Projet INF727 – Systèmes Répartis

Sara Boutigny

2021-2022

# Code source

# Structure du code optimisé

Dans les étapes qui vont suivre, on considère la machine personnelle comme faisant partie du cluster. C’est cette machine que l’on appellera « master ». Celle-ci exécute les principales étapes du programme. Les passages en surbrillance sont ceux qui constituent une amélioration par rapport aux 13 étapes constitutives du projet.

**Etape 0 : Clean**

Effacement de tous les dossiers /tmp/<login> sur les machines utilisées.

**Etape 1 : Deploy**

* Création du dossier dans /tmp et des sous-dossiers sur chaque machine. (P)[[1]](#footnote-2)
* Envoi des scripts python sur chaque machine. (P)

Notons que chaque commande identique effectuée sur plusieurs machine est parallélisée, mais que le programme attend ensuite la fin d’éxécution avant d’en relancer d’autres.

**Etape 2 : Split**

* Le master créé les splits et les déploie simultanément sur chaque machine. (P)
* Le master attend la fin du dépoiement sur toutes les machines avant de passer à la suite. Pour cela, on lance les commandes ssh/scp successivement grâce à la méthode Popen de subprocess et seulement ensuite, on applique la méthode communicate aux process en cours d’éxécution.

**Etape 3 : Map**

* Le master organise l’exécution simultanée du script slave.py sur chacune des machine afin de créer les map. Ces map sont au format .txt. (P)
* La master attend la fin d’éxécution de tous les slaves (de façon similaire à l’étape précédente)

**Etape 4 : Shuffle**

* Le master organise l’éxécution simultanée du script slave.py sur chacune des machines (P) afin de créer les shuffle. Le fait de créer un fichier par hashcode était gourmand en mémoire, on décide donc de créer autant de fichiers que de machines. Cela réduit drastiquement la mémoire utilisée et le temps d’éxécution. On utilise le hashcode seulement pour déterminer quel clé est envoyée vers quelle machine. Les clés sont stockées dans des dataframes pour faciliter le traitement puis enregistrées au format pickle. Chaque DataFrame correspond à un fichier pickle et se réfère à une seule machine, sa machine de destination.
* On note que cette étape est parmi les plus chronophages du map reduce : plus le cluster est grand, plus il y a de fichiers à créer et à envoyer aux machines. On bride alors le nombre de fichiers shuffle à 4 machines maximum. Le reste des étapes est donc parallélisé sur maximum 4 machines.
* Chaque fichier pickle est envoyé à sa machine de destination. (P)

**Etape 5 : Reduce**

* Le master organise l’éxécution simultanée du script slave.py sur toutes les machines afin de créer les fichiers reduce (P). La tâche est optimisée puisque l’on transforme les fichiers pickle en DataFrame, conservant ainsi leur format d’origine et nous donnant accès aux méthodes de la classe DataFrame. En un simple groupby, on obtient alors le wordcount partiel pour la machine concernée, sous forme d’un DataFrame, puis enregistré au format pickle.
* Le master attend la fin de l’exécution de l’étape précédente puis envoie les wordcounts partiels à la machine 0 (P), qui sera chargée du wordcount final.

**Etape 6 : WordCount final**

* La machine 0 recueille tous les wordcount partiels et exécute le wordcount final à partir de tous les fichiers pickle reçus. Une fois de plus, l’objet DataFrame et ses méthodes aident à obtenir le résultat final en un temps optimal. Cette étape n’est pas parallélisée puisque s’effectue sur une seule machine.
* Le résultat du wordcount est au format .txt dans le dossier « wordcount » de la machine 0.

# Temps d’éxécution, taille du fichier input, et nombre de machines

Pour cette étape, on utilise des fichiers .txt de différentes tailles, allant de 1Ko à 6Go.

"C:\Users\Sara\OneDrive - Université Paris-Dauphine\MS Big Data\GitHub\MS-Big-Data-2021-2022\INF727 - Systèmes Répartis\MapReduce\venv\Scripts\python.exe" "C:/Users/Sara/OneDrive - Université Paris-Dauphine/MS Big Data/GitHub/MS-Big-Data-2021-2022/INF727 - Systèmes Répartis/MapReduce/main.py"

D:/txt\_files/CC-MAIN-20170322212949-00140-ip-10-233-31-227.ec2.internal.warc.wet 398783064

\*\*\* DEPLOYING...

nombre de machines : 20

dict\_keys(['tp-4b01-38', 'tp-4b01-42', 'tp-4b01-41', 'tp-4b01-40', 'tp-4b01-39', 'tp-3b01-06', 'tp-3b01-05', 'tp-3b01-04', 'tp-3b01-03', 'tp-3b01-02', 'tp-3b01-01', 'tp-3b01-07', 'tp-3b01-08', 'tp-3b01-09', 'tp-3b01-10', 'tp-3b01-11', 'tp-3b01-12', 'tp-3b01-13', 'tp-3b01-14', 'tp-3b01-15'])

(Deploy time : 0:01:55.939433)

Split time : 0:00:42.373884

Mapping time : 0:00:11.269846

Shuffle time : 0:00:14.472555

Reduce time : 0:00:12.163147

Wordcount final time : 0:00:30.006628

\*\*\* CLEANING...

