

Omkar Gurav

TE IT Batch & T3

Roll No:- 8048

Assignment-3

- Aim:- Design a distributed application using MapReduce which processes a logfile of a system. List out the users who have logged maximum times of the system. Use simple log file and process it using pseudo distribution mode on Hadoop platform.

- Theory:-

Q.1. Explain job execution in Hadoop.

→ MapReduce is a programming model designed to process huge amounts of data by dividing the job into independent local tasks.

When user submits a mapreduce job to Hadoop, the local job client prepares the job for submission & hands it off to the job tracker.

The job tracker schedules the job & distributes the work amongst multiple task trackers for parallel processing.

Each task tracker issues a map task. These tasks are assigned with task IDs. Job initialization & job

cleanup task are created & run by these task trackers.

Once mapping phase results are available, job tracker distributes the reduce work among the task trackers for parallel processing.

Each task tracker issues a reduce task to perform the work. Job tracker receives progress information from task trackers. Job client keeps polling the job tracker for progress.

Once job is completed, cleanup task gets processed. Task tracker sends the job completion status to the job tracker. Job tracker sends job completion message to the client. The tool process causes Job Clients `waitForJobToComplete` method to return.

Q.2. Explain following classes:-

① `IntWritable`



① `IntWritable` is the wrapper class in Hadoop which is similar to `Integer` class in Java. It is optimized to provide serialization in Hadoop.

② It implements Comparable, Writable & WritableComparable interfaces.

② Iterable

→ ① Java iterable interface represents a collection of objects which can be iterated. A class implementing this interface can have its elements iterated.

eg. `Iterable<String> = new Iterable[];`

③ Context

→ ① The context object allows the Mapper/Reducer to interact with the rest of the Hadoop system.

② It includes configuration for the job and provides functions to write to an area of memory the outputs of various tasks.

eg: `Context con;`

`con.write(key-val pair)`

- Conclusion:- Map Reduce application to process log file is Successfully implemented.

Driver class : Driver.java

```
package log;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Driver {

    @SuppressWarnings("deprecation")
    public static void main(String[] args) throws Exception{
        //creating object of configuration class
        Configuration c = new Configuration();

        //Assigning job to new configuration object
        Job job = new Job(c);

        //setting jar class
        job.setJarByClass(log.Driver.class);

        job.setMapperClass(log.LogMapper.class);

        job.setReducerClass(log.LogReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        //Adding a Path to the list of inputs
        FileInputFormat.addInputPath(job, new Path(args[0]));

        //Setting the Path of the output directory
        FileOutputFormat.setOutputPath(job,new Path(args[1]));

        //wait till job is completed
        job.waitForCompletion(true);

        //file system object
        FileSystem fs = FileSystem.get(c);

        FileStatus[] status = fs.listStatus(new
Path("hdfs://localhost:9000"+args[1]));
        FSDataInputStream fd = fs.open(status[1].getPath());

        String str = fd.readLine();
        String ip = "";
        int max = 0;

        while(str != null)
        {

            String parts[] = str.split("\t");

            if(max<Integer.parseInt(parts[1])) {

                max = Integer.parseInt(parts[1]);
                ip = parts[0];
            }

            str = fd.readLine();
        }

        System.out.println("IP address : " + ip);
        System.out.println("No. of occurrences : " + max);
    }
}
```

Mapper class : LogMapper.java

```
package log;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class LogMapper extends Mapper <LongWritable,Text,Text,IntWritable> {

    public void map(LongWritable key, Text value, Context con) throws IOException,
    InterruptedException {

        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);

        con.write(new Text((tokenizer.nextToken())),new IntWritable(1));

    }
}
```

Reducer class : LogReducer.java

```
package log;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class LogReducer extends Reducer <Text,IntWritable,Text,IntWritable> {

    public void reduce (Text word, Iterable<IntWritable> values, Context con)
    throws IOException, InterruptedException {

        int sum=0;

        for(IntWritable value : values)
        {
            sum += value.get();
        }

        con.write(word, new IntWritable(sum));

    }

}
```

