WordCount-Example-in-Hadoop

Enabling technologies for Data Science

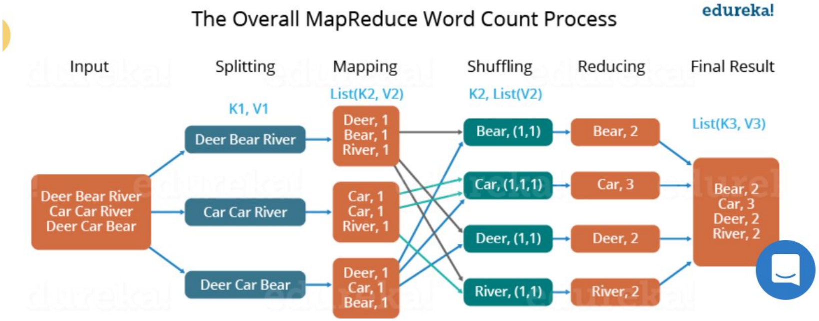
2017msbda008
Samiksha Agarwal

August 6, 2018

Running a Map/Reduce job

What is MapReduce?

MapReduce is a programming framework that allows us to perform distributed and parallel processing on large data sets in a distributed environment.
Let us understand, how a MapReduce works by taking an example where I have a text file whose contents are as follows:
Dear, Bear, River, Car, Car, River, Deer, Car and Bear
Now, suppose, we have to perform a word count using MapReduce. So, we will be finding the unique words and the number of occurrences of those unique words.



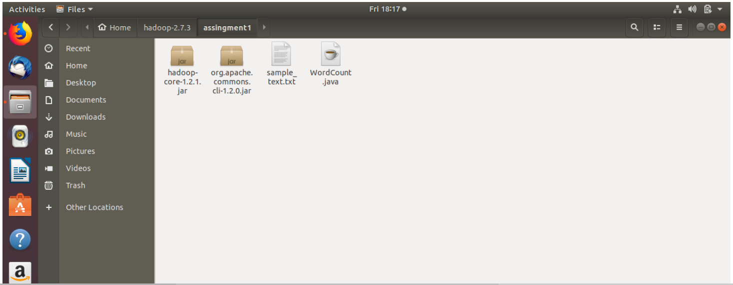
we will understand this process by program and for details of each step you can follow this link Click Here

Steps for run the program :

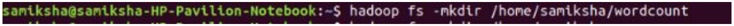
First, start all daemons

Command: ./start-all.sh

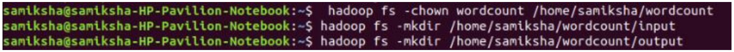
- Step 1: Download the hadoop-1.2.1.jar.zip file Click Here
- Step 2: Download org.apache.commons.cli-1.2.0.jar.zip Click Here
- Step 3: Create a folder (assingment1) and extract both zip file here.
- Step 4: Make java file (WordCount.java) and for making this file you can prefer this link Click Here
- Step 5: Make a text file.



- Step 5: crtl+alt+t (open terminal).
- Step 6: Before you run the sample, you must create input and output locations in HDFS. Use the following commands to create the input directory /user/wordcount/input in HDFS:
  Command: hadoop fs -mkdir /home/samiksha/wordcount
  Command: hadoop fs -chown wordcount /home/samiksha/wordcount
  Command: hadoop fs -mkdir /home/samiksha/wordcount /input
  Command: hadoop fs -mkdir /home/samiksha/wordcount /output

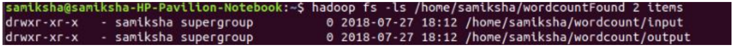


```
samiksha@samiksha-HP-Pavilion-Notebook:~$ hadoop fs -mkdir /home/samiksha/wordcount
```



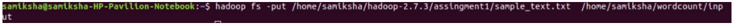
```
samiksha@samiksha-HP-Pavilion-Notebook:~$  hadoop fs -chown wordcount /home/samiksha/wordcount
samiksha@samiksha-HP-Pavilion-Notebook:~$ hadoop fs -mkdir /home/samiksha/wordcount/input
samiksha@samiksha-HP-Pavilion-Notebook:~$ hadoop fs -mkdir /home/samiksha/wordcount/output
```

  you can check your input and output file
  Command: hadoop fs -ls /home/samiksha/wordcount



