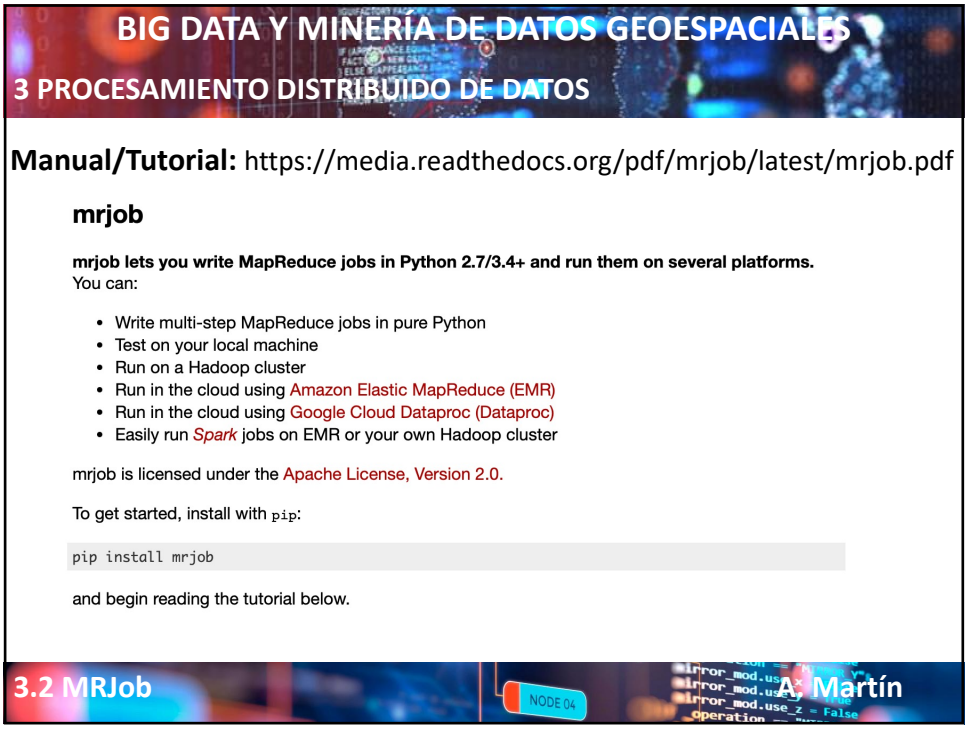

BIG DATA Y MINERÍA DE DATOS GEOESPACIALES
3 PROCESAMIENTO DISTRIBUIDO DE DATOS
3.2 MapReduce en Phython. La clase MRJob

11


BIG DATA Y MINERÍA DE DATOS GEOESPACIALES
3 PROCESAMIENTO DISTRIBUIDO DE DATOS
Manual/Tutorial: https://media.readthedocs.org/pdf/mrjob/latest/mrjob.pdf
mrjob
mrjob lets you write MapReduce jobs in Python 2.7/3.4+ and run them on several platforms.
You can:
Write multi-step MapReduce jobs in pure Python
Test on your local machine
Run on a Hadoop cluster
Run in the cloud using Amazon Elastic MapReduce (EMR)
Run in the cloud using Google Cloud Dataproc (Dataproc)
Easily run Spark jobs on EMR or your own Hadoop cluster
mrjob is licensed under the Apache License, Version 2.0.
To get started, install with pip:
pip install mrjob
and begin reading the tutorial below.
3.2 MRJob
NODE 04
A. Martín

12

## 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.

Here are a number of features of mrjob that make writing MapReduce jobs easier:

- Keep all MapReduce code for one job in a single class
- Easily upload and install code and data dependencies at runtime
- Switch input and output formats with a single line of code
- Automatically download and parse error logs for Python tracebacks
- Put command line filters before or after your Python code

If you don't want to be a Hadoop expert but need the computing power of MapReduce, mrjob might be just the thing for you.







**Clases en Python**: modelos que definen y agrupan las características y propiedades que tendrán los objetos que las instancien. En python, una clase se define con la instrucción *class* seguida de un nombre genérico para el objeto.

