

TP 2 : Map-Reduce (Part 1)

idir.benouaret@epita.fr

Partie TD

pas d'implémentation, réfléchir aux exercices et écrire un pseudo-code.

EXERCICE I : Moyenne

On considère un fichier CSV contenant des mesures de température. Chaque ligne du fichier correspond à "year, month, temperature".

Q1 – Écrire un algorithme Map-Reduce capable de calculer la température moyenne pour chaque année

EXERCICE II : Histogramme

Supposons que vous avez un fichier très volumineux, qui contient la liste de toutes les villes de la planète ainsi que la population. Par exemple une ligne correspond à : `lyon,1416545`

Q1 – Écrivez un programme MapReduce qui calcule l'histogramme de fréquence des villes

On utilisera une échelle logarithmique pour déterminer les classes d'équivalences des villes. Soit $v1$ et $v2$ deux villes $v1$ et $v2$ sont dans la même classe d'équivalence si et seulement si $\text{int}(\log(\text{pop}(v1))) == \text{int}(\log(\text{pop}(v2)))$.

Le fichier généré par votre "reducer" devrait ressembler à cela :

```
10 34
100 85
1000 9429
10000 13065
100000 8830
1000000 979
10000000 57
```

Ce qui veut dire : il y a 34 villes avec entre 0 et 10 habitants, 9429 avec une population entre 100 et 1000, etc.

Partie TP

Pour cette partie d'abord réfléchir aux pseudo-codes. Puis implémenter avec la librairie de map-reduce MRJOB :

EXERCICE III : Découverte de la librairie MRJOB

<https://mrjob.readthedocs.io/>

Q1 – Installer mrjob

Q2 – lire, tester les programmes de bases de mrjob :

<https://mrjob.readthedocs.io/en/latest/guides/quickstart.html>

EXERCICE IV : WordCount

Télécharger les fichiers suivants et les mettre sur votre répertoire hdfs.

https://gitlab.cri.epita.fr/idir.benouaret/bda_datasets/-/tree/main/Livre

Q1 – Implémenter le programme Mapreduce qui permet de compter le nombre d'apparition de chaque mot dans le texte.

Q2 – Ecrire un programme qui compte le nombre exact de caractères, de mots et de ligne dans un fichier texte.

Q3 – Ecrire un programme qui retourne le mot plus long sur chaque ligne

Q4 – Écrire un programme qui retourne le mot le plus long du texte et les 10 mots les plus longs.

EXERCICE V : Exercices sur le fichier d'arbres

Vous allez écrire plusieurs programmes *MapReduce* sur le fichier des arbres remarquables à Paris, qui se trouve ici :

- https://gitlab.cri.epita.fr/idir.benouaret/bda_datasets/-/blob/main/arbres.csv

Q1 – Tout d'abord, placer ce fichier sur un répertoire HDFS nommé data. i.e. le fichier doit se trouver sur hdfs : `:///user/votrelogin/data/arbres.csv`

Pour chacun des programmes suivants que vous devez écrire, testez sur votre machine locale puis sur

hadoop.

Q2 – Écrire un programme *mapreduce* qui affiche la liste de tous les arrondissement distincts contenant des arbres dans ce fichier. Il y en a forcément moins de 20 sauf s’il existe des erreurs de saisie.

Q3 – Écrire un programme qui calcule le nombre d’arbres existants pour chacun des genres. Par exemple, il y a 32 *Platanus*, 11 *Quercus*, etc.

Q4 – Écrire un programme qui la hauteur du plus grand arbre de chaque genre. par exemple, le plus haut *Zelkova* fait 26 mètres

Q5 – Écrire un programme qui affiche l’arrondissement où se trouve le plus haut arbre

Q6 – Écrire un programme qui affiche l’arrondissement qui contient le plus d’arbres