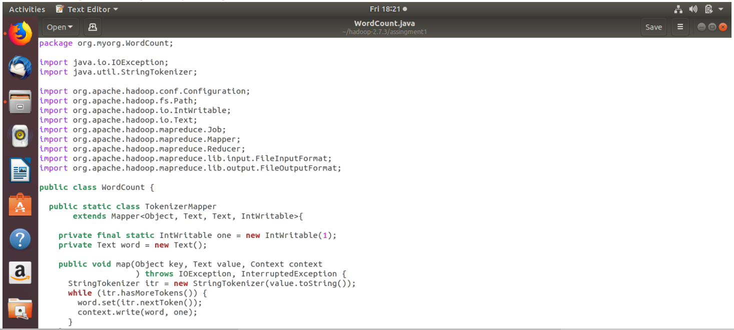
```
samiksha@samiksha-HP-Pavilion-Notebook:~$ hadoop fs -ls /home/samiksha/wordcountFound 2 items
drwxr-xr-x   - samiksha supergroup          0 2018-07-27 18:12 /home/samiksha/wordcount/input
drwxr-xr-x   - samiksha supergroup          0 2018-07-27 18:12 /home/samiksha/wordcount/output
```

- Step 7: move your sample_text.txt to /home/samiksha/wordcount /input directory in HDFS.
  Command: hadoop fs -put /home/samiksha/hadoop-2.7.3/assingment1/sample_text.txt /home/samiksha/wordcount /input



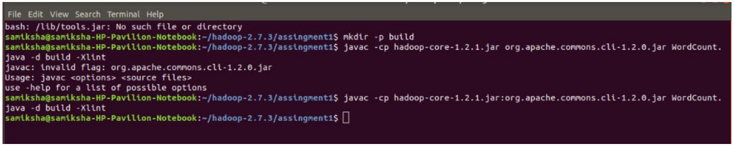
```
samiksha@samiksha-HP-Pavilion-Notebook:~$ hadoop fs -put /home/samiksha/hadoop-2.7.3/assingment1/sample_text.txt  /home/samiksha/wordcount/inp
ut
```

- Step 8: open WordCount.java and You can use an appropriate package for your domain, or keep the generic version.
  package org.myorg.WordCount;



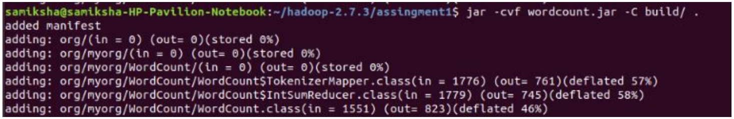
```
package org.myorg.WordCount;

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;

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);
      }
```

- step 9: now, go to the folder assingment1 and open terminal.
- Step 10: Compile the WordCount class.
  Command: mkdir -p build
  Command: javac -cp
  hadoop-core-1.2.1.jar:org.apache.commons.cli-1.2.0.jar
  WordCount.java -d build -Xlint



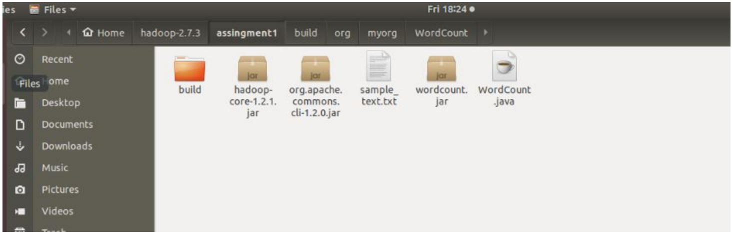
```
File  Edit  View  Search  Terminal  Help
bash: /lib/tools.jar: No such file or directory
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ mkdir -p build
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ javac -cp hadoop-core-1.2.1.jar org.apache.commons.cli-1.2.0.jar WordCount.
java -d build -Xlint
javac: invalid flag: org.apache.commons.cli-1.2.0.jar
Usage: javac <options> <source files>
use -help for a list of possible options
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ javac -cp hadoop-core-1.2.1.jar:org.apache.commons.cli-1.2.0.jar WordCount.
java -d build -Xlint
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ 
```

- step 11: Create a JAR file for the WordCount application.
  Command: jar -cvf wordcount.jar -C build/ .

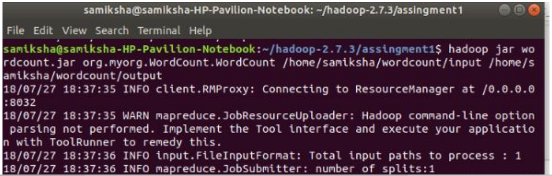


```
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ jar -cvf wordcount.jar -C build/ .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/myorg/(in = 0) (out= 0)(stored 0%)
adding: org/myorg/WordCount/(in = 0) (out= 0)(stored 0%)
adding: org/myorg/WordCount/WordCount$TokenizerMapper.class(in = 1776) (out= 761)(deflated 57%)
adding: org/myorg/WordCount/WordCount$IntSumReducer.class(in = 1779) (out= 745)(deflated 58%)
adding: org/myorg/WordCount/WordCount.class(in = 1551) (out= 823)(deflated 46%)
```

you can check jar file into the assingment1 folder :