File Edit Format Run Options Window Help

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 18 19:08:33 2018

@author: angelmartinfurones
"""
#CLASE QUE SIMULA UNA CALCULADORA
class Calculadora(object):
    #EL CONSTRUCTOR INICIA "VALOR" A 0
    def __init__(self): #SELF HACE REFERENCIA A SI MISMO
        self.valor=0
    #SUMA UN NUMERO 'N' AL VALOR
    def suma(self,n):
        self.valor +=n
    def getValor(self):
        return self.valor

calc = Calculadora() #INSTANCIA UN OBJETO EN LA VARIABLE CALC

calc.suma(2) #SUMA 2 AL VALOR DE LA CALCULADORA

print(calc.getValor()) #MOSTRARIA UN DOS EN LA SHELL

calc.suma(2) #SE SUMA 2 A LA VARIABLE VALOR

print(calc.getValor())

```

```python
from mrjob.job import MRJob

class Purchases(MRJob):

    def mapper(self, key, line):
        if len(line.strip().split(","))==6:
            (date,time,store,item,cost,payment) = line.strip().split(",")
            try:
                yield str(store),float(cost)
            except:
                pass

    def reducer(self, store, cost):
        yield str(store), sum(cost)

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

Desde la terminal:

$ Python3 SolMRJob.py purchases.txt | sort > salidaMRJob.txt



15



```python
from mrjob.job import MRJob

class Purchases(MRJob):

    def mapper(self, key, line):
        if len(line.strip().split(","))==6:
            (date,time,store,item,cost,payment) = line.strip().split(",")
            try:
                yield str(store),float(cost)
            except:
                pass

    def reducer(self, store, cost):
        yield str(store), sum(cost)

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

*Clase MRJob que se compone de una función mapper y otra reducer*

*Inicio, se le dice a python que ejecute el programa*



16


BIG DATA Y MINERÍA DE DATOS GEOESPACIALES
3 PROCESAMIENTO DISTRIBUIDO DE DATOS
from mrjob.job import MRJob
class Purchases(MRJob):
def mapper(self, key, line):
if len(line.strip().split(","))==6:
(date,time,store,item,cost,payment) = line.strip().split(",")
try:
yield str(store),float(cost)
except:
pass
def reducer(self, store, cost):
yield str(store), sum(cost)
if __name__ == '__main__':
Purchases.run()
Ln: 14 Col: 35
• Line hace referencia a cada línea de los datos de entrada sin procesar
• Key hace referencia a la clave de cada una de las líneas de entrada, en este caso no es necesario asignar ninguna clave por lo que se puede sustituir key por "_", con lo que la key es ignorada
3.2 MRJob
NODE 04
A. Martín

17


BIG DATA Y MINERÍA DE DATOS GEOESPACIALES
3 PROCESAMIENTO DISTRIBUIDO DE DATOS
from mrjob.job import MRJob
class Purchases(MRJob):
def mapper(self, key, line):
if len(line.strip().split(","))==6:
(date,time,store,item,cost,payment) = line.strip().split(",")
try:
yield str(store),float(cost)
except:
pass
def reducer(self, store, cost):
yield str(store), sum(cost)
if __name__ == '__main__':
Purchases.run()
Ln: 14 Col: 35
La salida se guardará en la función yield como un par clave,valor
3.2 MRJob
NODE 04
A. Martín

18

```
1  from mrjob.job import MRJob
2
3  class Purchases(MRJob):
4
5      def mapper(self, key, line):
6          if len(line.strip().split(","))==6:
7              (date,time,store,item,cost,payment) = line.strip().split(",")
8              try:
9                  yield str(store),float(cost)
10             except:
11                 pass
12
13     def reducer(self, store, cost):
14         yield str(store), sum(cost)
15
16 if __name__ == '__main__':
17     Purchases.run()
18
```

Ln: 14 Col: 35

*Sort and Group, lo hace MRJob por nosotros, no hay que programar nada*



19



```
1  from mrjob.job import MRJob
2
3  class Purchases(MRJob):
4
5      def mapper(self, key, line):
6          if len(line.strip().split(","))==6:
7              (date,time,store,item,cost,payment) = line.strip().split(",")
8              try:
9                  yield str(store),float(cost)
10             except:
11                 pass
12
13     def reducer(self, store, cost):
14         yield str(store), sum(cost)
15
16 if __name__ == '__main__':
17     Purchases.run()
18
```

Ln: 14 Col: 35

*La entrada son, en este caso, las tiendas y el coste, pero ya ordenados (al coste le podemos asignar otro nombre si queremos, por ejemplo occurences*



20

# BIG DATA Y MINERÍA DE DATOS GEOESPACIALES

## 3 PROCESAMIENTO DISTRIBUIDO DE DATOS

