BIG DATA SYSTEMS – ASSIGNMENT 2

Title:

Implement an efficient data layout and retrieval strategy for a Hadoop Cluster

Overview & background:

A database may have some common data repeated across records. For example, in the attached CSV file (that is exported from a database) some column values are same among multiple rows. These common values are repetitively stored in the database records, which increases the storage cost but reduces retrieval time for analytical queries.

However, we need to create a layout of this kind of dataset on a Hadoop cluster at a reduced storage cost. So, we need to understand the commonality of values across records and create a data layout that avoids duplicate values. But at the same time, we need to allow retrieval of a complete data record from the storage, given a record identifier.

Input: CSV data with flat schema with multiple records and features.

Description:

1. STORAGE:

Each Storage Node will store the data based on below condition.

a. Mutually Exclusive feature data (column value) which is not common across records (rows): private node
b. Feature data common in two records: 2-way shared node
c. Feature data common in four records: 4-way shared node.
d. Feature data common in eight records: 8-way shared node.

Note: Private node, 2,4,8- way shared nodes are storage nodes which stores feature values which are common in 2, 4, 8 records respectively.

2. METADATA

Maintain record ID wise metadata about above storage deployments, which will explain how the feature values are stored across the storage nodes. The meta-data can be stored on a specific node.

3. RETRIEVAL:

For provided record ID, retrieval of record will refer step 2 to fetch all the required features (column values) from respective storage nodes to form the original record.

NOTE: You can apply different techniques to understand the similarity of feature values like normalization, standardization, vectorization etc.

Submission Requirements:

1. A Python / Java / Spark code which enables
   a. the given CSV data to be written, using the distributed storage layout strategy described, to reduce duplicate data, and
   b. retrieval of any record given the record ID from the distributed storage.
2. Report compression ratio achieved using above approach, i.e. how much storage reduction happens using the de-duplicated data layout on the cluster.
3. **You can use a Hadoop cluster, a plain cluster of a set of nodes, or any BigData storage framework to demonstrate your data storage and retrieval code. Describe your setup in detail.**
4. You should provide clear instructions to reproduce the submission on the Evaluator's setup.
5. Your code and results should be reproducible
6. The implementation should be general purpose for any other CSV input file.

General Notes:

- Using Canvas, only the first member of the group has to upload the file. No submission over email will be considered.
- Submit the code and a document as a zip file. The document should be a Word or a PDF describing your setup, how to run your code and results as described in "Submission requirements".
- The document in the zip file should have full names of the group members along with the BITS Registration no. of each group member.
- Name the zip file in format like "Grp_<your_group_number>.zip" only. Don't add anything into the file names.
- Make sure that you upload the file well ahead of the deadline. At the last moment, we have seen several groups have faced issues while doing the submissions.

- **Note - As it's a group assignment, only one submission is expected from each group. Unnecessarily don't upload the solution on individual basis. If it's observed, then the penalty (25% reduction) will be applicable on it.**
- **Plagiarism will be strictly dealt with and if found will result in cancellation of the Assignment and 0 marks being awarded to all the group members.**
- **The last date of submission will not be extended in any case.**