Aymane Hinane 4IIR G5 emsi Centre

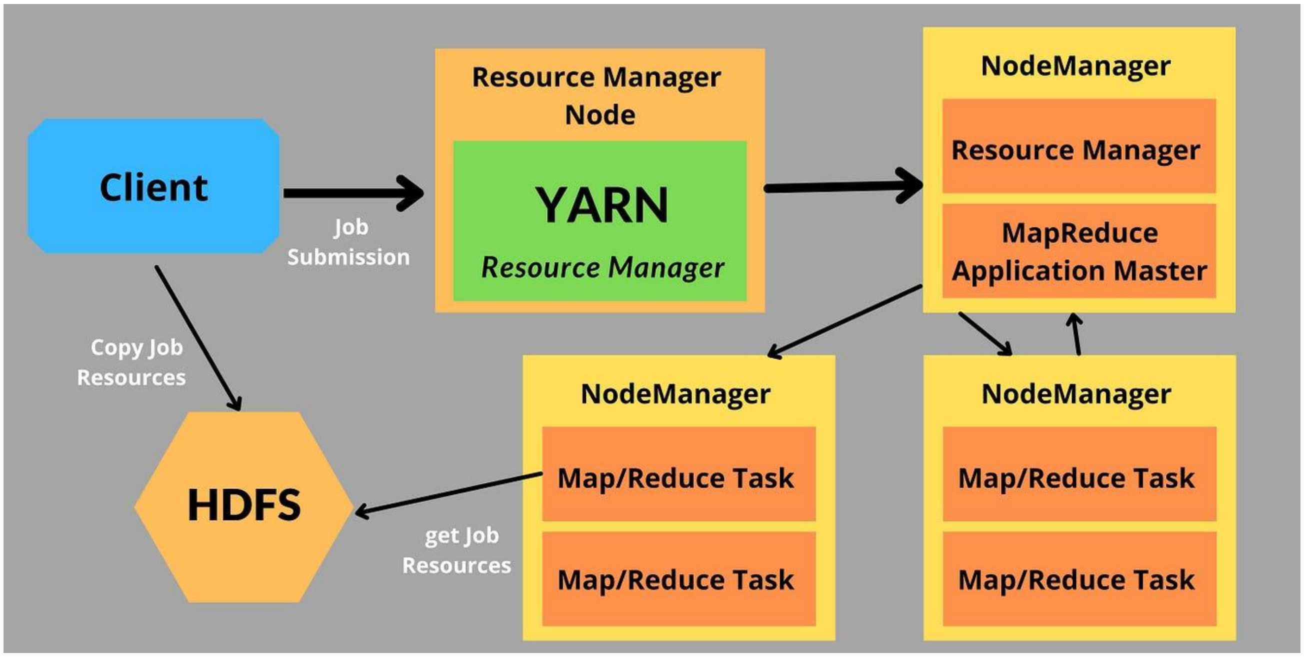
MapReduce with Python

MapReduce is a built-in programming model in Apache Hadoop. It will parallel process your data on the cluster.

How MapReduce Works

MapReduce will **transform** the data using **Map** by dividing data into key/value pairs, getting the output from a map as an input, and **aggregating** data together by **Reduce**.MapReduce will deal with all your cluster failures.

What’s happening under the hood



- Client node will submit MapReduce Jobs to The resource manager Hadoop YARN.

- Hadoop YARN resource managing and monitoring the clusters such as keeping track of available capacity of clusters, available clusters, etc. Hadoop Yarn will copy the needful data Hadoop Distribution File System(HTFS) in parallel.

- Next Node Manager will manage all the MapReduce jobs. MapReduce application master located in Node Manager will keep track of each of the Map and Reduce tasks and distribute it across the cluster with the help of YARN.

- Map and Reduce tasks connect with HDFS cluster to get needful data to process and output data.

