

Bigdata - Map-Reduce và bài toán Word Count

Mô hình tính toán MapReduce và bài toán Wordcount

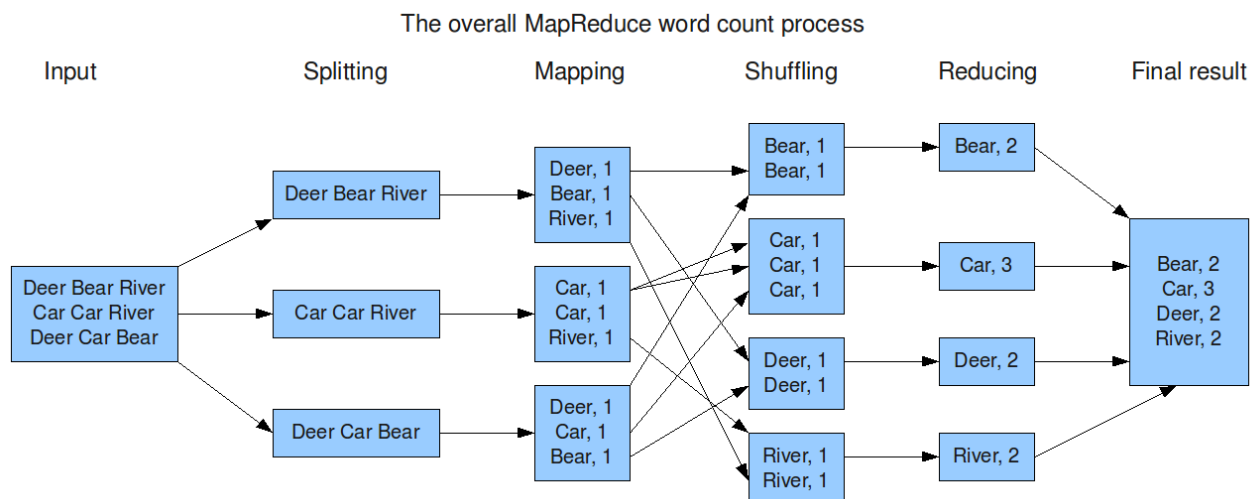
Ref: <https://blog.duyet.net/>

Bài toán word-count (đếm từ) là bài toán dễ hiểu nhất minh họa cho MapReduce (MR). Bài toán có những đặc điểm sau:

- File cần đếm rất lớn (quá lớn để có thể được tải lên bộ nhớ chính của 1 máy)
- Mỗi cặp <từ ngữ, số lượng> quá lớn cho bộ nhớ.

MapReduce chia làm 3 thao tác:

- Map: quét file đầu vào và ghi lại từng bản ghi
- Group by Key: sắp xếp và trộn dữ liệu cho mỗi bản ghi sinh ra từ Map
- Reduce: tổng hợp, thay đổi hay lọc dữ liệu từ thao tác trước và ghi kết quả ra File.



Về mặt định nghĩa thuật toán, ta có thể mô tả MR như sau:

- Input: dữ liệu dưới dạng Key \rightarrow Value
- Lập trình viên viết 2 thủ tục:
- $\text{Map}(k, v) \rightarrow \langle k', v' \rangle^*$
- $\text{Reduce}(k', \langle v' \rangle) \rightarrow \langle k', v'' \rangle$

