BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

3.1 MapReduce programming model

1

BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

- Popularized by Google from the publication :
  J.Dean and, S.Ghemawat,"MapReduce:SimplifiedDataProcessingon Large Clusters", Communications of the ACM, Jan 2008, Vl 51 No. 1.

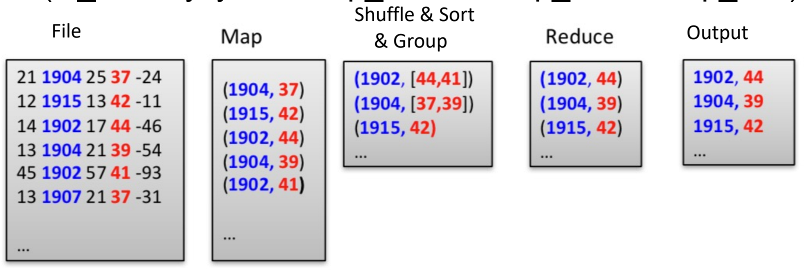
3.1 MapReduce A. Martín

2

0. (**Splitting**). Data splitting (if necessary and possible)
1. (**Map**). Design a set of simple ”Map" tasks that generate a set of intermediate results on an input data partition that are pairs of the type: (key, value)
2. (**Suffle and Sort).** The intermediate results (key, value) are grouped and sorted (by key).
3. (**Reduce**). The pairs ordered by key are processed by another simple set of tasks ”Reduce" to produce the result.

3

Text file where each line contains metrorological measurements of a station in a country
(id_country, year, temp_mean, temp_max, temp_min)

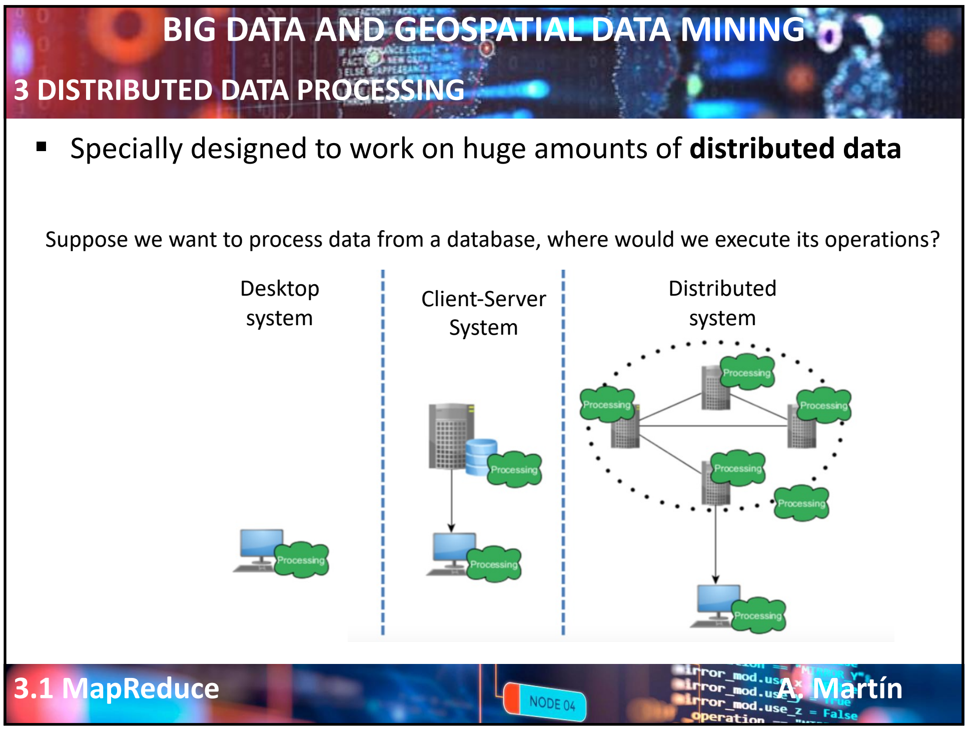


File	Map	Shuffle & Sort & Group	Reduce	Output
21 1904 25 37 -24	(1904, 37)	(1902, [44,41])	(1902, 44)	1902, 44
12 1915 13 42 -11	(1915, 42)	(1904, [37,39])	(1904, 39)	1904, 39
14 1902 17 44 -46	(1902, 44)	(1915, 42)	(1915, 42)	1915, 42
13 1904 21 39 -54	(1904, 39)	…	…	…
45 1902 57 41 -93	(1902, 41)			
13 1907 21 37 -31	…			
…				

cat data.txt | mapper.py | sort | reducer.py > salida.txt

(1902, 44)
(1902, 41)
(1904, 37)
(1904, 39)
(1915, 42)