Exemple d’explication:

MapReduce est écrit on java , mais il peut etre executer dans different langauge C++,Python…

Dans cette exemple nous allons utiliser Python avec MP job package

Step 1 :

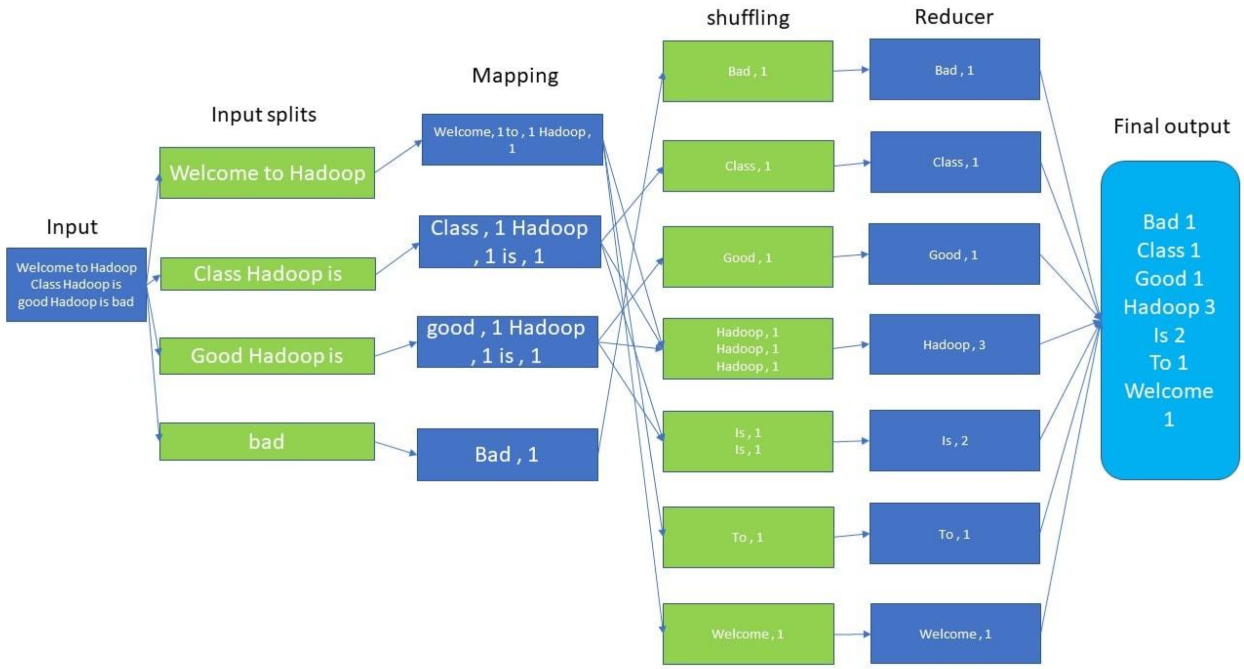
Mapper vat recevoir des lignes text d’un fichier et vat les convertir on (cle,valeur)

Step2 :

Shuffling and Sort est une etape ou MapReduce vat regrouper toute les value qui ont la meme cle   (cle , list(value))

Step 3 :

Reduce permet de calculer la list des value pour chaque cle



Exemple d’application 1:

