**Assignment No:**

**Assignment Title:** Design a distributed application using MapReduce and Hadoop for:

1. Character counting in a given text file: [Implementation Logic](#_Design_a_distributed)
2. Counting no. of occurrences of every word in a given text file.: [Implementation Logiccc](#_Aim:_a)_Character)

Objective:

1. To learn the how to use the MapReduce and Hadoop
2. To learn how to write MapReduce Program in Java or C++ or Python

Input: Hello Hadoop

Expected Output: Total Number of characters are 14.

**Implementation Details:**

# Design a distributed application using MapReduce under Hadoop for Character counting in a given text file.

Steps:

First install hadoop

# Create sample.txt file and 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 exp5a->new class-> CharMap

#Add following code in that class

package exp5a;

import java.io.IOException;

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 CharMap extends Mapper<LongWritable, Text, Text, IntWritable> {

public void map(LongWritable key, Text value, Context context)

throws IOException, InterruptedException {

String line = value.toString();

char[] carr = line.toCharArray();

for (char c : carr) {

System.out.println(c);

context.write(new Text(String.valueOf(c)), new IntWritable(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 exp5a 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 exp5a 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 exp5a project- >paste

# Right click on pasted commons-cli-1.2.jar-> Buid path-> add to buid path

# In eclipse->right click on project exp5a->new class-> CharReduce

#Add following code in that class

package exp5a;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

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

public void reduce(Text key,Iterable<IntWritable> values,Context

context)throws IOException,InterruptedException{

int count = 0;

IntWritable result = new IntWritable();

for (IntWritable val : values) {

count +=val.get();

result.set(count);

}

context.write(key, result);

}

}

# Save the file

# In eclipse->right click on project exp5a->new class-> CharCount

#Add following code in that class

package exp5a;

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.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class CharCount {

public static void main(String[] args) throws Exception {

// TODO Auto-generated method stub

Configuration conf = new Configuration();

@SuppressWarnings("deprecation")

Job job = new Job(conf, "Charcount");

job.setJarByClass(CharCount.class);

job.setMapperClass(CharMap.class);

job.setReducerClass(CharReduce.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path( args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

# Save the file

# In eclipse->Right click on project exp5a-> export->java->jar file->next-> select the export

destination -> /home/your\_user\_name/exp5a.jar -> next -> next -> select main class ->browse ->

CharCount -> finish

# exp5a.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 ~/exp5a.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

-----------------------------------------------------------------------------------------------------------------

# 

# Aim: a) Character counting in a given text file.

**Steps:**

First install hadoop.

# Create the sample.txt file 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 exp5a->new class-> CharMap

#Add following code in that class

package exp5a;

import java.io.IOException;

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 CharMap extends Mapper<LongWritable, Text, Text, IntWritable> {

public void map(LongWritable key, Text value, Context context)

throws IOException, InterruptedException {

String line = value.toString();

char[] carr = line.toCharArray();

for (char c : carr) {

System.out.println(c);

context.write(new Text(String.valueOf(c)), new IntWritable(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 exp5a 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 exp5a 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 exp5a project- >paste

# Right click on pasted commons-cli-1.2.jar-> Buid path-> add to buid path

# In eclipse->right click on project exp5a->new class-> CharReduce

#Add following code in that class

package exp5a;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

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

public void reduce(Text key,Iterable<IntWritable> values,Context

context)throws IOException,InterruptedException{

int count = 0;

IntWritable result = new IntWritable();

for (IntWritable val : values) {

count +=val.get();

result.set(count);

}

context.write(key, result);

}

}

# Save the file

# In eclipse->right click on project exp5a->new class-> CharCount

#Add following code in that class

package exp5a;

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.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class CharCount {

public static void main(String[] args) throws Exception {

// TODO Auto-generated method stub

Configuration conf = new Configuration();

@SuppressWarnings("deprecation")

Job job = new Job(conf, "Charcount");

job.setJarByClass(CharCount.class);

job.setMapperClass(CharMap.class);

job.setReducerClass(CharReduce.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path( args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

# Save the file

# In eclipse->Right click on project exp5a-> export->java->jar file->next-> select the export

destination -> /home/your\_user\_name/exp5a.jar -> next -> next -> select main class ->browse ->

CharCount -> finish

# exp5a.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 ~/exp5a.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:

1. YouTube Video

[HOW TO INSTALL HADOOP 2 9 0 SINGLE NODE CLUSTER AND RUN WORDCOUNT PROGRAM ON UBUNTU 16 04 - YouTube](https://www.youtube.com/watch?v=5rJTPMLKsq0)

1. Hadoop the definitive guide, O’Reilly Publications, by Tom White