



School of Science and Engineering

Project 2: Batch Processing of Logs Using MapReduce and Hive

Lmoubariki Meriem

Dr. Tajjeeddine Rachidi

Al Akhawayn University in Ifrane

October 03, 2024

Table of Contents

- 1. Introduction**
- 2. Setup and Configuration**
- 3. Verification of Hadoop and Hive**
- 4. WordCount Job Execution**
- 5. Writing Nginx Log Files onto HDFS**
- 6. First MapReduce Job**
- 7. Second MapReduce Job**
- 8. Querying Using Hive**
- 9. References**

Introduction

In today's data-driven world, the ability to process large volumes of data efficiently is crucial for gaining insights and making informed decisions. Apache Hive, sitting atop Hadoop, provides a powerful platform for data warehousing and SQL-like querying that is especially suited for handling large datasets. This report delves into the practical application of Hive for managing and querying data in a structured format.

Our focus is on setting up a Hive environment and performing a series of data operations that include creating databases, inserting data, and executing queries that range from basic retrieval to complex data aggregation. Through these operations, we demonstrate Hive's capabilities and its SQL-like language, HQL (Hive Query Language), which facilitates easy interaction with data stored in a distributed storage like Hadoop's HDFS.

Setup and Configuration

Verification of Hadoop and Hive

The screenshots demonstrate the successful initialization of Hive, which is crucial for conducting any data operations. These initial steps include launching the Hive shell and loading Apache Derby, the default backend used for managing Hive's metadata. The proper functioning of this setup is validated by logs that indicate the loading of necessary configurations and initialization sequences, confirming that the environment is ready for executing data processing tasks.

The console output confirms that Hive and its dependencies are properly configured and operational. This setup is pivotal as it ensures that Hive can access the Hadoop Distributed File System (HDFS) and execute queries without issues.

```
hdfs://295b5fd20c:/opt/hive/bin
2.Jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.access.uniquewithduplicatenullsort in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.classManagerJ2 in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.JNDIAuthentication in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.rawStore.log in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.access.heap in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.database.slave in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.env.jdbc.resourceAdapterJ2 in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.mgmt.null in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.env.classes.resourceAdapterJ2 in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.access in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.replication.slave in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.lf in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.uuid1 in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.access.btree in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main] Ignored duplicate property derby.module.rawStore in jar:file:/opt/apache-hive-3.1.2-bin/lib/hive-druid-handler-3.1.2.jar!/org/apache/derby/modules.properties
Tue Oct 08 22:19:42 GMT 2024 Thread[60634223;ade6-4d29-8177-82ee65e04ac2 main,5,main]
-----
Tue Oct 08 22:19:44 GMT 2024:
Booting Derby Version The Apache Software Foundation - Apache Derby - 10.14.1.0 - (1808820); instance a816c00e-0192-6e37-80e4-00000a8af100
on database directory /home/hdfs/.metastore_db/metastore_db with class loader sun.misc.Launcher$AppClassLoader@326de728
Loaded from file:/opt/apache-hive-3.1.2-bin/lib/derby-10.14.1.0.jar
java.vendor=Oracle Corporation
java.runtime.version=1.8.0_292-Bit/Bin-Ubuntu1-20.04-b10
user.dir=/opt/apache-hive-3.1.2-bin/bin
os.name=Linux
os.arch=x86_64
os.version=5.15.13.1-microsoft-standard-WSL2
derby.system.home=null
Database Class Loader started - derby.database.classpath=''''
Hive Session ID = abcf0bc-a755-4e90-9634-78ab1a85786b
hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive -
```

The creation of a database in Hive mimics traditional SQL database systems, showcasing

Hive's capability to organize data in a structured manner.

