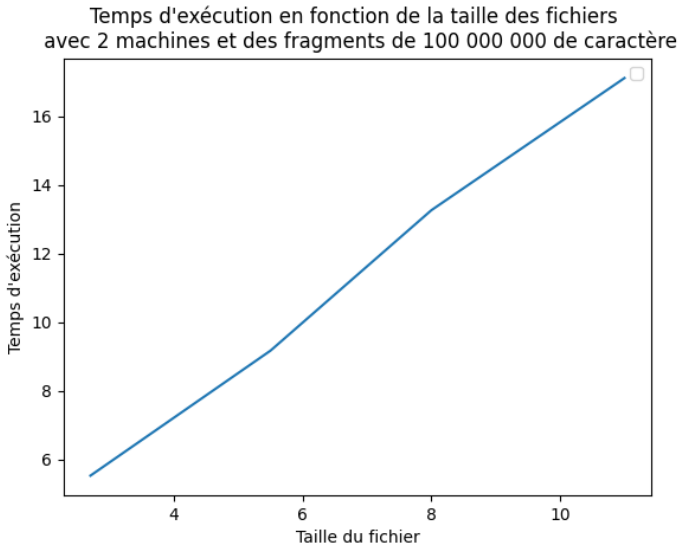
3.1 Variations du nombre de machine



Sur cette figure, la courbe représente la variation du temps d'exécution de Hagidoop en fonction du nombre de machines utilisées pour la distribution des fragments. La taille des fichiers et des fragments a

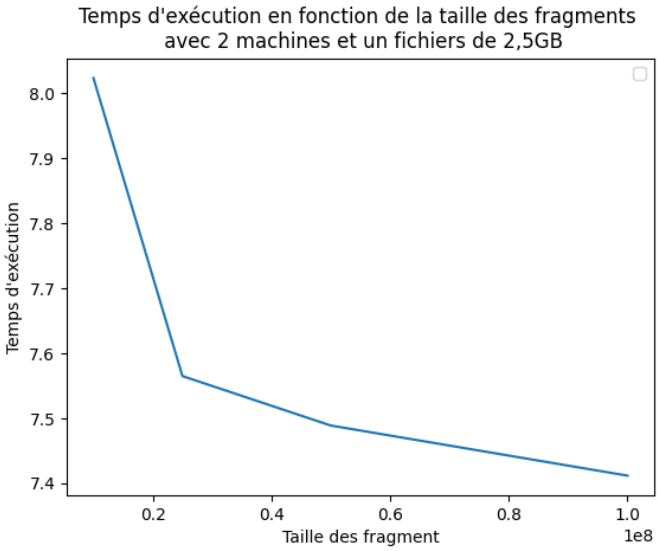
été maintenue constante. On observe que la répartition du travail sur un plus grand nombre d'ordinateurs entraîne une exécution plus rapide. Cela confirme la pertinence de l'adage "diviser pour mieux régner".

3.2 Variations de la taille du fichier



Sur ce graphique, l'évolution de la taille des fichiers principaux envoyés à HDFS est représentée. Une tendance linéaire est observable, ce qui semble logique étant donné que le temps d'exécution évolue de manière linéaire en fonction de la taille des fichiers. Plus le fichier est grand, plus le temps d'exécution est prolongé. Le coefficient directeur de la courbe, calculé à 1.303, confirme cette relation linéaire entre la taille des fichiers et le temps d'exécution.

3.3 Variations de la taille des fragments



Sur cette représentation graphique, la variation de la taille des fragments envoyés pour traitement aux machines est examinée. On observe que plus la taille des fragments est importante, plus l'exécution est rapide. On peut interpréter cela en considérant que l'augmentation de la taille des fragments entraîne une diminution du nombre de threads sur les différentes machines, ce qui libère davantage de ressources. Cependant, il est intéressant de noter que simplement créer plus de threads de taille réduite n'accélère

pas nécessairement l'exécution. Il est crucial de trouver un équilibre optimal entre le nombre de threads et la taille des fragments.