

# Développer des programmes map reduce avec Python

**Exercice 1.** Créer un nouveau fichier et créer votre premier programme permettant de compter les fréquences de chaque rating (note: nombre d'étoiles) utilisés pour noter les films (movies) dans le fichier u.data

*Extrait du fichier u.data (l'entête est ajoutée pour clarification)*

| USER ID | MOVIE ID | RATING | TIMESTAMP |
|---------|----------|--------|-----------|
| 196     | 242      | 3      | 881250949 |
| 186     | 302      | 3      | 891717742 |
| 22      | 377      | 1      | 878887116 |
| 244     | 51       | 2      | 880606923 |
| 166     | 346      | 1      | 886397596 |
| 298     | 474      | 4      | 884182806 |
| 115     | 265      | 2      | 881171488 |

2

Enregistrer le programme dans le dossier  
“d:\BigData” que vous venez de créer en donnant au  
programme le nom “RatingCounter”

Pour exécuter le programme, taper dans l'éditeur  
inférieur de spyder:

```
!python RatingCounter.py ml-100k\u.data
```

**Note:** par défaut, l'environnement d'exécution pour  
python est c:\users\computer\_name

Il faut modifier l'environnement de travail vers d:\bigdata  
en tapant dans l'éditeur inférieur de spyder :

```
cd d:\bigdata
```

**Exercice 2.** Ecrire un programme map-reduce qui cherche la température maximale pour chaque région dans un ensemble de données réel de 1825 enregistrements dans un fichier apple 1800.csv.

Ouvrez le fichier 1800.csv et examiner son contenu

```
ITE00100554,18000101,TMAX,-75,,,E,  
ITE00100554,18000101,TMIN,-148,,,E,  
GM000010962,18000101,PRCP,0,,,E,  
EZE00100082,18000101,TMAX,-86,,,E,  
EZE00100082,18000101,TMIN,-135,,,E,
```

**Exercice 3.** Modifier le programme map-reduce de l'exercice 2 et trouver le maximum de température en fahrenheit pour chaque région.

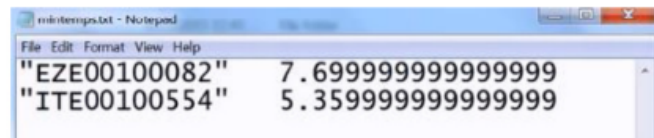
Nous avons besoin d'une **fonction MakeFahrenheit** qui convertit les températures du celsius à fahrenheit. La conversion est effectuée comme suit:

$\text{fahrenheit} = \text{Celcius} * 1,8 + 32$

La température existant dans le fichier est sous forme celsius\*10

**Exercice 4.** écrire un programme map-reduce qui cherche la température minimale en fahrenheit par region en utilisant le même fichier de l'exercice 3.

## Résultats



**Exercice 5.** Ecrire un programme map-reduce qui calculi la moyenne des amis par âge.

## ***Extrait du fichier friends.csv***    ***Quels sont les attributs?***

```
0,Will,33,385
1,Jean-Luc,26,2
2,Hugh,55,221
3,Deanna,40,465
4,Quark,68,21
5,Weyoun,59,318
6,Gowron,37,220
7,Will,54,307
8,Jadzia,38,380
9,Hugh,27,181
10,Odo,53,191
```

User ID, Name, Age, Number  
of Friends

**Exercise 6.** Revenons au programme qui compte le nombre de mots dans un livre et examinons le résultat.

```
File Edit Format View Help
"by." 1
"c-suite." 1
"c#" 1
"c++" 1
"cabl" 1
"cache" 2
"aching" 1
"caffeine-fueled" 1
"calculating" 1
"calculations" 1
"calendar" 1
"california" 1
"california," 1
"call" 8
"call." 1
"called" 9
"calling" 1
"calls" 1
"calls," 3
"came" 11
"campaign" 17
"campaign," 2
"campaign." 1
"campaigns" 12
"campaigns," 3
"campaigns,\" 1
```

```
wordcount.txt - Notepad
File Edit Format View Help
"campaigns," 3
"campaigns,\" 1
"campaigns." 5
"can't" 22
"can" 340
"can," 3
"can." 6
"canada," 1
"cannot" 9
"capabilities" 1
"capital" 4
"capital," 1
"capitalists" 2
"capture" 2
"capture." 1
"car" 2
"car)," 1
"car,)" 1
"car?" 2
"card" 1
"card," 1
"cardboard" 1
"cards" 1
"cards." 1
"care" 22
"care." 1
```

Remarquons que le mot « campaign » a une fréquence 17, suivi du mot « campaign, » de fréquence 2 et « campaign. » de fréquence 1. Il s'agit du même mot mais il y'a les signes de ponctuations qui rendent les mots différents.

