SL-V BE IT
EXP 5 Part B (According to new revised assignments)

Aim: Design a distributed application using MapReduce under Hadoop for:
b) Counting no. of occurrences of every word in a given text file.

Steps:
First install hadoop (if not installed yet) by,
https://sl6it.wordpress.com/2015/12/04/1-study-and-configure-hadoop-for-big-data/

# Download **sample.txt** file (attached with this post)
# Paste sample.txt in your **home** folder

# Open terminal
whoami
# It will display your user name, we will use it later.

# Open eclipse->new java project->project name **exp5b**->new class-> **`WordCount`**

# Add following code in that class

```
package exp5b;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
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.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf,
args).getRemainingArgs();
    if (otherArgs.length < 2) {
      System.err.println("Usage: wordcount <in> [<in>...] <out>");
      System.exit(2);
    }
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    for (int i = 0; i < otherArgs.length - 1; ++i) {
      FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }
    FileOutputFormat.setOutputPath(job,
      new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

# Save the file

# It will display some errors, so we are going to import three jar files in our project.

# Copy hadoop-mapreduce-client-core-2.7.1.jar from ~/hadoop/share/hadoop/mapreduce directory
# In eclipse-> right click on exp5b project- >paste
# Right click on pasted hadoop-mapreduce-client-core-2.7.1.jar-> Buid path-> add to buid path

#Copy hadoop-common-2.7.1.jar from ~/hadoop/share/hadoop/common directory
# In eclipse-> right click on exp5b project- >paste
# Right click on pasted hadoop-common-2.7.1.jar-> Buid path-> add to buid path

#Copy commons-cli-1.2.jar from ~/hadoop/share/hadoop/common/lib directory
# In eclipse-> right click on exp5b project- >paste
# Right click on pasted commons-cli-1.2.jar-> Buid path-> add to buid path

# In eclipse->Right click on project exp5b-> export->java->jar file->next-> select the export destination -> /home/**your_user_name**/exp5b.jar -> next -> next -> select main class ->browse -> **WordCount** -> finish

# **exp5b.jar** file will be created in your home folder

# Open terminal

```
# Now Start NameNode daemon and DataNode daemon:

        ~/hadoop/sbin/start-dfs.sh

# Make the HDFS directories required to execute MapReduce jobs

        ~/hadoop/bin/hdfs dfs -mkdir /user

        ~/hadoop/bin/hdfs dfs -mkdir /user/your_user_name


# Put sample.txt file in hdfs

        ~/hadoop/bin/hdfs dfs -put ~/sample.txt input_data

# Perform MapReduce job

        ~/hadoop/bin/hadoop jar ~/exp5b.jar input_data output_data


# Output
        ~/hadoop/bin/hdfs dfs -cat output_data/*

# Our task is done, so delete the distributed files (input_data & output_data)
        ~/hadoop/bin/hdfs dfs -rm -r input_data output_data

# Stop haddop

        ~/hadoop/sbin/stop-dfs.sh

jps
```

Reference : Hadoop the definitive guide, O’Reilly Publications, by Tom White

----

Prof. S. T. Kolhe
(Department of I.T – S.R.E.S C.O.E Kopargaon)