DS&BDL
Assignment 12

Title: Design a distributed application using MapReduce which

Processes a log file of a system.

Aim:

Write a code in JAVA to design a distributed application using MapReduce which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Objective:

By completing this task, students will learn the following

1. Hadoop Distributed File System.
2. MapReduce Framework.

Software/Hardware Requirements: 64-bit Open source OS-Linux, Java, Hadoop.

Theory:

Map and Reduce tasks in Hadoop-With in a MapReduce job there are two separate tasks map task and reduce task.

Map task- A MapReduce job splits the input dataset into independent chunks known as input splits in Hadoop which are processed by the map tasks in a completely parallel manner. Hadoop framework creates separate map task for each input split.

Reduce task- The output of the maps is sorted by the Hadoop framework which then becomes input to the reduce tasks.

Hadoop MapReduce framework operates exclusively on <key, value> pairs. In a MapReduce job, the input to the Map function is a set of <key, value> pairs and output is also a set of <key, value> pairs. The output <key, value> pair may have different type from the input <key, value> pair.

<K1, V1> -> map -> (K2, V2)

The output from the map tasks is sorted by the Hadoop framework. MapReduce guarantees that the input to every reducer is sorted by key. Input and output of the reduce task can be represented as follows.

<K2, list(V2)> -> reduce -> <K3, V3>

1. Input Data

Input to the MapReduce comes from HDFS where log files are stored on the processing cluster. By dividing log files into small blocks we can distribute them over nodes of Hadoop cluster. The format of input files to MapReduce is arbitrary but it is line-based for log files. As each line is considered as a one record as we can say one log.

2. MapReduce Algorithm

MapReduce is a simple programming model which is easily scalable over multiple nodes in a Hadoop cluster. MapReduce job is written in Java consisting of Map and Reduce function. MapReduce takes log file as an input and feeds each record in the log file to the Mapper. Mapper processes all the records in the log file and Reducer processes all the outputs from the Mapper and gives final reduced results.

Map Function: Input to the map method is the InputSplit of log file. It produces intermediate results in (key, value) pairs. For each occurrence of key it emits (key, „1") pair. If there are n occurrences of key, then it produces n (key, „1") pairs. OutputCollector is the utility provided by MapReduce framework to collect output from mapper and reducer and reporter is to report a progress of application.

```
Map(LongWritable key, Text value, OutputCollector output, Reporter reporter) { For each key in the value; EmitIntermediate(key, „1"); }
```

Reduce Function: Input to reduce method is (key, values) pairs. It sums together all counts emitted by map method. If input to the reduce method is (key, (1,1,1,….n times)) then it aggregates all the values for that key producing output (key, n) pair. OutputCollector and Reporter works in similar way as in map method.

```
reduce(Text key, Iterator values, OutputCollector output, Reporter reporter)

{

        int sum = 0;

        for each v in values;

         sum += ParseInt(v);

        output.collect(key,(sum));

}
```

3. Creating Pig Query

Pig queries are written in Pig Latin language. Pig Latin statements are generally organized in the following manner:
A LOAD statement reads data from the Hadoop file system.
A series of "transformation" statements process the data.
A STORE statement writes output to the Hadoop file system.

Conclusion:

In this assignment, we have learned what is HDFS and How Hadoop MapReduce framework is used to process a log file of system