```python
from mrjob.job import MRJob

class Purchases(MRJob):

    def mapper(self, key, line):
        if len(line.strip().split(","))==6:
            (date,time,store,item,cost,payment) = line.strip().split(",")
            try:
                yield str(store),float(cost)
            except:
                pass

    def reducer(self, store, cost):
        yield str(store), sum(cost)

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

Ln: 14 Col: 35

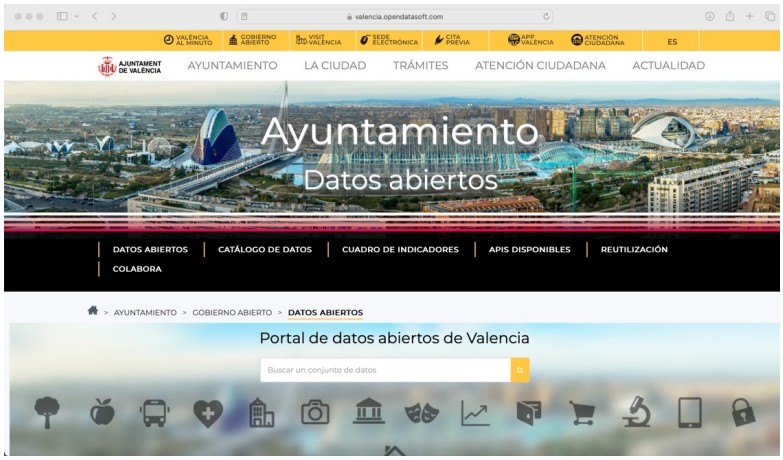
*La salida será un par clave/valor con la tienda y la suma de los costes*



21

# BIG DATA Y MINERÍA DE DATOS GEOESPACIALES

## 3 PROCESAMIENTO DISTRIBUIDO DE DATOS






22

trafico2013-2014.csv

```
Punto,Mes,Dia,Anyo,Hora,Intensidad
106,4,1,2013,0,142
106,4,1,2013,1,82
106,4,1,2013,2,54
106,4,1,2013,3,68
106,4,1,2013,4,33
106,4,1,2013,5,41
106,4,1,2013,6,61
106,4,1,2013,7,71
106,4,1,2013,8,100
106,4,1,2013,9,134
106,4,1,2013,10,183
106,4,1,2013,11,227
106,4,1,2013,12,334
106,4,1,2013,13,342
106,4,1,2013,14,300
106,4,1,2013,15,214
106,4,1,2013,16,279
106,4,1,2013,17,345
106,4,1,2013,18,329
106,4,1,2013,19,429
106,4,1,2013,20,447
106,4,1,2013,21,387
106,4,1,2013,22,227
106,4,1,2013,23,171
106,4,2,2013,0,100
106,4,2,2013,1,59
106,4,2,2013,2,40
106,4,2,2013,3,26
106,4,2,2013,4,16
```