Nous avons un fichier u.data qui contient une list de review pour chaque film

Le premier champ : MovieID
Le dexieme champ : ranting

1005	4
1006	2
1007	3
1008	4
1009	3

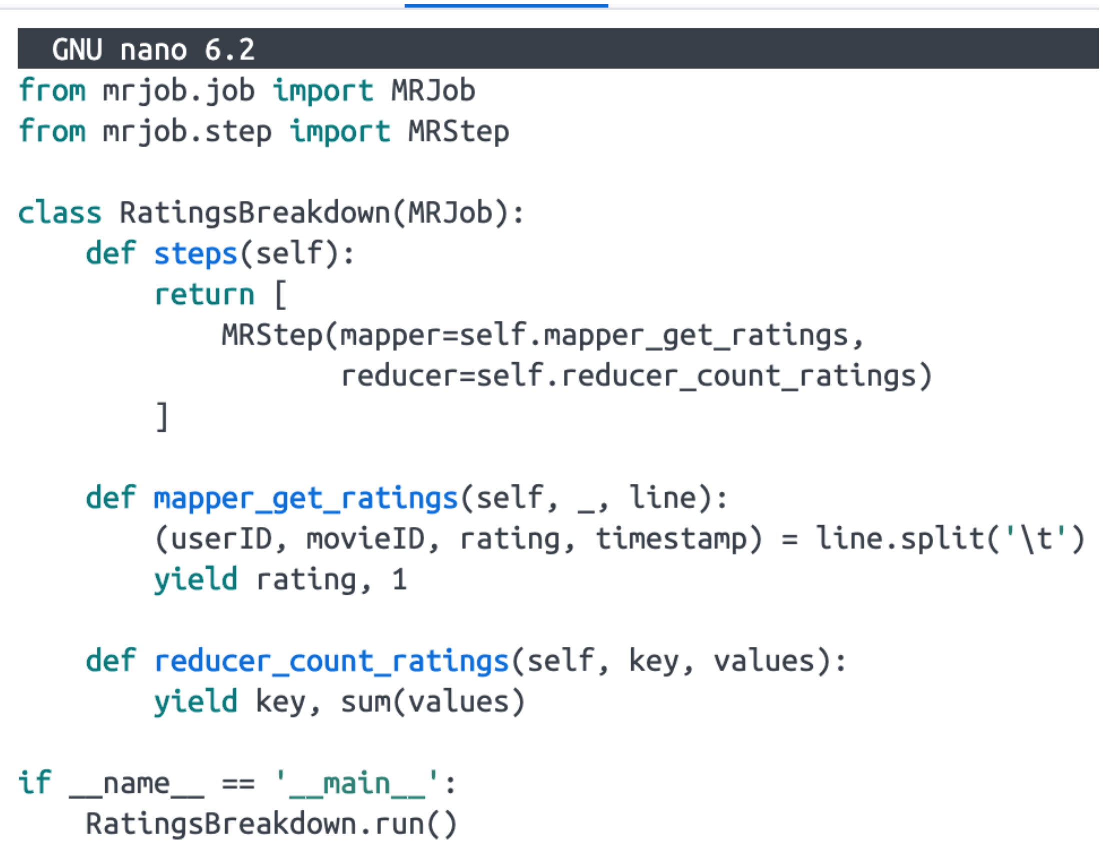
Nous voulons compter combien de fois un ranting se repete

Algorithme

Map --> (4,1) (2,1) (3,1) (4,1) (3,1) ---> shuffle & sort

--> (4,(1,1)) (2,1) (3,(1,1)) ---> Reduce --> (4,2) (2,1) (3,2)