24

Améliorer le programme en ajoutant une expression régulière (RE) qui tient en considération que le mot est constitué uniquement de caractères mots (lettres, chiffres et \_) et élimine les signes de ponctuation, etc.:

`WORD_REGEX=re.compile(r"[\w']+")`

← script définissant une RE

Doit être avant la définition de classe

`\w`: caractère mot: pour les caractères alphanumériques et \_  
équivalent à l'ensemble `[a-zA-Z0-9_]`

`\w'`: considérer les caractères mots en ignorant les ponctuation, les espaces, etc.

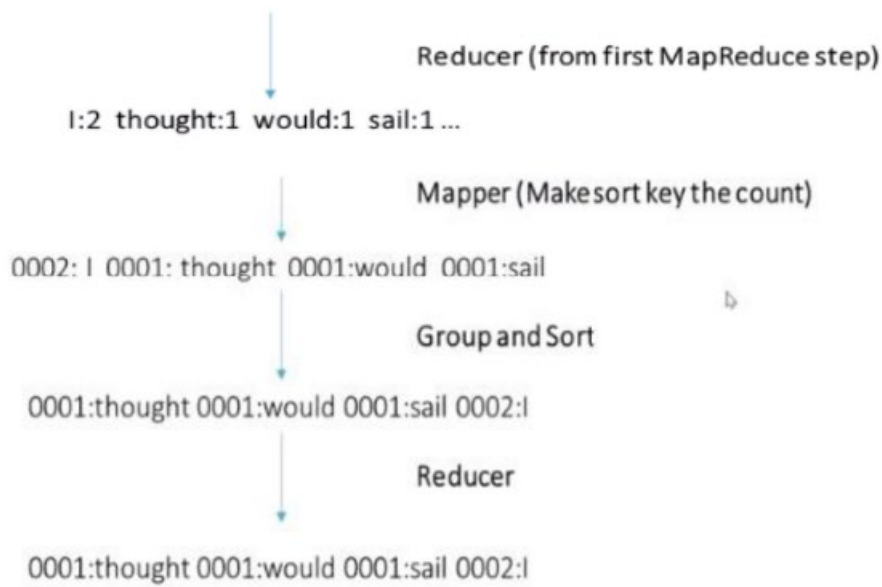
`+`: un ou plusieurs répétitions du RE

25

## Utilisation du Steps dans map reduce

**Exercise 7.** Le même programme changera de résultat qui sera **trié par la frequences de mots**.

Trier les résultats finaux: enchaîner les étapes (Steps)



- Nous avons un « step job », il faut ajouter l'instruction suivante après l'importation de MRJob:

```
from mrjob.Step import MRStep
```

- Nous avons une **nouvelle fonction** appliquée **steps**
- Au lieu d'un mapper et un reducer, nous avons deux mappers et deux reducers.

**La première étape map-reduce** (1<sup>ère</sup> MRStep):

- Fonction `mapper-get-words` pour le mapper (le même code du précédent mapper)
- Fonction `reducer-count-words` pour le reducer (le même code du précédent reducer)



## La seconde étape map-reduce (2<sup>ème</sup> MRStep):

La sortie du premier reducer, sera l'entrée du deuxième **mapper** dans lequel la **clé** sera **word** (le mot) et count **count** est la fréquence du mot. La sortie de ce mapper sera **(count, word)**:

```
yield '%04d'% int(count), word
```

Cast **count** into **int**

Présenter **count** avec 4 digits, ajouter  
**0** à gauche si nécessaire

**Exercise 8.** Ecrire un programme map-reduce qui calcule le total de "order amounts" par un client (customer)

*Extrait du fichier customer-orders.csv*

*Quels sont les attributs?*

CUSTOMER, ITEM, ORDER AMOUNT

```
44,8602,37.19
35,5368,65.89
44,3391,40.64
47,6694,14.98
29,680,13.08
91,8900,24.59
70,3959,68.68
```