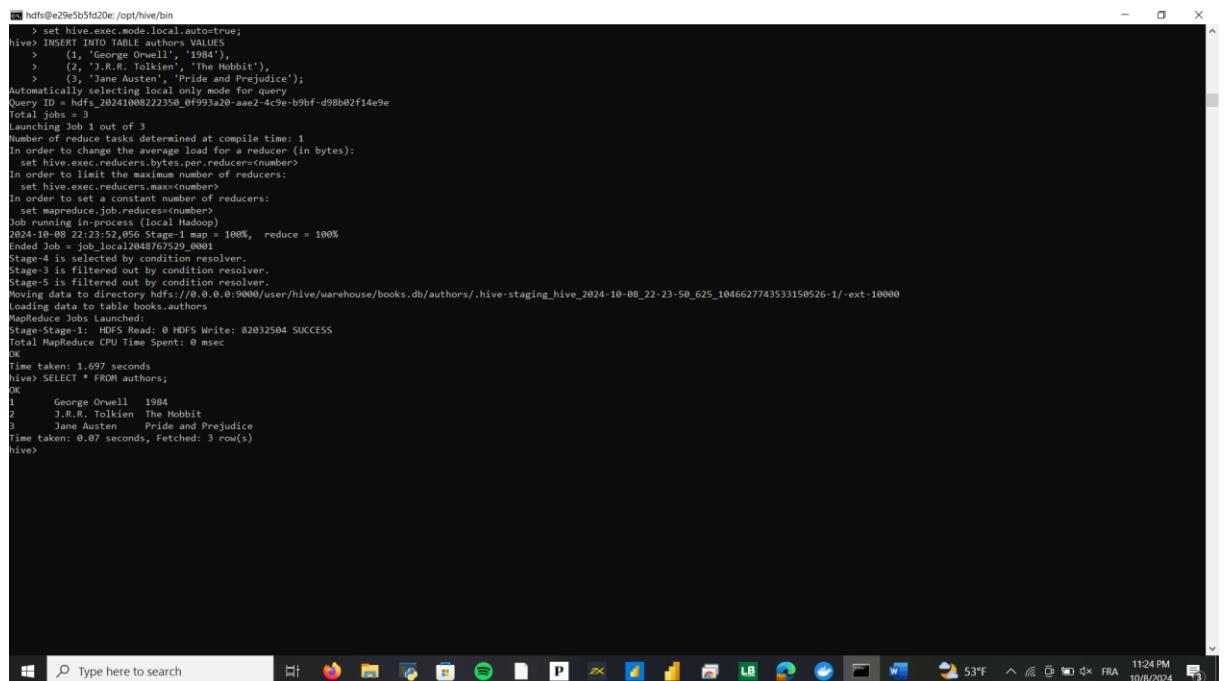
Once the database was set up, a table within this database was created and populated with

data. The table, also named `authors`, was structured to hold records with multiple fields: an

author's name, the publication year, and the title of the book. The use of the `INSERT INTO`

`TABLE` statement to populate the table demonstrates Hive's DML (Data Manipulation

