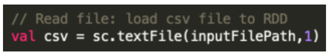
Introduction

The intention of this assignment is to use RDD storage with Scala. The main task is to process a csv file records HTTP GET and POST requests, and store payloads of each requests in to RDD. Finally, output the minimum, maximum, mean and variance of payloads of each client.

Input



```
// Read file: load csv file to RDD
val csv = sc.textFile(inputFilePath,1)
```

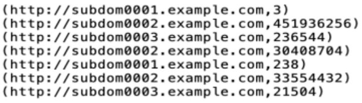
The input of this project is a csv file and is further loaded to RDD for processing. The method takes the local directory of the file and stores it as RDD. And in the Spark Shell, a special interpreter-aware SparkContext is predefined as *sc*.

Pre-processing

Remove the comma of each HTTP request and the redundant columns. Moreover, map each HTTP request as (key, value) pair, where the name of a client is the key and the payload of such client is the value.

The payloads need to be further processed with mathematical operation, so they should be stored as integer type (or Long). And all of the value should be unified as Byte.

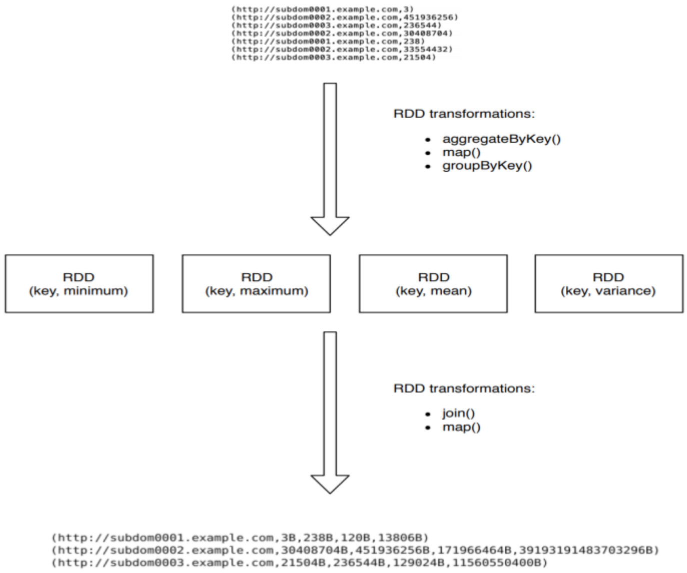
After the processing the data is stored in RDD as follow:



```
(http://subdom0001.example.com,3)
(http://subdom0002.example.com,451936256)
(http://subdom0003.example.com,236544)
(http://subdom0002.example.com,30408704)
(http://subdom0001.example.com,238)
(http://subdom0002.example.com,33554432)
(http://subdom0003.example.com,21504)
```

Processing

The main idea of the assignment is to use RDD Transformation to the maximum, minimum, mean and variance respectively. And then merge them together to get the output template.



```
(http://subdom0001.example.com,3)
(http://subdom0002.example.com,451936256)
(http://subdom0003.example.com,236544)
(http://subdom0002.example.com,30408704)
(http://subdom0001.example.com,238)
(http://subdom0002.example.com,33554432)
(http://subdom0003.example.com,21504)
```

RDD transformations:

- aggregateByKey()
- map()
- groupByKey()

RDD
(key, minimum) | RDD (key, maximum) | RDD (key, mean) | RDD (key, variance)

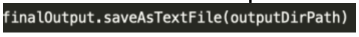
RDD transformations:

- join()
- map()

```
(http://subdom0001.example.com,3B,238B,120B,13806B)
(http://subdom0002.example.com,30408704B,451936256B,171966464B,39193191483703296B)
(http://subdom0003.example.com,21504B,236544B,129024B,11560550400B)
```

Output

Save RDD as csv in predefined directory.



```
finalOutput.saveAsTextFile(outputDirPath)
```