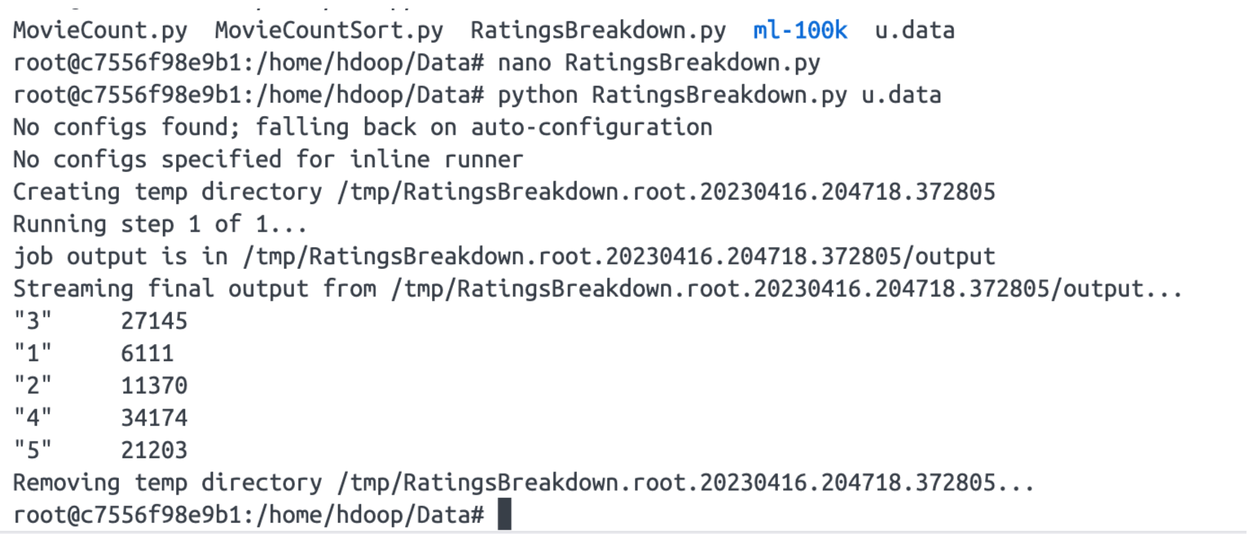
Code python



```
  GNU nano 6.2
from mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                   reducer=self.reducer_count_ratings)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield rating, 1

    def reducer_count_ratings(self, key, values):
        yield key, sum(values)

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

MRStep permet de declarer le mapper et le reducer

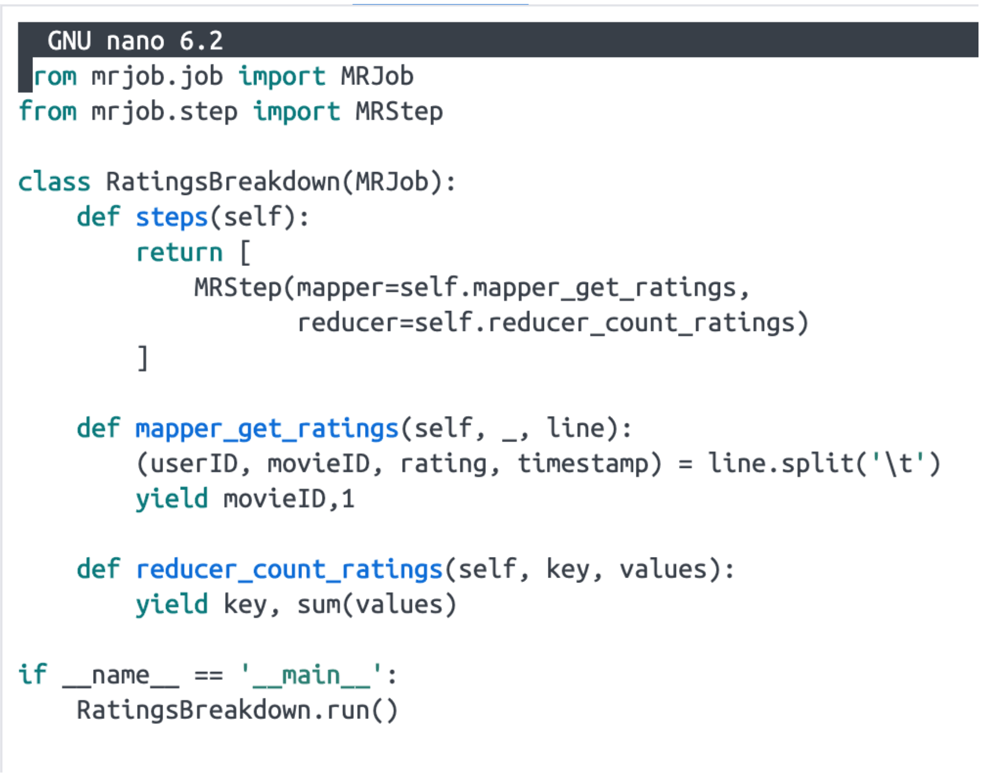
MRJob permet d’ecrir un map reduce Job on python est de l’executer sur different environnemt