Language) capabilities, which are crucial for any database operations.



```
hdfs@e29e5b5fd20e:/opt/hive/bin
> set hive.exec.mode.local.auto=true;
hive> INSERT INTO TABLE authors VALUES
>     (1, 'George Orwell', '1984'),
>     (2, 'J.R.R. Tolkien', 'The Hobbit'),
>     (3, 'Jane Austen', 'Pride and Prejudice');
Automatically selecting local only mode for query
Query ID = hdfs_20241008222350_0f9933a0-aee2-4c9e-b9bf-d98b02f14e9e
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the storage level for a reducer (in bytes):
  set mapreduce.reducer.bytes-per-reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reduces:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-10-08 22:53:52,956 [Stage-1 map 100%, reduce = 100%]
End-to-end Job: 2024-10-08 22:53:54,001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://0.0.0.0:9000/user/hive/warehouse/books.db/authors/.hive-staging_hive_2024-10-08_22-23-50_625_1046627743533150526-1/-ext-10000
Loading data to table books.authors
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 0 HDFS Write: 82052504 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.697 seconds
hive> SELECT * FROM authors;
OK
1    George Orwell 1984
2    J.R.R. Tolkien The Hobbit
3    Jane Austen   Pride and Prejudice
Time taken: 0.07 seconds, Fetched: 3 row(s)
hive>
```

```

> set hive.exec.mode.local.auto=true;
hive> INSERT INTO TABLE authors VALUES
    >     (1, 'George Orwell', '1984'),
    >     (2, 'J.R.R. Tolkien', 'The Hobbit'),
    >     (3, 'Jane Austen', 'Pride and Prejudice');
Automatically selecting local only mode for query
Query ID = hdfs_20241008222350_0f993a20-aac2-4c9e-b9bf-d98b02f14e9e
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-10-08 22:23:52,056 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local2048767529_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://0.0.0.0:9000/user/hive/warehouse/books.db/authors/.hive-staging_hive_2024-10-08_22-23-50_625_1046627743533150526-1/-ext-10000
Loading data to table books.authors
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 0 HDFS Write: 82032504 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.697 seconds
hive> SELECT * FROM authors;
OK
1      George Orwell 1984
2      J.R.R. Tolkien The Hobbit
3      Jane Austen  Pride and Prejudice
Time taken: 0.07 seconds, Fetched: 3 row(s)
hive>
```

```

Time taken: 1.697 seconds
hive> SELECT * FROM authors;
OK
1      George Orwell 1984
2      J.R.R. Tolkien The Hobbit
3      Jane Austen  Pride and Prejudice
Time taken: 0.07 seconds, Fetched: 3 row(s)
hive> SELECT name, book_title FROM authors;
OK
1      George Orwell 1984
J.R.R. Tolkien The Hobbit
Jane Austen  Pride and Prejudice
Time taken: 0.074 seconds, Fetched: 3 row(s)
hive> SELECT * FROM authors WHERE id > 2;
OK
3      Jane Austen  Pride and Prejudice
Time taken: 0.088 seconds, Fetched: 1 row(s)
hive> SELECT COUNT(*) FROM authors;
OK
3
Time taken: 0.079 seconds, Fetched: 1 row(s)
hive> SELECT name, COUNT(*) AS book_count
    > FROM authors
    > GROUP BY name;
Automatically selecting local only mode for query
Query ID = hdfs_20241008222442_c6bf7216-1925-41ac-acb-26e2c54a2983
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-10-08 22:24:44,281 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local14600044237_0002
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 1986 HDFS Write: 82032926 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
George Orwell 1
J.R.R. Tolkien 1
Jane Austen 1
Time taken: 1.325 seconds, Fetched: 3 row(s)
```

```

> set hive.exec.mode.local.auto=true;
hive> INSERT INTO TABLE authors VALUES
    >     (1, 'George Orwell', '1984'),
    >     (2, 'J.R.R. Tolkien', 'The Hobbit'),
    >     (3, 'Jane Austen', 'Pride and Prejudice');
Automatically selecting local only mode for query
Query ID = hdfs_20241008222350_0f993a20-aac2-4c9e-b9bf-d98b02f14e9e
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-10-08 22:23:52,056 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local2048767529_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://0.0.0.0:9000/user/hive/warehouse/books.db/authors/.hive-staging_hive_2024-10-08_22-23-50_625_1046627743533150526-1/-ext-10000
Loading data to table books.authors
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 0 HDFS Write: 82032504 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.697 seconds
hive> SELECT * FROM authors;
OK
1      George Orwell    1984
2      J.R.R. Tolkien   The Hobbit
3      Jane Austen      Pride and Prejudice
Time taken: 0.07 seconds, Fetched: 3 row(s)
hive>
```

```

[hdfs@e29e5b5fd20e /opt/hive/bin
> (2, 'J.R.R. Tolkien', 'The Hobbit'),
> (3, 'Jane Austen', 'Pride and Prejudice');
Query ID = hdfs_20241008222056_a22310ae-38c9-46b2-9a88-2040060c6591
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
File Input Format: org.apache.hadoop.mapred.TextInputFormat
File Output Format: org.apache.hadoop.mapred.TextOutputFormat
Kill Command = /opt/hadoop-3.1.1/bin/mapred job -kill job_1728422779698_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-10-08 22:21:06,904 Stage-1 map = 0%,  reduce = 0%
2024-10-08 22:21:15,098 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_1728422779698_0002 with errors
Error during job, obtaining debugging information...
Examining task ID: task_1728422779698_0002_m_000000 (and more) from job job_1728422779698_0002

Task with the most failures(4):
-----
Task ID:
task_1728422779698_0002_m_000000
-----
URL:
http://0.0.0.0:8088/taskdetails.jsp?jobid=job_1728422779698_0002&tid=task_1728422779698_0002_m_000000
-----
Diagnostic Messages for this Task:
Container launch failed for container_1728422779698_0002_01_000005 : org.apache.hadoop.yarn.exceptions.InvalidAuxServiceException: The auxService:mapreduce_shuffle does not exist
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
at org.apache.hadoop.yarn.api.records.impl.pb.SerializedExceptionPBImpl.instantiateExceptionImpl(SerializedExceptionPBImpl.java:171)
at org.apache.hadoop.yarn.api.records.impl.pb.SerializedExceptionPBImpl.instantiateException(SerializedExceptionPBImpl.java:182)
at org.apache.hadoop.yarn.api.records.impl.pb.SerializedExceptionPBImpl.deSerialize(SerializedExceptionPBImpl.java:186)
at org.apache.hadoop.mapreduce.v2.app.launcher.ContainerLauncherImpl$ContainerLaunch(ContainerLauncherImpl.java:163)
at org.apache.hadoop.mapreduce.v2.app.launcher.ContainerLauncherImpl$EventProcessor.run(ContainerLauncherImpl.java:394)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)

FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1  HDFS Read: 0  HDFS Write: 0  FAIL
Total MapReduce CPU Time Spent: 0 msec
hive> SELECT * FROM authors;
OK
Time taken: 0.187 seconds
```

Data Retrieval and Querying

We started by looking at all the information in the `authors` table using the command `SELECT * FROM authors`. This command shows us every piece of data in the table, which is helpful when you want to see everything at once.

