Emulation based system for distributed file storage and parallel computation

Team Members & Responsibilities:

Team Members:

1. Lahari Palem
2. Rohan Chaudhari

Team Number: 30 (Google Sheet)
29 (USC Den)

Responsibilities:

1. Lahari : Part 2 implementation , Part 1 Testing and Part 3 implementation and testing
2. Rohan : Part 1 implementation, Part 2 Testing and Part 3 implementation and testing

Background:

We as a team having the knowledge on spark, python, java,SQL, flask have come up with the below road map for the implementation of Emulation based system for distributed file storage and parallel computation

Introduction:

Through our project, we aim to emulate the hadoop distributed file system using MySQL-based emulation.

Datasets:

The particular link listed below is a cricket dataset that consists of 2 CSV files:

1. Ball by Ball data
2. Complete Match data

Here, we plan on using match_id from both the datasets to join them and carry out various analytical operations.

https://www.kaggle.com/datasets/vora1011/ipl-2022-match-dataset?select=IPL_Matches_2022.csv

Overview:

Task 1:

Task 1 corresponds to developing the EDFS and associated file system functionalities for it. For this task we plan on using the Flask framework of Python along with REST API calls. Every file system command will have a specific API call associated with it which would run a function to carry out the corresponding SQL query in the backend database to replicate that particular file system command in MySQL database terms. The backend consists of multiple tables, one table will store the metadata of files, including file id, filename, filetype and location of partition whereas another table would store the directory structure of file, parent file and an array of child files. The CSV would be partitioned into k partitions depending on the input given by the user and stored in K different tables.

Task 2:

Task 2 corresponds to developing a MapReduce function on the dataset tables. Based on the pre-defined analytical query a mapPartition function that maps the partitions for the query and the reduce operation to fetch results will be implemented. This will be developed using SPARK map and reduce functions.

Task 3:

Task 3 corresponds to building an UI for the EDFS. We would be fulfilling this requirement using Jupyter Notebook, where for Task 1 we take an input from the user s to which file system command it wants to execute and carry out the execution. Additionally for the search and analytical queries, we will be taking the input as an SQL statement and outputting the final results in addition to the input given to the map function and reduce function (Intermediate results) and will facilitate the execution steps of the related query.

Timeline:

Mid-term report:

By Mid-term we will be able to

1.Enhance the datasets and build TWO tables for metadata and directory structure tables.

2.As per guidelines for a team of two members, we will come up with 6 tables (2 Data datasets, 2 metadata and 2 directory structure tables.)

3.Build EDFS by using MySQL and Rest API for partitioned datasets.
4.Plug in SPARK cluster to the datasets to come up with map reduce functions.

Post Mid-term report:
After Mid-term for Final Report we will be able to
1.Implement MapPartition function using spark where it takes input partition, performs MapReduce function on the partitions and fetches output from those partitions respectively for a given analytical query.
2. Developing User-Interface for EDFS to facilitate operations of Task 1 i.e, mkdir,cd,ls,getPartition etc.using Jupyter Notebook.
3.For the search and analytical queries, we will build getRequests to take SQL statements as input and output the final results where it points to the mapreduce functions and explains the steps that fetches the output of a query.
4.Testing of EDFS pipeline and submitting for evaluation.