Big Data with HADOOP

Adrien GRINCOURT   -

TP1

The goal of this exercise is to create a pseudo-code for the `Map()` and `Reduce()` function in a theoretical HADOOP program which purpose is to pivot a .csv file.

These files are composed of different elements separated by two characters: commas and carriage returns. Since .csv files are meant to represent arrays, commas separate the columns and carriage returns the rows.

Hence, this data:

A		B	C	
	D	E		F

Will be represented by:

A,,B,C

,D,E,,F

The input reader shall separate every line of the file and send it to the `Map()` function so it receives 3 parameters: `key` (row number), `value` (the whole line) and `list` where all the data is stored.

Note that every `key` value is going to be unique so it is not grouped afterwards.


```
##We will use Pseudo Code for this TP

##Input reader:
##Described above

##Map
Map (key,value,list) { #Called for each line of the .csv file
     String cell;
     Int column = 0;
     foreach (a : key) {
          column++;

          ##Storing the value of the cell in the String
          cell = value.read(',')

          ##If it is an empty cell, we don't treat the data
          if(cell != "")
          {
               list.put(cell,[key,column])
               ## (a, [1,1]), (b, [1,2]) and so on…
          }
     }
     Return list;
}
```

```
##Shuffle and sort

##In the example we saw in class, the shuffle and sort had to "give" parts
of the file to workers such that Worker 1 would get words beginning with
letters from A to L, another would get M to S etc

##Now, we have N lines in the file. So, we must set an arbitrary number of
lines to the workers because we don't know N. I don't have to program it
for this project though.
```

The role of the `Reduce()` function is simple : invert the 2 values stored in the key (rows & columns) such that the file is going to be pivoted.

```
##Reduce (key,list_val,list){
      foreach(v : key){
            list_val.put(list.value, [ list.key[1], list.key[0] ])
##Storing in list_val the inverted position
      }
      Return list_val;
}
```