```
MovieCount.py  MovieCountSort.py  RatingsBreakdown.py  ml-100k  u.data
root@c7556f98e9b1:/home/hdoop/Data# nano RatingsBreakdown.py
root@c7556f98e9b1:/home/hdoop/Data# python RatingsBreakdown.py u.data
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/RatingsBreakdown.root.20230416.204718.372805
Running step 1 of 1...
job output is in /tmp/RatingsBreakdown.root.20230416.204718.372805/output
Streaming final output from /tmp/RatingsBreakdown.root.20230416.204718.372805/output...
"3"     27145
"1"     6111
"2"     11370
"4"     34174
"5"     21203
Removing temp directory /tmp/RatingsBreakdown.root.20230416.204718.372805...
root@c7556f98e9b1:/home/hdoop/Data# 
```

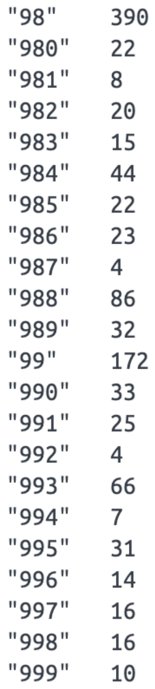
Exemple d’application 2 :

On veut compter le nombre de fois qu’un film a était noter



```
  GNU nano 6.2
rom mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                   reducer=self.reducer_count_ratings)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield movieID,1

    def reducer_count_ratings(self, key, values):
        yield key, sum(values)