23



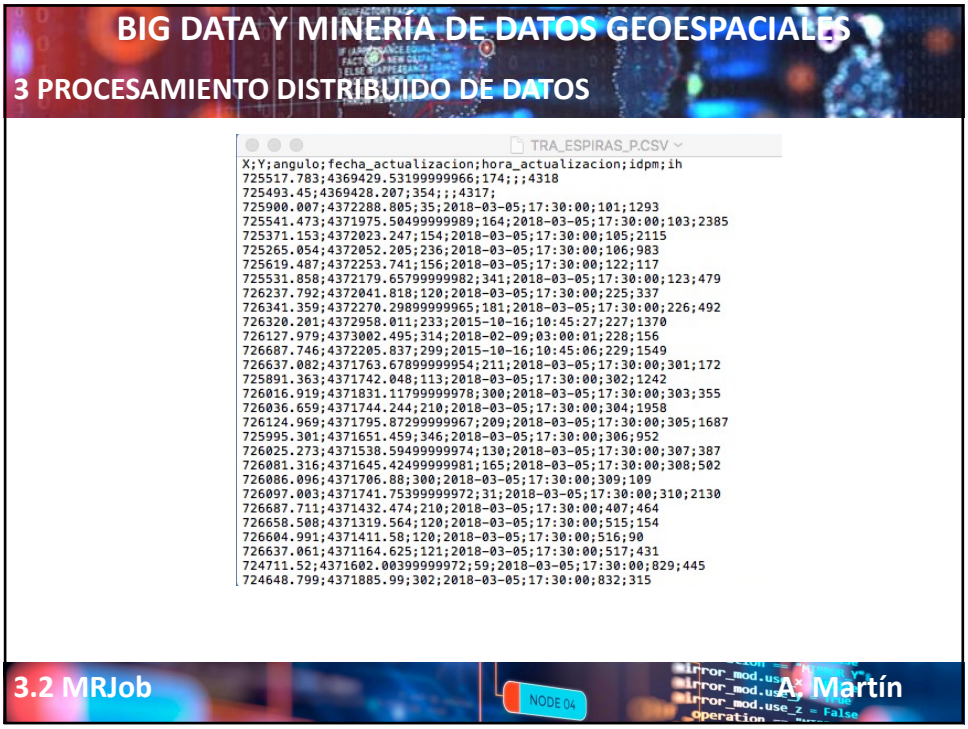
File Edit Format Run Options Window Help

```
1  from mrjob.job import MRJob
2
3  class IntensidadPuntos(MRJob):
4
5      def mapper(self, key, line):
6          (punto,mes,dia,anyo,hora,intensidad) = line.split(',')
7          if punto !='Punto':
8              yield str(punto), int(intensidad)
9
10     def reducer(self, punto, intensidad):
11         total = 0
12         numElements = 0
13         for x in intensidad:
14             total += x
15             numElements += 1
16
17         yield punto, int(total / (numElements))
18
19 if __name__ == '__main__':
20     IntensidadPuntos.run()
21
```

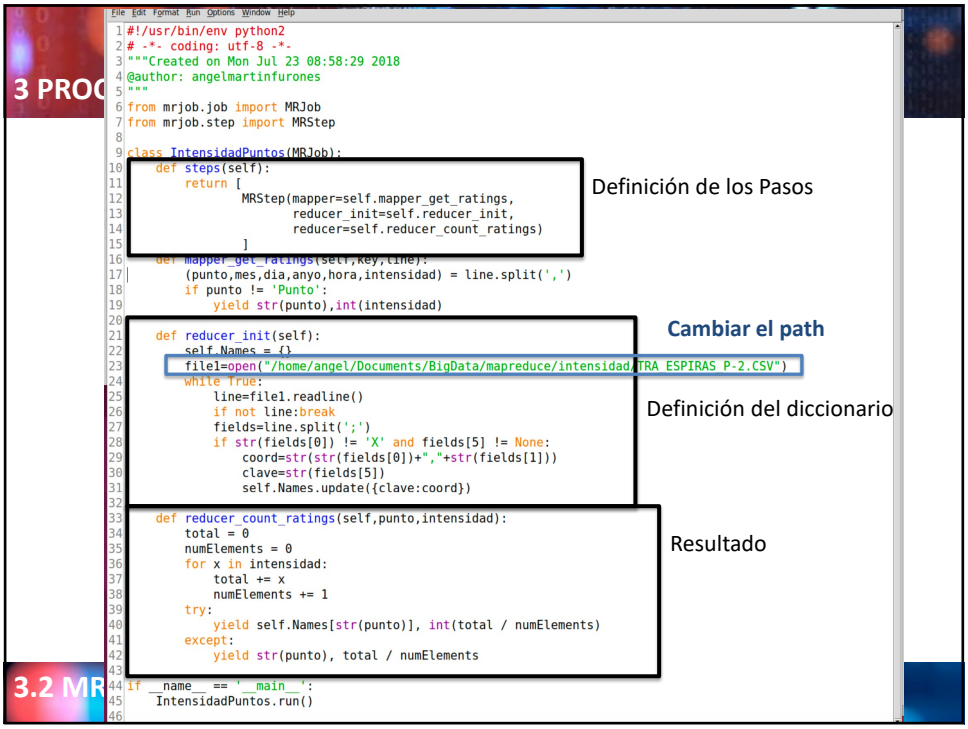
Ln: 20 Col:



