ε.δε.μ^2
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ACADEMIC YEAR 2022-2023
SPRING SEMESTER
LESSON: Large Scale Data Management
Final Project

Deadline: 25/06/2023

For the Final project you're going to use Apache spark to execute queries on datasets. There are 2 basic APIs for query execution.

1. RDD API: https://spark.apache.org/docs/latest/rdd-programming-guide.html
2. DataFrame API / SQL API: https://spark.apache.org/docs/latest/sql-programming-guide.html

You can read more about SQL query execution in spark here: https://spark.apache.org/docs/latest/sql-programming-guide.html#running-sql-queries-programmatically

To download the Dataset for the final project you need to execute the following commands in your okeanos master machine:

```
wget http://www.cslab.ece.ntua.gr/~ikons/bigdata/datasets2023.tar.gz

tar -xzf datasets2023.tar.gz
```

Explanation:

This dataset that we will use is from Common Crawl and contains raw web page data and metadata, all stored in Amazon S3. This dataset has been used to train the ChatGPT bot. The data can be downloaded using HTTPS or S3.

The data is available in the WARC(Web ARChive) file format and also contains metadata (WAT) and text data (WET) extracts. We will use part of the dataset through the aforementioned commands since we can't fit the whole data collected from 2008 in our okeanos VMs.

The file contains 5 files: warc.csv, wat.csv, wet.csv, departmentsR.csv, employeesR.csv.

The files are in the csv format, executing queries in this format is not efficient. To optimize data access, traditionally databases load data into special designed binary formats. Spark has a similar approach too; we can convert datasets to a special format named "Parquet"[1].

The parquet file formats have major benefits:

1. has a smaller footprint in memory and disk and therefore optimizes I / O (input / output) reducing execution time

[1] https://spark.apache.org/docs/latest/sql-data-sources-parquet.html

2. Maintains additional information, such as statistics on the dataset, which helps on more efficient processing

The parquet file format is a columnar file format, you can read more about it here https://parquet.apache.org/ . On how to read and write parquet files you can find information here: https://spark.apache.org/docs/3.3.1/sql-data-sources-parquet.html

Explaining the dataset

This is a brief explanation of what the datasets looks like:

For the file warc.csv:

This file contains the warc related data. Example line is:

```
2017-03-22T22:13:57Z , <urn:uuid:3096cbb6-4a95-4bf9-bda4-d69ef8315d66>
, response , 40802 , 213.155.18.48 ,
http://03online.com/news/zheltaya_kozha_litsa_i_shelusheniya/2014-3-
30-17340 , Apache/2.2.3 , '<html>\n<head>\n \n<meta charset=""utf-8""
/><script type=""text/javascript"">(window………</html>'
```

First is the warc date, then the warc record id, the warc type (e.g. metadata, response, etc), the content length, the public address IP, the target URL, the server that the site is running(e.g. apache, nginx, etc), and finally the whole context of the page with the whole HTML DOM.

For the file wat.csv:

This file contains wat related data. Example line is:

```
<urn:uuid:66713fe5-07cf-43ca-a23f-49850026c2ef> , 1053 ,
http://00pon00.blog130.fc2.com/blog-category-8.html
```

First is the warc record id, then the content length of the metadata, and finally the target URL (may be different from the target URL of warc data).

For the file wet.csv:

This file contains wet related data. Example line is:

```
<urn:uuid:a2327dea-2a4a-4a59-9442-da698e3706ce> ,
”b'1755\n\nAGROVILLAFRANCA, SL - Informe mercantil de la
empresa\nEmpresas.info Informaci\xc3\xb3n de
Empresas\nInicio\nEmpresas\nAGROVILLAFRANCA, SL\nDebes estar
registrado para ver toda la informaci\xc3\xb3n de esta
empresa.\nIniciar sesi\xc3\xb3n Registro\nAGROVILLAFRANCA,
SL\nEstado\nActiva\nCargos Activos\n2\nAnuncios
BORME\n4\n\xc3\x9altima actualizaci\xc3\xb3n\n11/10/2016\nDatos
identificativos y actividad\nPerfil de empresa\nDenominaci\xc3\xb3n
Social:\nAGROVILLAFRANCA, …..”
```

First is the warc record id and then is the extracted plaintext of the url (it may be in ascii).

Part 1

Task 1:

Create a files directory on HDFS and then upload the CSVs to the files directory: provide the commands needed for the directory creation and the files upload. Provide a print screen that shows the csv files in the directory you created (**0.5 points**). Convert the files to parquet and upload them on HDFS (**0.5 points**)

Task 2:

Using RDDs write code to answer the following queries (Q1-Q5) you can use the csv or parquet files you uploaded on HDFS (**2.75 points, 0.55x5**)

Task 3:

Using DataFrames write code to answer the following queries (Q1-Q5) using the parquet files you created and uploaded to HDFS (**2.75 points, 0.55x5**)

Task 4:

For every query (Q1-Q5) measure the execution time of each scenario:

1. Map/Reduce – RDD API
2. Spark SQL on csv files (you need to define the schema of each file)
3. Spark SQL on parquet files

Create a bar chart with the execution times grouped by query number and write a paragraph explaining the results (**0.5 points**).