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

python MovieCount.py u.data

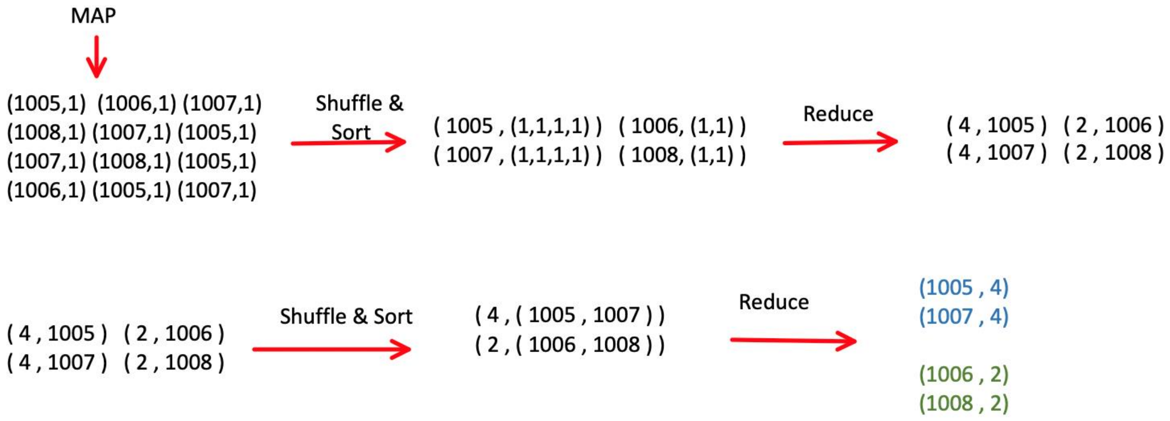


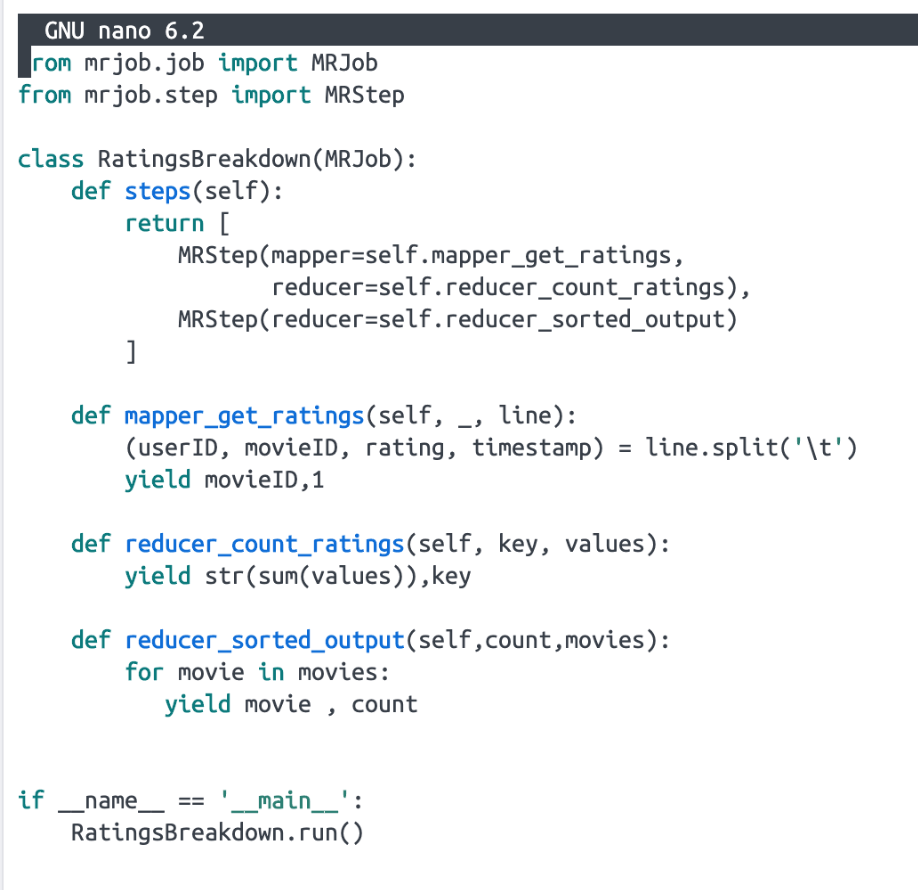
```
"98"     390
"980"    22
"981"    8
"982"    20
"983"    15
"984"    44
"985"    22
"986"    23
"987"    4
"988"    86
"989"    32
"99"     172
"990"    33
"991"    25
"992"    4
"993"    66
"994"    7
"995"    31
"996"    14
"997"    16
"998"    16
"999"    10
```

Exemple d’application 2 :

Pour cette exemple nous voulons regrouper les filme qui ont le meme nombre de review

Algorithme :



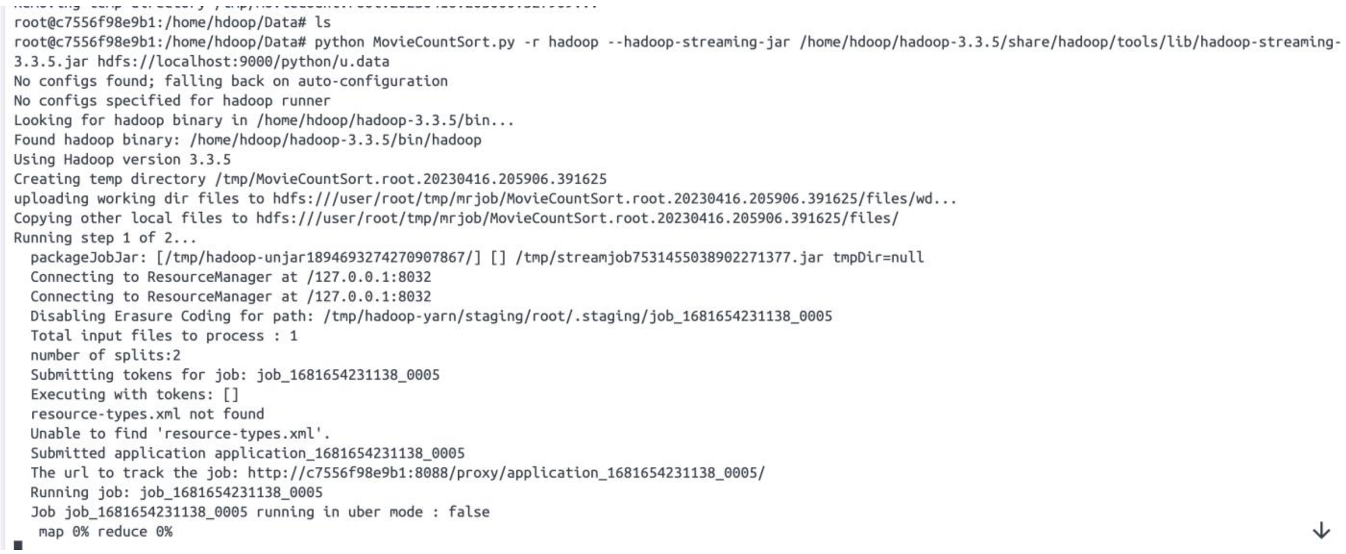


```
rom mrjob.job import MRJob
from mrjob.step import MRStep

class RatingsBreakdown(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ratings,
                   reducer=self.reducer_count_ratings),
            MRStep(reducer=self.reducer_sorted_output)
        ]

    def mapper_get_ratings(self, _, line):
        (userID, movieID, rating, timestamp) = line.split('\t')
        yield movieID,1

    def reducer_count_ratings(self, key, values):
        yield str(sum(values)),key

    def reducer_sorted_output(self,count,movies):
        for movie in movies:
           yield movie , count


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

Exécution :

