Implementing Map Reduce Programs with Hadoop Streaming

Map Reduce

MapReduce is a programming model and processing framework designed for processing and generating large datasets in a parallel and distributed manner. It was originally developed by Google and popularized by the Apache Hadoop project. The MapReduce model simplifies the development of large-scale data processing applications by abstracting the complexities of distributed computing.

Here are the key components and concepts of MapReduce:

1. Map Function:
    - The MapReduce process begins with the application of a map function to each input record.
    - The map function takes an input key-value pair and produces an intermediate set of key-value pairs.
2. Shuffling and Sorting:
    - The intermediate key-value pairs are shuffled and sorted based on the keys.
    - This process groups together all intermediate values associated with the same key.
3. Reduce Function:
    - The grouped intermediate key-value pairs are passed to reduce functions.
    - The reduce function takes a key and its associated values and produces the final output.

Hadoop Streaming

Hadoop Streaming is a utility that comes with the Hadoop distribution and allows you to use any executable or script as a mapper and/or reducer in a Hadoop MapReduce job. It enables you to write MapReduce programs in languages other than Java, such as Python, Perl, Ruby, or shell scripts. This flexibility makes it easier for developers who are more comfortable with scripting languages to work with Hadoop.

Hadoop Streaming provides a simple way to leverage the power of Hadoop for distributed data processing while using scripts or executables in languages other than Java. It's especially useful for quick prototyping and testing of MapReduce jobs.

Task 1: Install all the packages for Python streaming and MRJob in Hadoop

- **Log in as a root user**

```
su root

password: hadoop

change password: hadoop

new password: *********

retype: **********
```

- yum-config-manager --save --setopt=HDP-SOLR-2.6-100.skip_if_unavailable=true
- yum install https://repo.ius.io/ius-release-el7.rpm https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
- yum install python-pip
- pip install pathlib
- pip install mrjob==0.7.4
- pip install PyYAML==5.4.1
- yum install nano

Download the data from the repository

- wget https://github.com/DoctorVinay8097/bigdata/blob/main/u.data

Task 2: Write down the map reduce code for computing the rating count in the dataset

```
from mrjob.job import MRJob

from mrjob.step import MRStep

class RatingsCount(MRJob):

    def steps(self):

        return [

            MRStep(mapper = self.mapper,

                reducer = self.reducer)
        ]
    def mapper(self, _ , line):
        (user_id, film_id, rating, timestamp) = line.split("\t")
        yield rating, 1
    def reducer(self, key, values):
        yield key, sum(values)
if __name__  = '__main__':
    RatingsCount.run()
```

Execution

```
python demo.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar u.data
```

Task 3: Modify the above code for finding the films popularity