**Exercise 10.** Trier le résultat de l'exercice précédent suivant les totaux dépensés par les clients.

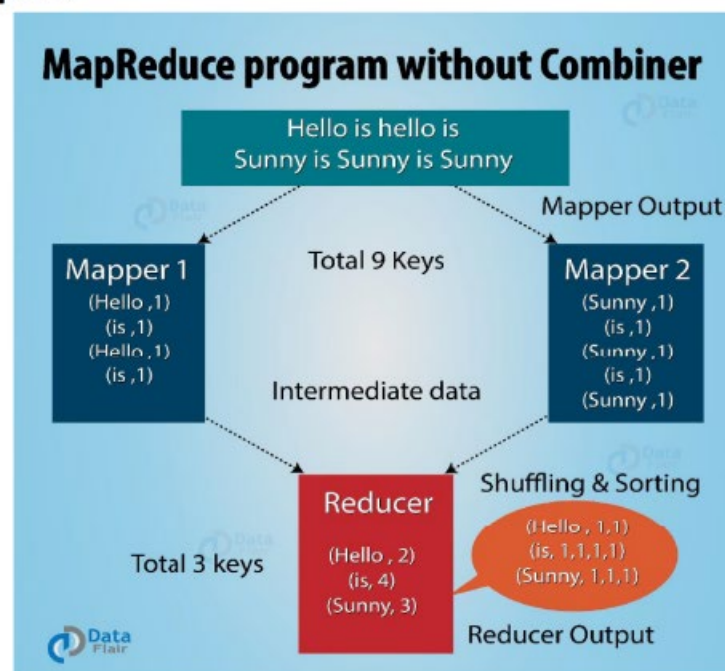
## Utiliser un combineur en map-reduce

Le combineur est utilisé entre le Mapper et le Reducer pour réduire le volume de données transféré entre eux. Toujours la sortie de la tâche map est large et les données transférées à la tâche reduce sont élevées.

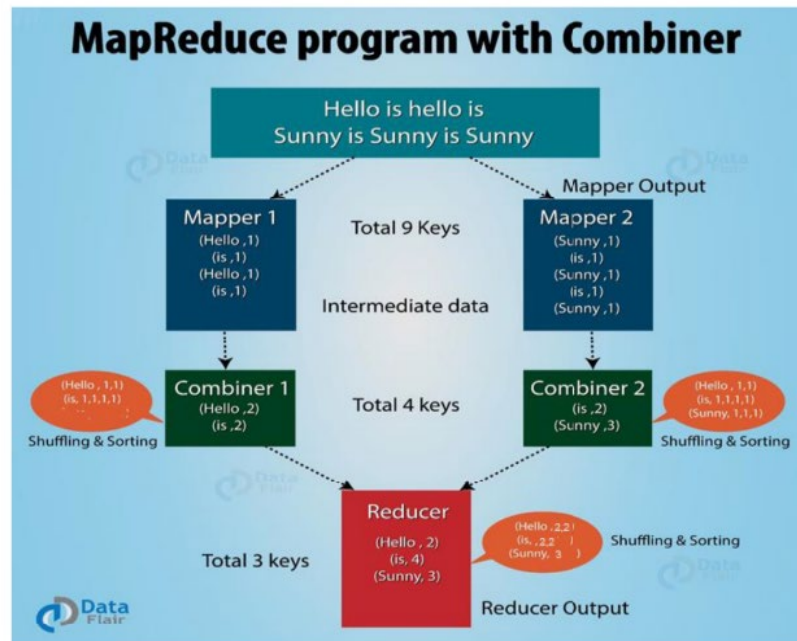
- Le combineur opère sur chaque clé sortie du mapper. Il doit avoir la même sortie clé, valeur que le reducer.
- Le combineur produit un résumé d'information à partir d'un large ensemble de données sorties du mapper.

47

### Exemple:



<https://data-flair.training/blogs/hadoop-combiner-tutorial/>



<https://data-flair.training/blogs/hadoop-combiner-tutorial/>

**Exercise 11.** réécrire le programme de fréquence de mot avec les combineurs

**Exercise 12.** écrire un programme map-reduce qui trouve le film le plus noté à partir du fichier u.data

## Most-Rated Movie

```
User, Movie, Rating, Timestamp
196 242 3 881250949
186 302 3 891717742
22 242 1 878887116
244 51 2 880606923
```

Mapper

```
242:1 302:1 242:1 51:1
```

Group and Sort

```
242:1,1 302:1 51:1
```

Reducer

```
None: (2, 242) None: (1, 302) None: (1, 51)
```

Reducer 2

```
2, 242
```

**Results:**

```
In [2]: !python MostPopularMovie.py ml-100k/u.data
583 "50"
```