Contents

1. Abstract

The objective we achieved is creation of Data lakes on HDFS. User should be able to manage i.e. upload, download, find a file, search in a data lake.

2. About Metadata

Metadata for data lake is stored in a text file in metadata folder of HDFS. Metadata about each file loaded in lakes is stored. The entry on each line in metadata file correspond to a file and the details in a line are as follow:

- The file format
- the path in hdfs where the file is loaded
- the size of file in bytes
- the user who uploaded the file in lake
- the date and time when the file was uploaded
- the access permission on file

Each metadata field will consist of files uploaded in particular year, in specific month of a specific category.

3. Features provided in data lakes

1. Uploading a Specific file
2. Bulk upload
3. Downloading a specific file.
4. Downloading all or particular files at a particular path in hdfs.
5. View all files at a particular path.
6. Searching and downloading files related to a specific category
7. Searching files having specific format, in a specific category, belonging to specific year, month, date
8. Finding location of file in lake with a specific file name or containing a specific sequence of character.
9. Finding in which all files in the lake a specific term exist.
10. Viewing downloading all files created a specific year or specific year, month or specific year, month, day or year, month, day, format.
11. Deleting files at particular path.

4. Description of implementation of features:

Feature 1 & 2:

- This is implemented with the help of Hadoop Filesystem used in java. While uploading user needs to provide category of each file in a category_file (mapping file). The format of this file is as follows:
  Category<tab>filename
- According to category provided file is organized in HDFS directory of data lakes.

Feature 3:

- With the help of Hadoop Filesystem, this is done.

Feature 4 & Feature 5:

- Metadata stored is used for this. According to the path provided, it is identified which metadata file will contain this information. The identified metadata file is provided to Map Reduce framework to find files with matching path. In case of downloading files the same approach is used for finding the full path of files which lie at a particular path and they are downloaded to local from Hadoop Filesystem.

Feature 6:

- Metadata is referred to find the HDFS location of files belonging to a specific category. All this files are downloaded to local of user if user wants it.

Feature 7:

- According to the input provided by the user. HDFS Path is generated and the user is provided with the files at this location.

Feature 8:

- All the metadata files which are created for lake is provided as an input to Map Reduce framework. The Mapper helps to identify the hdfs paths of all file matching the filename or file pattern provided by user if it exists in the lake.

Feature 9:

- An inverted index is created for all the document in the lake with the help of Map Reduce framework with term(words) as the key in the inverted index and all file

path where the term occur as its values. If the searched key (user searched term) occur then all the values attached to that index is provided to the user.

Feature 10:

- Based on the Search condition the metadata files which will have these files are identified. This metadata files are provided as the input to Map Reduce framework to apply further filtering if required.

Feature 11:

- Files deleted from hadoop Filesystem.

5. Organization of data

While loading data in data lakes the following Structure is followed. Data is loaded According to Category. For Assigning Data to category the user is provided with an option of tagging each file with a category. If a category Exist files are added to it else a new category is created in HDFS. In case a user does not provide category to file Category is provided automatically, with the help of MapReduce framework the most frequently occurring word in the file is found and it is made the category of file. Automatic category allocation is only allowed for Text files format.