FINAL PRACTICAL WORK

IMPLEMENT BRIN AND PAGE'S PAGERANK ALGORITHM

BASE ALGORITHM TO WEIGHT WEBPAGES

2022/23

The PageRank algorithm (http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf), proposed by Sergey Brin and Larry Page in 1998, was the foundation to implement the original Google Search Engine for the internet.

The goal of this algorithm was to set a weight for each webpage on the internet, based on the number of external links pointing to the page.

You can find the full paper in the previous link.

GOALS OF THE PRACTICAL WORK

Implement the original algorithm to set the weights for each page in Wikipedia (English version), using the image stored in the Databricks Databases Set.
The URI for the database is:

dbfs:/databricks-datasets/wikipedia-datasets/data-001/en_wikipedia/articles-only-parquet

ANALYSIS OF THIS DATABASE

The database contains 5823210 entries, stored in a parquet set of files.
You can create a Spark DataFrame from those parquet files with the command:

```
wikipediaDF=spark.read.parquet("dbfs:/databricks-datasets/wikipedia-datasets/data-001/en_wikipedia/articles-only-parquet")
```

Instead to use the full database, it is recommended to use a smaller version to analyse the structure, with 0.01% of records (approx. 582 records).

The DataFrame contains 7 columns named:

- title
- id
- revisionId
- revisionTimestamp
- revisionUsername
- revisionUsernameId
- text

Where the important information is stored in the columns named “title”, “id” and “text”.

To create the forward link matrix, you will have to identify how the outgoing links are noted (they are identified with double “[[]]”, and select the row “id” for this hyperlink.

In this way, you can create a matrix with sparse values with a source page id in one column, and a set of target page ids in another column.

Once created the forward links matrix, you can create the backward links matrix, and use the algorithm described in Brin and Page’s paper to set the PageRank value for each webpage (record).

EXPECTED RESULTS:

You must use Apache Spark Dataframes to handle the Wikipedia Database and intermediate results.

The final result must be a Pandas DataFrame with the columns: “title”, ”id” and ”pagerank” with the values for each one of the webpages/records in Wikipedia.

EVALUATION:

There are five items to evaluate:

- Your code runs without errors (20%)
- Your code execution is scalable: We will test your code with other Wikipedia databases ten times larger in a cluster with 80 execution workers. (20%)
- Code optimization: (20%)
- Code documentation. (10%)
- Conclusions. (30%)

SUGGESTIONS:

Implement a User Defined Function (UDF), which parses the text field from each record, and extracts the outgoing links. All the external references and resources can be ignored.

With the parsed version DataFrames, search and replace the links with the ids of the corresponding documents, to handle just numbers. You can handle this information as a Sparse Matrix.

Use a smaller version (10%) of the Database, to test your algorithms.

Set a maximum of 20 iterations to calculate the PageRank, if your algorithm does not converge in this number of iterations, the loop must stop and show the end results.