LAB 1

MAP REDUCE IN JAVA

Vidhi Kapoor – J021

Kartikay Laddha – J025

PROCEDURE ON CLOUDERA –

In eclipse, we make 3 classes and then add 2 jar files to it.

Process to add 2 jars:

Right click word count

Build path

Configure build path

Add external jar

Click on Root directory

Usr folder

Folder Lib

Hadoop 0.20 mapreduce

Add **Hadoop core 2.6.0 mr1-CDl 5.13.0 jar** file

And then again add one file

That is in lib folder

Hadoop folder

Add **Hadoop common 2.6.0 cdh5.13.0 jar** file

These 2 jar files we add

And then again right click on word count, we export it to jar file.

If everything gets executed perfectly, then we see cloudera home → workspace → we can see wordcount.jar file

Now to execute the code, go to lab 1 where WCFile.txt is saved.

Open in terminal:

(base) [cloudera@quickstart Lab_1]$ **cat WCFile.txt**

Apache Hadoop (/həˈduːp/) is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. Hadoop was originally designed for computer clusters built from commodity hardware, which is still the common use.[3] It has since also found use on clusters of higher-end hardware.[4][5] All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.[6]

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality,[7] where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.[8][9]

(base) [cloudera@quickstart Lab_1]$ **hadoop fs -put WCFile.txt WCFile.txt**

put: `WCFile.txt': File exists

(base) [cloudera@quickstart Lab_1]$ (base) [cloudera@quickstart Lab_1]$ cat WCFile.txt

bash: syntax error near unexpected token `[cloudera@quickstart'

(base) [cloudera@quickstart Lab_1]$ Apache Hadoop (/həˈduːp/) is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. Hadoop was ^C

(base) [cloudera@quickstart Lab_1]$

After saving the file in Hadoop system, we open the terminal in workspace.

(base) [cloudera@quickstart workspace]$ **hadoop jar WordCount.jar WCDriver WCFile.txt WCOutput**

21/03/05 23:56:14 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

21/03/05 23:56:15 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

21/03/05 23:56:15 WARN security.UserGroupInformation: PriviledgedActionException as:cloudera (auth:SIMPLE) cause:org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://quickstart.cloudera:8020/user/cloudera/WCOutput already exists

21/03/05 23:56:15 WARN security.UserGroupInformation: PriviledgedActionException as:cloudera (auth:SIMPLE) cause:org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://quickstart.cloudera:8020/user/cloudera/WCOutput already exists

Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory hdfs://quickstart.cloudera:8020/user/cloudera/WCOutput already exists

at org.apache.hadoop.mapred.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:131)

at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java:272)

at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:143)

at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1307)

at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1304)

at java.security.AccessController.doPrivileged(Native Method)

at javax.security.auth.Subject.doAs(Subject.java:415)

at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1917)

at org.apache.hadoop.mapreduce.Job.submit(Job.java:1304)

at org.apache.hadoop.mapred.JobClient$1.run(JobClient.java:578)

at org.apache.hadoop.mapred.JobClient$1.run(JobClient.java:573)

at java.security.AccessController.doPrivileged(Native Method)

at javax.security.auth.Subject.doAs(Subject.java:415)

at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1917)

at org.apache.hadoop.mapred.JobClient.submitJobInternal(JobClient.java:573)

at org.apache.hadoop.mapred.JobClient.submitJob(JobClient.java:564)

at org.apache.hadoop.mapred.JobClient.runJob(JobClient.java:873)

at WCDriver.run(WCDriver.java:34)

at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)

at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:84)

```
        at WCDriver.main(WCDriver.java:41)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
(base) [cloudera@quickstart workspace]$ hadoop fs -cat WCOutput/part-00000
(       1
(HDFS), 1
/həˈduːp/)      1
All     1
Apache  2
Distributed     1
File    1
Hadoop  6
It      3
MapReduce       2
System  1
The     1
This    2
a       10
access  1
across  1
advantage       1
allows  1
also    1
amounts         1
and     7
approach        1
```

architecture	1
are	3
as	1
assumption	1
automatically	1
be	3
big	1
blocks	1
built	1
by	1
cluster.	1
clusters	2
code	1
collection	1
commodity	1
common	2
computation	1
computation.	1
computer	1
computers	1
consists	1
conventional	1
core	1
data	6
dataset	1
designed	2
distributed	2
distributes	1
efficiently	1
facilitates	1
failures	1

faster 1

file 1

files 1

for 2

found 1

framework 1

framework.[6] 1

from 1

fundamental 1

handled 1

hardware 1

hardware, 1

hardware.[4][5] 1

has 1

have 1

high-speed 1

higher-end 1

in 4

into 2

involving 1

is 3

it 1

known 1

large 1

locality,[7] 1

manipulate 1

many 1

massive 1

model. 2

modules 1

more 2

network 1

networking.[8][9] 1

nodes 3

occurrences 1

of 8

on 2

open-source 1

originally 1

packaged 1

parallel 1

parallel.1

part 1

part, 1

problems 1

process 1

processed 1

processing 2

programming 2

provides 1

relies 1

should 1

since 1

software 2

solve 1

splits 1

still 1

storage 2

supercomputer 1

system 1

takes 1

than 1

that	3
the	7
them	1
then	1
they	1
to	3
to.	1
transfers	1
use	1
use.[3]	1
using	2
utilities	1
via	1
was	1
where	2
which	2
with	1
would	1