Q1: For the time range between 2017-03-22 22:00 and 2017-03-22 23:00, find the most used servers in descending order.

INFO: For this query you will need to filter out the entries that have null values and process the datetime with a Python library.

Q2: For the warc target URL "http://1001.ru/articles/post/ai-da-tumin-443", find the content length of the metadata as well as the size of HTML dom (number of characters).

INFO: For this query you will need to filter based on the url and keep as warc type ''request".

Q3: What are the 5 warc record ids/target urls with the biggest content length that use as a server Apache in ascending order?

INFO: For this query you will need to filter out the entries that have null values.

Q4: For every server used, find the average content length of the warc records as well as the average content legth of their metadata (print top 5 servers with the most content length size).

INFO: For this query you will need to filter out the entries that have null values.

Q5: Find the most popular target URL (i.e., the record target URL that can be found in another record's HTML DOM).

INFO: For this query you will need to filter out the entries that have null values. You should first find for each warc record what is its target URL and what URLs exist in the HTML DOM, so you would have an intermediate result of: targetURL -> list(urls_in_html_dom). For the URLs you could strip them and keep a more simple format/subdomain, e.g. http(s)://X.X to have even more results.

Part 2

In this part of the final project, you will evaluate different join algorithms. More specifically you will implement the broadcast join algorithm (aka Map side join) and repartition join algorithm (aka Reduce side join). You can read more about the algorithms and their implementations here (for broadcast join Paragraph 3.2 , Pseudocode A.4. and for repartition join Paragraph 3.1, Pseudocode A.1)

Datasets explanation:

For the datasets employessR.csv and departmentsR.csv:

First line of employeesR:

1	Elizabeth Jordan	7

Employees id, name and department id

First line of departmentsR.csv:

1	Dep A

Department ID, Department Name

INFO: Join the two datasets on the department id key

Task 1:

Write code with Map/Reduce jobs that implements the broadcast join (RDD API) (**1 point**)

Task 2:

Write code with Map/Reduce jobs that implements the repartition join (RDD API) (**1 point**)

Task 3:

Spark SQL includes different join algorithms implementations. It also includes a query optimizer named "Catalyst". With the help of catalyst Spark SQL chooses the best join algorithm based on the tables we want to join. We can stop Spark from choosing a join algorithm automatically. You can find more information here and here.

You can use the following script to stop spark from choosing a join algorithm automatically. Execute the following query with the query optimizer enabled and with the query optimizer disabled. Create a bar graph with the execution time and explain the results. Take a screenshot of the SQL plan printed by the command spark.sql(query).explain() and give a brief explanation of the SQL plan (**1 point**).

INFO: you need to change every variable is defined with <> and provide your own custom values

To run the query with Catalyst on: spark-submit <Name_of_the_file>.py Y

To run the query with Catalyst off: spark-submit <Name_of_the_file>.py N

```
from pyspark.sql import SparkSession
import sys, time
disabled = sys.argv[1]
spark = SparkSession.builder.appName('query1-sql').getOrCreate()
if disabled == 'Y':
   spark.conf.set(<conf>,<value>)
elif disabled == 'N':
   pass
else:
   raise Exception #("This setting is not available.")

df = spark.read.format("parquet")
df1 = df.load(<load departments from hdfs>)
df2 = df.load(<load employees from hdfs>)
df1.registerTempTable("departments")
df2.registerTempTable("employees")

sqlString = \
"SELECT * " + \
"FROM " + \
" (SELECT * FROM employees LIMIT 50) as e JOIN departments as d ON e._c2 =
d._c0" + \
" JOIN departments d2 ON d2._c0 = d._c0 ;"
t1 = time.time()
spark.sql(sqlString).show()
t2 = time.time()
spark.sql(sqlString).explain()
print("Time with choosing join type %s is %.4f sec."%("enabled" if
disabled == 'N' else "disabled", t2-t1))
```

1. Team size: 2 members
2. The deadline of the final project is: 25/06/2023
3. This assignment constitutes 40% of the total grade of the course. Each team must successfully pass a mandatory oral examination of this assignment. Date of the oral examination will be announced.
4. You can ask questions about the assignment on the forum so question/answer can be available for everyone
5. Submit a zip file named the combination of the two members AM numbers separated by a "_" e.g., 03100000_03100001.zip. The zip file should contain a file with the code used named "code"

(you can create separate code files for every task-part of the assignment e.g. Q1RDD.py, Q1DF.py …) a file named "output" that will contain the outputs of every execution of all the parts of the assignment and a PDF named "Report" that will contain every screenshot and explanation of every part of the assignment .