24


BIG DATA Y MINERÍA DE DATOS GEOESPACIALES
3 PROCESAMIENTO DISTRIBUIDO DE DATOS
TRA_ESPIRAS_P.CSV
X;Y;angulo;fecha_actualizacion;hora_actualizacion;idpm;ih
725517.783;4369429.53199999966;174;;;4318
725493.45;4369428.207;354;;;4317;
725900.007;4372288.805;35;2018-03-05;17:30:00;101;1293
725541.473;4371975.50499999989;164;2018-03-05;17:30:00;103;2385
725371.153;4372023.247;154;2018-03-05;17:30:00;105;2115
725265.054;4372052.205;236;2018-03-05;17:30:00;106;983
725619.487;4372253.741;156;2018-03-05;17:30:00;122;117
725531.858;4372179.65799999982;341;2018-03-05;17:30:00;123;479
726237.792;4372041.818;120;2018-03-05;17:30:00;225;337
726341.359;4372270.29899999965;181;2018-03-05;17:30:00;226;492
726320.201;4372958.011;233;2015-10-16;10:45:27;227;1370
726127.979;4373002.495;314;2018-02-09;03:00:01;228;156
726687.746;4372205.837;299;2015-10-16;10:45:06;229;1549
726637.082;4371763.67899999954;211;2018-03-05;17:30:00;301;172
725891.363;4371742.048;113;2018-03-05;17:30:00;302;1242
726016.919;4371831.11799999978;300;2018-03-05;17:30:00;303;355
726036.659;4371744.244;210;2018-03-05;17:30:00;304;1958
726124.969;4371795.87299999967;209;2018-03-05;17:30:00;305;1687
725995.301;4371651.459;346;2018-03-05;17:30:00;306;952
726025.273;4371538.59499999974;130;2018-03-05;17:30:00;307;387
726081.316;4371645.42499999981;165;2018-03-05;17:30:00;308;502
726086.096;4371706.88;300;2018-03-05;17:30:00;309;109
726097.003;4371741.75399999972;31;2018-03-05;17:30:00;310;2130
726687.711;4371432.474;210;2018-03-05;17:30:00;407;464
726658.508;4371319.564;120;2018-03-05;17:30:00;515;154
726604.991;4371411.58;120;2018-03-05;17:30:00;516;90
726637.061;4371164.625;121;2018-03-05;17:30:00;517;431
724711.52;4371602.00399999972;59;2018-03-05;17:30:00;829;445
724648.799;4371885.99;302;2018-03-05;17:30:00;832;315
3.2 MRJob
NODE 04
A. Martín

25


3 PROC
File Edit Format Run Options Window Help
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""Created on Mon Jul 23 08:58:29 2018
@author: angelmartinfurones
"""
from mrjob.job import MRJob
from mrjob.step import MRStep
class IntensidadPuntos(MRJob):
def steps(self):
return [
MRStep(mapper=self.mapper_get_ratings,
reducer_init=self.reducer_init,
reducer=self.reducer_count_ratings)
]
def mapper_get_ratings(self,key,line):
(punto,mes,dia,anyo,hora,intensidad) = line.split(',')
if punto != 'Punto':
yield str(punto),int(intensidad)
def reducer_init(self):
self.Names = {}
file1=open("/home/angel/Documents/BigData/mapreduce/intensidad/TRA_ESPIRAS_P-2.CSV")
while True:
line=file1.readline()
if not line:break
fields=line.split(';')
if str(fields[0]) != 'X' and fields[5] != None:
coord=str(str(fields[0])+","+str(fields[1]))
clave=str(fields[5])
self.Names.update({clave:coord})
def reducer_count_ratings(self,punto,intensidad):
total = 0
numElements = 0
for x in intensidad:
total += x
numElements += 1
try:
yield self.Names[str(punto)], int(total / numElements)
except:
yield str(punto), total / numElements
if __name__ == '__main__':
IntensidadPuntos.run()
Definición de los Pasos
Cambiar el path
Definición del diccionario
Resultado
3.2 MR

26


27

## BIG DATA Y MINERÍA DE DATOS GEOESPACIALES

### 3 PROCESAMIENTO DISTRIBUIDO DE DATOS

MapReduce es tan solo un modelo de programación. Es necesario un *framework* que proporcione la infraestructura de soporte para la ejecución de programas MapReduce. Estos son **Apache Hadoop** y **Apache Spark** (aunque no son los únicos son los más usados y el estándar de uso *de facto*)



28