Next, we ran a special command `SELECT * FROM authors WHERE id > 2` to get only the records of authors whose IDs are greater than 2. This shows how Hive can pick out specific data based on a rule or condition we set.

We also used another command `SELECT name, book_title FROM authors` to look at just the authors' names and their book titles. This is useful when we only need some parts of the data, not everything.

Data Aggregation and Grouping

For more complex tasks, we counted how many records are in the `authors` table using `SELECT COUNT(*) FROM authors`. This command helps us understand the size of our data quickly.

We did a more detailed count with `SELECT name, COUNT(*) AS book_count FROM authors GROUP BY name`. This sorts the data by each author's name and tells us how many books each author has. It's a way to organize and summarize data so it's easier to understand.

Error Handling and Resolution

While running these commands, we saw some error messages. These errors happened because of some setup issues, which is common when working with big systems like Hive. Talking about these errors in our work is important because it shows we know how to find and fix problems. This is a big part of working with data, as things often don't go right the first time.

WordCount Job Execution

We created the file.

```
cm hdfs@e29e5b5fd20e: ~
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>docker exec -it hadoop-hive-container /bin/bash
hdfs@e29e5b5fd20e:~$ ls
'WordCount$IntSumReducer.class'      WordCount.jar      derby.log      file02.txt      var
'WordCount$TokenizerMapper.class'    WordCount.java    docker        input.txt      wordcount.jar
WordCount.class                      classes          file01.txt    local_text_file.txt wordcount_classes
hdfs@e29e5b5fd20e:~$
```

The java file:

```
C:\Users\LENOVO\wordcount\WordCountJava - Notepad++
Fichier Édition Recherche Affichage Encodeage Langage Paramètres Outils Macro Exécution Modules d'extension Documents ?
+ X
WordCount.java
1 package org.myorg;
2
3 import java.io.IOException;
4 import java.util.StringTokenizer;
5
6 import org.apache.hadoop.conf.Configuration;
7 import org.apache.hadoop.fs.Path;
8 import org.apache.hadoop.io.Text;
9 import org.apache.hadoop.io.IntWritable;
10 import org.apache.hadoop.mapreduce.Job;
11 import org.apache.hadoop.mapreduce.Mapper;
12 import org.apache.hadoop.mapreduce.Reducer;
13 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
14 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
15
16 public class WordCount {
17     public static class TokenizerMapper
18         extends Mapper<Object, Text, Text, IntWritable> {
19             private final static IntWritable one = new IntWritable();
20             private Text word = new Text();
21
22             public void map(Object key, Text value, Context context
23                             ) throws IOException, InterruptedException {
24                 StringTokenizer itr = new StringTokenizer(value.toString());
25                 while (itr.hasMoreTokens()) {
26                     word.set(itr.nextToken());
27                     context.write(word, one);
28                 }
29             }
30         }
31
32     public static class IntSumReducer
33         extends Reducer<Text, IntWritable, Text, IntWritable> {
34         private IntWritable result = new IntWritable();
35
36         public void reduce(Text key, Iterable<IntWritable> values,
37                           Context context
38                           ) throws IOException, InterruptedException {
39             int sum = 0;
40             for (IntWritable val : values) {
41                 sum += val.get();
42             }
43         }
44     }
}
Java source file
Type here to search
length:2,347 lines: 64 In: 64 Col: 1 Pos: 2,348 Windows (CR/LF) UTF-8 INS
Windows 11 53°F 11:47 PM 10/8/2024
```

The mapreduce generated some errors due to issues of our directories.

References :

[**MapReduce Tutorial | MapReduce In Hadoop | MapReduce Example | MapReduce For Beginners | Simplilearn \(youtube.com\)**](#)

[**How to run Word Count example on Hadoop MapReduce \(WordCount Tutorial\) \(youtube.com\)**](#)