Output Screenshots

```
hduser@omkar-VirtualBox:~$
hduser@omkar-VirtualBox:~$ hadoop jar /home/hduser/Downloads/log.jar log.Driver /Log/Log_input.txt /Log/Log_output 2/
21/05/22 16:53:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
21/05/22 16:53:39 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
21/05/22 16:53:39 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
21/05/22 16:53:39 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
21/05/22 16:53:40 INFO input.FileInputFormat: Total input files to process : 1
21/05/22 16:53:40 INFO mapreduce.JobSubmitter: number of splits:1
21/05/22 16:53:41 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local852939634_0001
21/05/22 16:53:42 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
21/05/22 16:53:42 INFO mapreduce.Job: Running job: job_local852939634_0001
21/05/22 16:53:42 INFO mapred.LocalJobRunner: OutputCommiter set in config null
21/05/22 16:53:42 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
21/05/22 16:53:42 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
21/05/22 16:53:42 INFO mapred.LocalJobRunner: OutputCommiter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
21/05/22 16:53:42 INFO mapred.LocalJobRunner: Waiting for map tasks
21/05/22 16:53:42 INFO mapred.LocalJobRunner: Starting task: attempt_local852939634_0001_m_000000_0
21/05/22 16:53:43 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
21/05/22 16:53:43 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
21/05/22 16:53:43 INFO mapred.MapTask: Using ResourceCalculatorProcessTree : [ ]
21/05/22 16:53:43 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/Log/Log_input.txt:0+143084
21/05/22 16:53:43 INFO mapreduce.Job: Job job_local852939634_0001 running in uber mode : false
21/05/22 16:53:43 INFO mapreduce.Job: map 0% reduce 0%
21/05/22 16:53:43 INFO mapred.MapTask: (EQUATOR) 0 kvt 26214396(104857584)
21/05/22 16:53:43 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
21/05/22 16:53:43 INFO mapred.MapTask: soft limit at 83886080
21/05/22 16:53:43 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
21/05/22 16:53:43 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
21/05/22 16:53:43 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
21/05/22 16:53:46 INFO mapred.LocalJobRunner:
21/05/22 16:53:46 INFO mapred.MapTask: Starting flush of map output
21/05/22 16:53:46 INFO mapred.MapTask: Spilling map output
21/05/22 16:53:46 INFO mapred.MapTask: bufstart = 0; bufend = 22902; bufvoid = 104857600
21/05/22 16:53:46 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26209220(104836880); length = 5177/6553600
21/05/22 16:53:46 INFO mapred.MapTask: Finished spill 0
21/05/22 16:53:47 INFO mapred.Task: Task:attempt_local852939634_0001_m_000000_0 is done. And is in the process of committing
21/05/22 16:53:47 INFO mapred.LocalJobRunner: map
21/05/22 16:53:47 INFO mapred.Task: Task:attempt_local852939634_0001_m_000000_0 done.
21/05/22 16:53:47 INFO mapred.LocalJobRunner: Finishing task: attempt_local852939634_0001_m_000000_0
21/05/22 16:53:47 INFO mapred.LocalJobRunner: map task executor complete.
21/05/22 16:53:47 INFO mapred.LocalJobRunner: Waiting for reduce tasks
21/05/22 16:53:47 INFO mapred.LocalJobRunner: Starting task: attempt_local852939634_0001_r_000000_0
21/05/22 16:53:47 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
21/05/22 16:53:47 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
21/05/22 16:53:47 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
21/05/22 16:53:47 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@63bb4eb8
21/05/22 16:53:47 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=363285696, maxSingleShuffleLimit=90821424, mergeThreshold=239768576, ioSortFactor=10, memToMemMergeOutputsThreshold=10
21/05/22 16:53:47 INFO reduce.EventFetcher: attempt_local852939634_0001_r_000000_0 Thread started: EventFetcher for fetching Map Completion Events
21/05/22 16:53:47 INFO mapreduce.Job: map 100% reduce 0%
```

```
21/05/22 16:53:49 INFO mapred.LocalJobRunner: Finishing task: attempt_local852939634_0001_r_000000_0
21/05/22 16:53:49 INFO mapred.LocalJobRunner: reduce task executor complete.
21/05/22 16:53:49 INFO mapreduce.Job: map 100% reduce 100%
21/05/22 16:53:50 INFO mapreduce.Job: Job job_local852939634_0001 completed successfully
21/05/22 16:53:50 INFO mapreduce.Job: Counters: 35
File System Counters
  FILE: Number of bytes read=59926
  FILE: Number of bytes written=1031084
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=286168
  HDFS: Number of bytes written=3611
  HDFS: Number of read operations=13
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4
Map-Reduce Framework
  Map input records=1295
  Map output records=1295
  Map output bytes=22902
  Map output materialized bytes=25498
  Input split bytes=104
  Combine input records=0
  Combine output records=0
  Reduce input groups=227
  Reduce shuffle bytes=25498
  Reduce input records=1295
  Reduce output records=227
  Spilled Records=2590
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=130
  Total committed heap usage (bytes)=351805440
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=143084
File Output Format Counters
  Bytes Written=3611
IP address : 10.02.30.199
No. of occurrences : 63
hduser@omkar-VirtualBox:~$
```