4

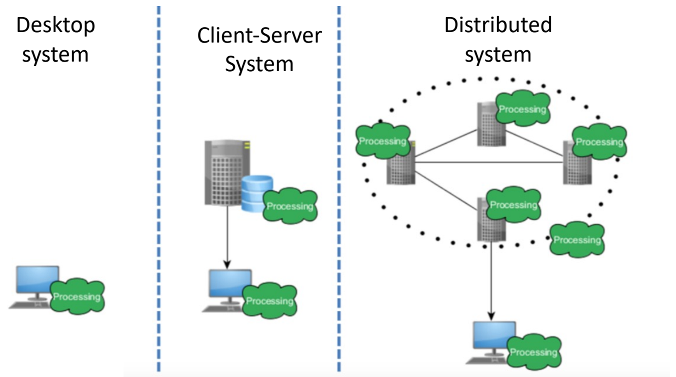


BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

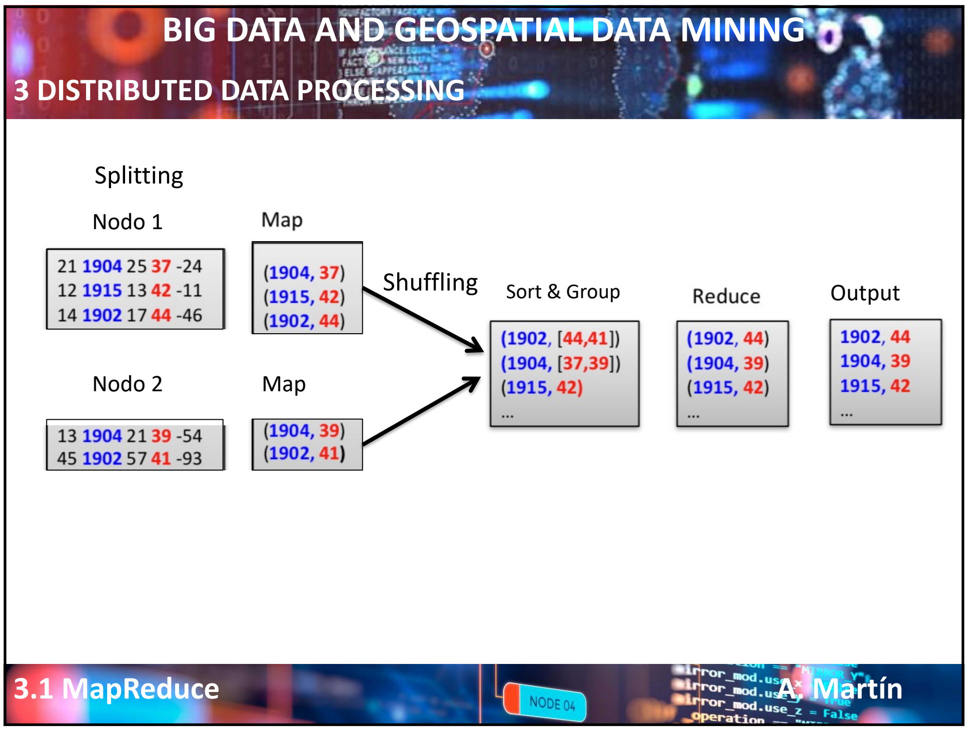
- Specially designed to work on huge amounts of **distributed data**

Suppose we want to process data from a database, where would we execute its operations?



3.1 MapReduce — A. Martín

5



BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

Splitting

Nodo 1

21 1904 25 37 -24
12 1915 13 42 -11
14 1902 17 44 -46

Map

(1904, 37)
(1915, 42)
(1902, 44)

Nodo 2

13 1904 21 39 -54
45 1902 57 41 -93

Map

(1904, 39)
(1902, 41)

Shuffling

Sort & Group

(1902, [44,41])
(1904, [37,39])
(1915, 42)
…

Reduce

(1902, 44)
(1904, 39)
(1915, 42)
…

Output

1902, 44
1904, 39
1915, 42
…

3.1 MapReduce — A. Martín

6

BIG DATA AND GEOSPATIAL DATA MINING

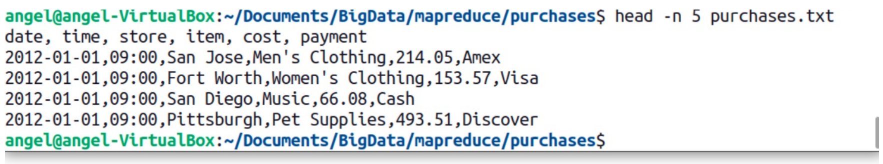
3 DISTRIBUTED DATA PROCESSING

- MapReduce is a distributed programming environment on distributed data
- Reduces the natural complexity of a distributed system
- Take advantage of the local storage of the data
- It allows to process large amount of data efficiently

BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

EXERCISE: Shopping data set, purchases.txt file



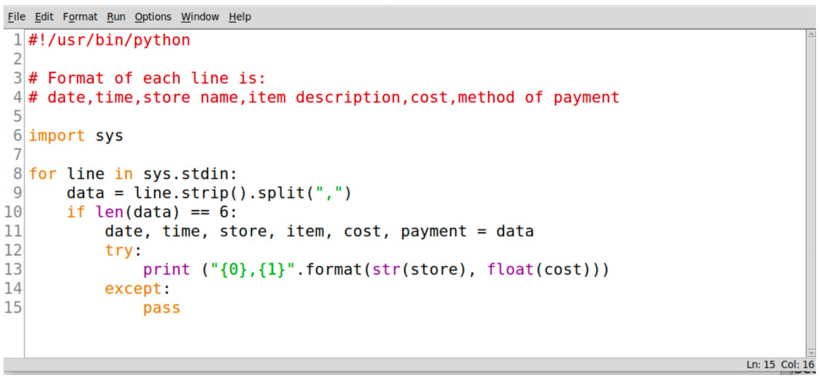
```
angel@angel-VirtualBox:~/Documents/BigData/mapreduce/purchases$ head -n 5 purchases.txt
date, time, store, item, cost, payment
2012-01-01,09:00,San Jose,Men's Clothing,214.05,Amex
2012-01-01,09:00,Fort Worth,Women's Clothing,153.57,Visa
2012-01-01,09:00,San Diego,Music,66.08,Cash
2012-01-01,09:00,Pittsburgh,Pet Supplies,493.51,Discover
angel@angel-VirtualBox:~/Documents/BigData/mapreduce/purchases$
```

Calculate total values by store (by city)

BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

MAPPER.PY



```python
#!/usr/bin/python

# Format of each line is:
# date,time,store name,item description,cost,method of payment

import sys

for line in sys.stdin:
    data = line.strip().split(",")
    if len(data) == 6:
        date, time, store, item, cost, payment = data
        try:
            print ("{0},{1}".format(str(store), float(cost)))
        except:
            pass
```

3.1 MapReduce

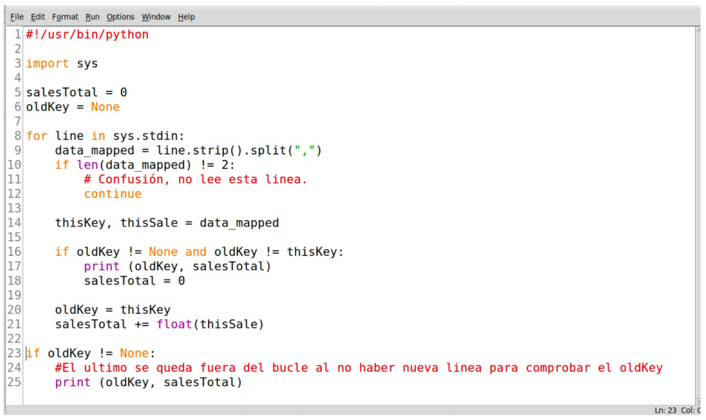
A. Martín

9

BIG DATA AND GEOSPATIAL DATA MINING

3 DISTRIBUTED DATA PROCESSING

REDUCER.PY



```python
#!/usr/bin/python

import sys

salesTotal = 0
oldKey = None

for line in sys.stdin:
    data_mapped = line.strip().split(",")
    if len(data_mapped) != 2:
        # Confusión, no lee esta linea.
        continue

    thisKey, thisSale = data_mapped

    if oldKey != None and oldKey != thisKey:
        print (oldKey, salesTotal)
        salesTotal = 0

    oldKey = thisKey
    salesTotal += float(thisSale)

if oldKey != None:
    #El ultimo se queda fuera del bucle al no haber nueva linea para comprobar el oldKey
    print (oldKey, salesTotal)
```

```
cat purchases.txt | python3 mapper.py | sort | python3 reducer.py > salida.txt
```

3.1 MapReduce

A. Martín

10