D:/txt\_files/random\_generated\_1Go.txt 1025449846

\*\*\* DEPLOYING...

nombre de machines : 20

dict\_keys(['tp-4b01-38', 'tp-4b01-42', 'tp-4b01-41', 'tp-4b01-40', 'tp-4b01-39', 'tp-3b01-06', 'tp-3b01-05', 'tp-3b01-04', 'tp-3b01-03', 'tp-3b01-02', 'tp-3b01-01', 'tp-3b01-07', 'tp-3b01-08', 'tp-3b01-09', 'tp-3b01-10', 'tp-3b01-11', 'tp-3b01-12', 'tp-3b01-13', 'tp-3b01-14', 'tp-3b01-15'])

(Deploy time : 0:00:45.466835)

Split time : 0:01:07.794042

Mapping time : 0:00:08.976116

Shuffle time : 0:00:23.010374

Reduce time : 0:00:18.642178

Wordcount final time : 0:00:09.704412

\*\*\* CLEANING...

D:/txt\_files/random\_generated\_3Go.txt 3243706723

\*\*\* DEPLOYING...

nombre de machines : 20

dict\_keys(['tp-4b01-38', 'tp-4b01-42', 'tp-4b01-41', 'tp-4b01-40', 'tp-4b01-39', 'tp-3b01-06', 'tp-3b01-05', 'tp-3b01-04', 'tp-3b01-03', 'tp-3b01-02', 'tp-3b01-01', 'tp-3b01-07', 'tp-3b01-08', 'tp-3b01-09', 'tp-3b01-10', 'tp-3b01-11', 'tp-3b01-12', 'tp-3b01-13', 'tp-3b01-14', 'tp-3b01-15'])

(Deploy time : 0:00:53.574359)

Split time : 0:03:08.505458

Mapping time : 0:00:16.935227

Shuffle time : 0:00:52.162326

Reduce time : 0:00:45.184425

Wordcount final time : 0:00:20.309324

\*\*\* CLEANING...

D:/txt\_files/random\_generated\_5Go.txt 5377347808

\*\*\* DEPLOYING...

nombre de machines : 20

dict\_keys(['tp-4b01-38', 'tp-4b01-42', 'tp-4b01-41', 'tp-4b01-40', 'tp-4b01-39', 'tp-3b01-06', 'tp-3b01-05', 'tp-3b01-04', 'tp-3b01-03', 'tp-3b01-02', 'tp-3b01-01', 'tp-3b01-07', 'tp-3b01-08', 'tp-3b01-09', 'tp-3b01-10', 'tp-3b01-11', 'tp-3b01-12', 'tp-3b01-13', 'tp-3b01-14', 'tp-3b01-15'])

(Deploy time : 0:00:47.361115)

Split time : 0:06:11.517268

Mapping time : 0:00:11.167081

Shuffle time : 0:01:03.094239

Reduce time : 0:00:18.780723

Wordcount final time : 0:00:09.082169

\*\*\* CLEANING...

D:/txt\_files/random\_generated\_6Go.txt 6412746603

\*\*\* DEPLOYING...

nombre de machines : 20

dict\_keys(['tp-4b01-38', 'tp-4b01-42', 'tp-4b01-41', 'tp-4b01-40', 'tp-4b01-39', 'tp-3b01-06', 'tp-3b01-05', 'tp-3b01-04', 'tp-3b01-03', 'tp-3b01-02', 'tp-3b01-01', 'tp-3b01-07', 'tp-3b01-08', 'tp-3b01-09', 'tp-3b01-10', 'tp-3b01-11', 'tp-3b01-12', 'tp-3b01-13', 'tp-3b01-14', 'tp-3b01-15'])

(Deploy time : 0:00:45.266801)

Split time : 0:06:59.225776

Mapping time : 0:00:14.888865

Shuffle time : 0:01:00.648691

Reduce time : 0:00:17.592675

Wordcount final time : 0:00:09.295328

\*\*\* CLEANING...

D:/txt\_files/CC-MAIN-20170322212949-00140-ip-10-233-31-227.ec2.internal.warc.wet 398783064

\*\*\* DEPLOYING...

1. (P) : action parallélisée sur les différentes machines [↑](#footnote-ref-2)