COLLÈGE ROSEMONT

Valorisation de données

Par : Abderrazak Sahraoui

Sommaire

- Programmation Python pour Hadoop
- Installation MRJob
- Exemples de programmes MapReduce

Programmation MapReduce en Python

Why mrjob?

Overview

mrjob is the easiest route to writing Python programs that run on Hadoop. If you use mrjob, you'll be able to test your code locally without installing Hadoop or run it on a cluster of your choice.

Additionally, mrjob has extensive integration with Amazon Elastic MapReduce. Once you're set up, it's as easy to run your job in the cloud as it is to run it on your laptop.

https://mrjob.readthedocs.io/en/latest/index.html

- Le framework **Hadoop** a été écrit en **Java**. Cependant, Il est tout a fait possible d'écrire des **job MapReduce** dans un langage autre que Java.
- **Python** est largement utilisé pour faire des programmes pour Hadoop. Python possède un module (bibliothèque) appelé **mrjob**. Cette bibliothèque permet de tester sur un ordinateur local (pc) les programmes Python écrits pour Hadoop.
- L'installation de **mrjob** sur un poste local se fait par le programme **pip**

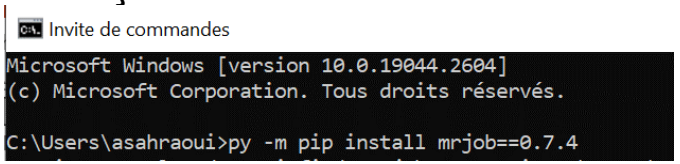
Installation mrjob

pip

pip is the package installer for Python. You can use it to install packages from the Python Package Index and other indexes.

https://pip.pypa.io/en/latest/

- L'installation de **mrjob** sur un poste local se fait par l'invite de commande de la façon suivante



- On choisit ici une version compatible 0.7.4 avec notre plateforme big data HDP 2.5
- L'installation du module **pathlib** permet de gérer les chemins d'accès aux fichiers sur tout système d'exploitation

```
C:\Users\asahraoui>py -m pip install pathlib
```

Exemple de job MapReduce en Python

```python
""" Programme pour compter les mots d'un texte et afficher pour chaque mot le nombre d'occurence"""
from mrjob.job import MRJob
from mrjob.step import MRStep

class CompteMots(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_mots,
                   reducer=self.reducer_compte_mots)       ]

    def mapper_get_mots(self, _, line):
        line = line.strip()
        line = line.lower()
        line = line.replace(","," ")
        line = line.replace("."," ")
        line = line.replace(";"," ")
        line = line.replace("?"," ")
        line = line.replace("!"," ")
        line = line.replace(":"," ")

        mots = line.split()
        for mot in mots:
            yield mot, 1

    def reducer_compte_mots(self, mot, uns):
        yield mot, sum(uns)

if __name__ == '__main__':
    CompteMots.run()
```

Pour executer ce programme, on utilise un terminal ou l'invite de commande et on soumet la ligne suivante :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\asahraoui> cd documents\valorisation
PS C:\Users\asahraoui\documents\valorisation> Py CompteMots.py VictorHugo.txt
```

Exemple de fichier texte à traiter

C: > Users > asahraoui > Documents > Valorisation > VictorHugo.txt

```
JeanThe Project Gutenberg EBook of Les misérables Tome I, by Victor Hugo

...

L'hôte alors se pencha à son oreille, et lui dit d'un accent qui le fit
tressaillir:

--Allez-vous en.

Le voyageur était courbé en cet instant et poussait quelques braises
dans le feu avec le bout ferré de son bâton, il se retourna vivement,
et, comme il ouvrait la bouche pour répliquer, l'hôte le regarda
fixement et ajouta toujours à voix basse:

--Tenez, assez de paroles comme cela. Voulez-vous que je vous dise votre
nom? Vous vous appelez Jean Valjean. Maintenant voulez-vous que je vous
dise qui vous êtes? En vous voyant entrer, je me suis douté de quelque
chose, j'ai envoyé à la mairie, et voici ce qu'on m'a répondu.
Savez-vous lire?

En parlant ainsi il tendait à l'étranger, tout déplié, le papier qui
venait de voyager de l'auberge à la mairie, et de la mairie à l'auberge.
L'homme y jeta un regard. L'aubergiste reprit après un silence:

--J'ai l'habitude d'être poli avec tout le monde. Allez-vous-en.

L'homme baissa la tête, ramassa le sac qu'il avait déposé à terre, et
s'en alla. Il prit la grande rue. Il marchait devant lui au hasard,
rasant de près les maisons, comme un homme humilié et triste. Il ne se
```

https://www.gutenberg.org/cache/epub/17489/pg17489.txt

Exemple de résultats sur un petit extrait

```
"demi-heure"    1
"derri\u00e8re" 1
"des"   2
"dessinait"     1
"deux"  1
"devant"        2
"devin\u00e9"   1
"dise"  2
"distinguer"    1
"dit"   4
"doigt" 1
"donne" 2
"doublant"      1
"douceur"       1
"dout\u00e9"    1
"du"    7
"ebook" 1
"effet" 1
"elle"  1
"en"    8
"encore"        1
"entour\u00e9"  1
"entourait"     1
"entra" 1
"entre" 2
"entrer"        2
"envoy\u00e9"   1
"escoublon)"    1
"esp\u00e8ce"   1
"est"   2
"et"    25
"eux"   1
"faim"  1
"faisait"       3
"fatigu\u00e9"  1
"fatigue"       2
```

Exemple 2 : Fréquences triées

```python
""" Programme pour compter les mots d'un texte et et les afficher en ordre croissant selon la fréquence"""
from mrjob.job import MRJob
from mrjob.step import MRStep

class CompteMots(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_mots,
                   reducer=self.reducer_compte_mots),
            MRStep(reducer=self.reducer_frequence_mots)    ]
    def mapper_get_mots(self, _, line):
        line = line.strip()
        line = line.lower()
        line = line.replace(","," ")
        line = line.replace("."," ")
        line = line.replace(";"," ")
        line = line.replace("?"," ")
        line = line.replace("!"," ")
        line = line.replace(":"," ")

        mots = line.split()
        for mot in mots:
            yield mot, 1
    def reducer_compte_mots(self, mot, uns):
        yield  str(sum(uns)).zfill(3), mot
    def reducer_frequence_mots(self, mot, frequences):
        for frequence in frequences :
            yield  frequence, mot
if __name__ == '__main__':
    CompteMots.run()
```

Fréquences triées

```
"cabaret"       "006"
"feu"   "006"
"l'homme"       "006"
"ne"    "006"
"pas"   "006"
"quelque"       "006"
"sur"   "006"
"du"    "007"
"rue"   "007"
"dans"  "008"
"en"    "008"
"les"   "008"
"lui"   "008"
"que"   "008"
"se"    "008"
"vous"  "008"
"par"   "009"
"son"   "009"
"qui"   "010"
"un"    "011"
"une"   "013"
"\u00e0"        "017"
"il"    "020"
"et"    "025"
"le"    "029"
"la"    "033"
"de"    "039"
```