Big Data: Large Volume Data Analysis

Second Group Work
Group 7

(master) Thomas Psaltikidis, 15173
Athanasios Farmakis, dai17102
George Zdoupas, 15336

Stylianos Gogas, dai17116

PROBLEMS THAT WERE TACKLED

- **Cluster:**

  We needed to re-create and rearrange the cluster, change the master and create the workers. In the process as we ran our code in python via cluster the workers could not find the path for the file so we used hdfs and transferred our code there so that it is in a common for the whole cluster space.

- **Code syntax, imports:**

  We changed our code (python) many times in an attempt to solve the task. We all had problems with imports in the spark so we needed to install them there as well. We encountered problems with the lamda (...) function process in python and how it will read and process data from the file. Also when we imported HIGGS.csv to read it in the code and there we had problems in how to find the right path.

- **RDD-Hdfs issues:**

  Here our problems were that we had to empty space so that HIGGS.csv could accept and read it. Also, even though HIGGS.csv was in hdfs, the code still could not find and read them until we could figure out how to spell the path correctly to find it.

- **Caching-persistance-parsing:**

  Our biggest problem was this area of machine memory. It displayed warnings "not enough space to cache rdd_someNumber in memory" so it does not run. We used the .persist command (StorageLevel.DISK_ONLY) to cache the disk and not the memory.

- **Troubleshooting:**

  Most problems were solved with the help of stackoverflow and the official Apache documentation for Spark. In the last problem we encountered we sent you an email where you directed us on what we need to do to solve it.

CODE CONFIGURATION AND AVERAGE TIMES

- **Parameters:**

In the code we implemented we have 4 parameters that can affect its output:

```
popSize = 35
crossover_prob = 1
mutation_prob = 0.4
nbGen = 5
```

doing test runs we found that for the above values comes out the highest possible accuracy (58%) for the model we implemented.

- **Average times:**

  There is a file (project2-outputs.txt) that contains in detail (learning time, accuracy etc) all the outputs from the 6 executions we did. The average execution times are:

  - Single node runs average time: 4095 seconds ≈ 68 mins
  - Cluster runs average time: 1106 seconds ≈ 18 mins 30 seconds