Gopal Joshi
PE-18

BDA-LAB-05

* Aim: Implement Map reduce in Hadoop and perform file hendling in HDFS for any domain agriculture / healthcare etc.

* Objective:
1. To learn & implement Map Reduce program
2. To learn use of Map reduce processing in various Big data.

* Problem statement:
Write a java program to perform map reduce processing for input files related to weather Monitoring & sales data.

* Theory:
- Working of map reduce with suitable example.
1. Record Reader, 2 Map, 3 combine, 4 Partition, 5 shuffle & sort / Reduce 7. Output format.

① Job tracker: acts like a master
② Task tracker: act like slaves

- A job is divided into multiple tasks which are then run into multiple data nodes in a cluster.
- It is the responsibility of job tracker to coordinate the activity by scheduling tasks to run.

- In case of failure, the JT tracker can reschedule it a diffrent task tracker.

② Advantges:

~~+ Highly a [illegible]~~

1. Highly scalable
2. very self effective situation for business
3. Highley flexible
4. Taches very less time to proces very large data
5. Provides fault tolerance
6. Based on simple programing model

* Input: Text / csv file with data

* Output: year wise max temperature ~~fd~~

* Platform: Windows

* Conclusion: Thus performed map reduce

FAQs:

Q1) How are the keys and values, extracted from input files in the mapper?

→ The mapReduce framework operates on <key, value> pairs i.e framework views the output to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types

(Q2) Explain the command word to run a Map Reduce program from command line with example command

→ command:

```
hadoop jar path-of-jar-file name-of-driver
input-file-path-hdfs output-folder-name
```

eg:

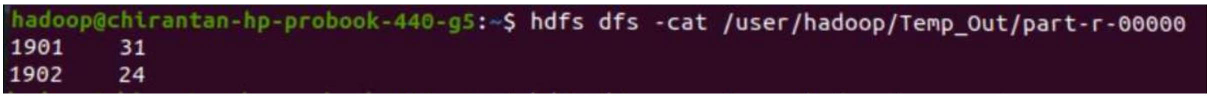
```
hadoop jar D:\MR.jar Driver /bda/input.txt.
   MR-out
```

(Q3) Discuss the importance of Job class in the Map Reduce program

→ The job class, allows the user to configure the job, submit it control into execution and query the state

Max Reducer

```
import java.io.IOException;

import java.util.Iterator;


import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;



public class MaxReducer extends MapReduceBase implements
Reducer<Text,IntWritable,Text,IntWritable>{


                public void reduce(Text key, Iterator<IntWritable> value1,

                        OutputCollector<Text, IntWritable> values, Reporter r)

                        throws IOException {


                        int maxvalue=Integer.MIN_VALUE;

                        while(value1.hasNext()){

                                maxvalue=Math.max(maxvalue, value1.next().get());

                        }

                        values.collect(key, new IntWritable(maxvalue));

        }

}
```



```
hadoop@chirantan-hp-probook-440-g5:~$ hdfs dfs -cat /user/hadoop/Temp_Out/part-r-00000
1901    31
1902    24
```