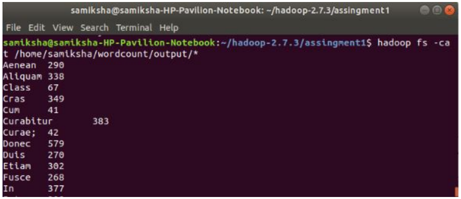


- step 12: Run the WordCount application from the JAR file, passing the paths to the input and output directories in HDFS.
  Command: hadoop jar wordcount.jar org.myorg.WordCount.WordCount /home/samiksha/wordcount/input /home/samiksha/wordcount/output


samiksha@samiksha-HP-Pavilion-Notebook: ~/hadoop-2.7.3/assingment1
File Edit View Search Terminal Help

```
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ hadoop jar wo
rdcount.jar org.myorg.WordCount.WordCount /home/samiksha/wordcount/input /home/s
amiksha/wordcount/output
18/07/27 18:37:35 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
18/07/27 18:37:35 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
18/07/27 18:37:36 INFO input.FileInputFormat: Total input paths to process : 1
18/07/27 18:37:36 INFO mapreduce.JobSubmitter: number of splits:1
```

- step 12: When you look at the output, The number of occurrences from all input files has been reduced to a single sum for each word.
  Command: hadoop fs -cat /home/samiksha/wordcount/output/*

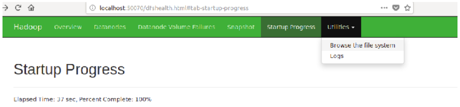

samiksha@samiksha-HP-Pavilion-Notebook: ~/hadoop-2.7.3/assingment1
File Edit View Search Terminal Help

```
samiksha@samiksha-HP-Pavilion-Notebook:~/hadoop-2.7.3/assingment1$ hadoop fs -ca
t /home/samiksha/wordcount/output/*
Aenean  290
Aliquam 338
Class   67
Cras    349
Cum     41
Curabitur       383
Curae;  42
Donec   579
Duis    270
Etiam   302
Fusce   268
In      377
```

- Step 13: If you want to run the sample again, you first need to remove the output directory. Use the following command.
  Command: hadoop fs -rm -r /home/samiksha/wordcount/output
  Don't forget to stop all daemons
  Command: ./stop-all.sh

if you want check the output file

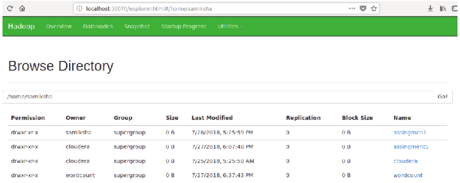
go to Utilities → Browse the file system



Startup Progress

Elapsed Time: 37 sec, Percent Complete: 100%

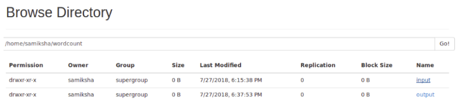
now, go to home → samiksha → wordcount



Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	samiksha	supergroup	0 B	7/26/2018, 5:25:59 PM	0	0 B	assingmen1
drwxr-xr-x	cloudera	supergroup	0 B	7/27/2018, 6:07:48 PM	0	0 B	assingment1
drwxr-xr-x	cloudera	supergroup	0 B	7/25/2018, 5:25:58 AM	0	0 B	cloudera
drwxr-xr-x	wordcount	supergroup	0 B	7/27/2018, 6:37:43 PM	0	0 B	wordcount

here is your input and output file

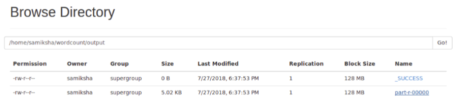


Browse Directory

/home/samiksha/wordcount Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	samiksha	supergroup	0 B	7/27/2018, 6:15:38 PM	0	0 B	input
drwxr-xr-x	samiksha	supergroup	0 B	7/27/2018, 6:37:53 PM	0	0 B	output

now, go to output →



Browse Directory

/home/samiksha/wordcount/output Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	samiksha	supergroup	0 B	7/27/2018, 6:37:53 PM	1	128 MB	_SUCCESS
-rw-r--r--	samiksha	supergroup	5.02 KB	7/27/2018, 6:37:53 PM	1	128 MB	part-r-00000

open part-r-0000 → download the file

YARN Web UI

(for more details you can follow this link Click Here)

Thank You