```
hadoop fs -mkdir /python
hadoop fs -put u.data /python
```

```
# python MovieCountSort.py -r hadoop --hadoop-streaming-jar
/home/hdoop/hadoop-3.3.5/share/hadoop/tools/lib/hadoop-streaming-
3.3.5.jar hdfs://localhost:9000/python/u.data
```



```
root@c7556f98e9b1:/home/hdoop/Data# ls
root@c7556f98e9b1:/home/hdoop/Data# python MovieCountSort.py -r hadoop --hadoop-streaming-jar /home/hdoop/hadoop-3.3.5/share/hadoop/tools/lib/hadoop-streaming-
3.3.5.jar hdfs://localhost:9000/python/u.data
No configs found; falling back on auto-configuration
No configs specified for hadoop runner
Looking for hadoop binary in /home/hdoop/hadoop-3.3.5/bin...
Found hadoop binary: /home/hdoop/hadoop-3.3.5/bin/hadoop
Using Hadoop version 3.3.5
Creating temp directory /tmp/MovieCountSort.root.20230416.205906.391625
uploading working dir files to hdfs:///user/root/tmp/mrjob/MovieCountSort.root.20230416.205906.391625/files/wd...
Copying other local files to hdfs:///user/root/tmp/mrjob/MovieCountSort.root.20230416.205906.391625/files/
Running step 1 of 2...
  packageJobJar: [/tmp/hadoop-unjar1894693274270907867/] [] /tmp/streamjob7531455038902271377.jar tmpDir=null
  Connecting to ResourceManager at /127.0.0.1:8032
  Connecting to ResourceManager at /127.0.0.1:8032
  Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1681654231138_0005
  Total input files to process : 1
  number of splits:2
  Submitting tokens for job: job_1681654231138_0005
  Executing with tokens: []
  resource-types.xml not found
  Unable to find 'resource-types.xml'.
  Submitted application application_1681654231138_0005
  The url to track the job: http://c7556f98e9b1:8088/proxy/application_1681654231138_0005/
  Running job: job_1681654231138_0005
  Job job_1681654231138_0005 running in uber mode : false
   map 0% reduce 0%
```

```
"744"   "92"
"578"   "92"
"549"   "92"
"163"   "92"
"17"    "92"
"212"   "92"
"1011"  "93"
"576"   "93"
"820"   "93"
"430"   "93"
"529"   "93"
"412"   "93"
"90"    "95"
"477"   "95"
"131"   "95"
"919"   "96"
"755"   "96"
"290"   "96"
"306"   "96"
"429"   "97"
"356"   "97"
"33"    "97"
"126"   "97"
"1014"  "98"
"155"   "98"
"507"   "98"
"436"   "99"
```