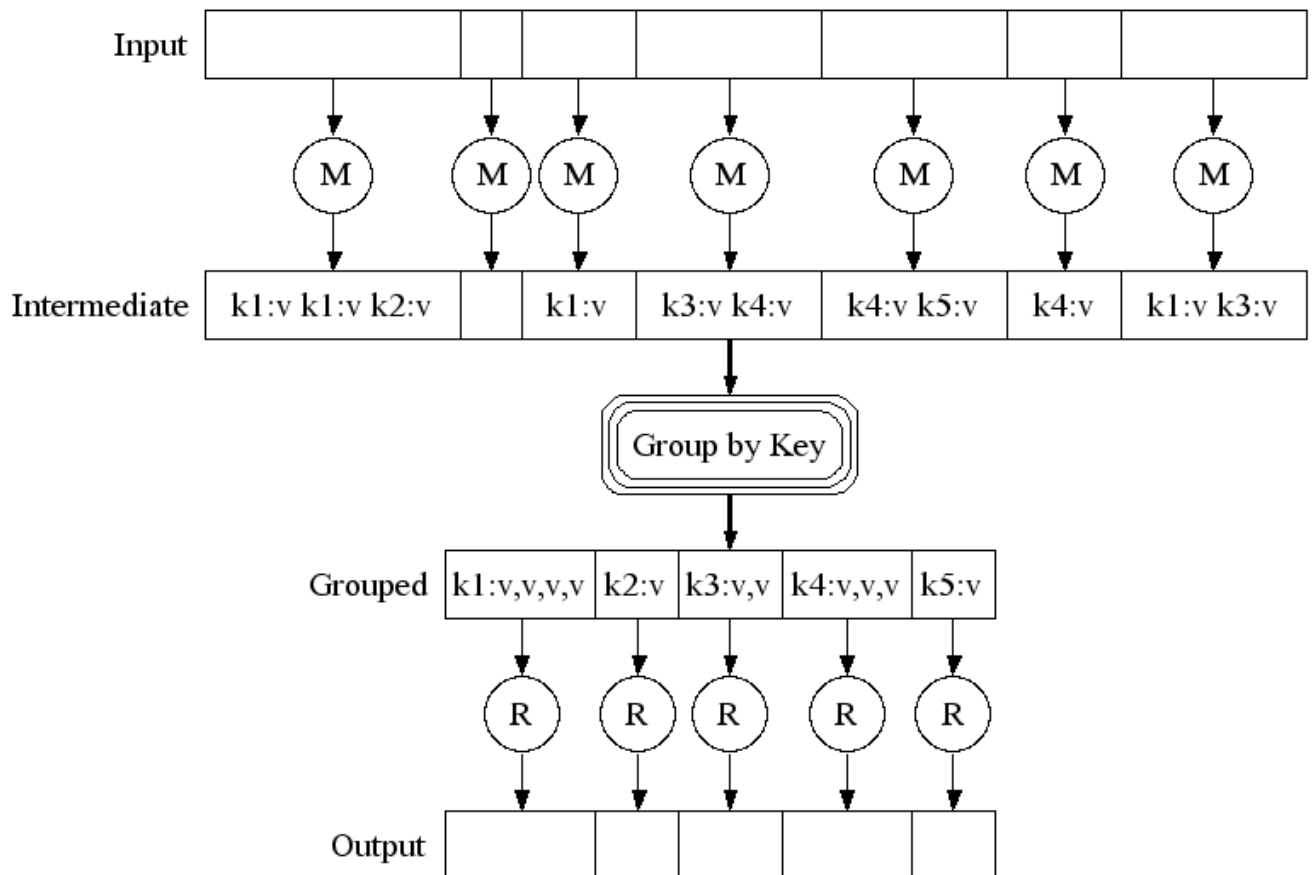
Với:

- Map biến mỗi key k thu được bằng thành cặp $\langle k', v' \rangle$.
- Reduce nhận đầu vào là khoá k' và danh sách các giá trị v' và trả về kết quả là cặp $\langle k', v'' \rangle$.

Ví dụ với hình mô tả ở trên thì Map trả về danh sách: $\langle \text{Bear}, 1 \rangle$, $\langle \text{Bear}, 1 \rangle$ còn Reduce nhận kết quả trên và trả về $\langle \text{Bear}, 2 \rangle$.

Lập lịch và dòng dữ liệu

Sau khi đã có cách đầu nối và phương pháp tính toán, vấn đề tiếp theo cần bàn là tính thế nào, khi nào và ra sao. Map-Reduce có một đặc điểm thú vị là chỉ cần phân chia các File thành các vùng đọc lập thì các thủ tục Map không hoàn toàn liên quan đến nhau có thể thực hiện song song.

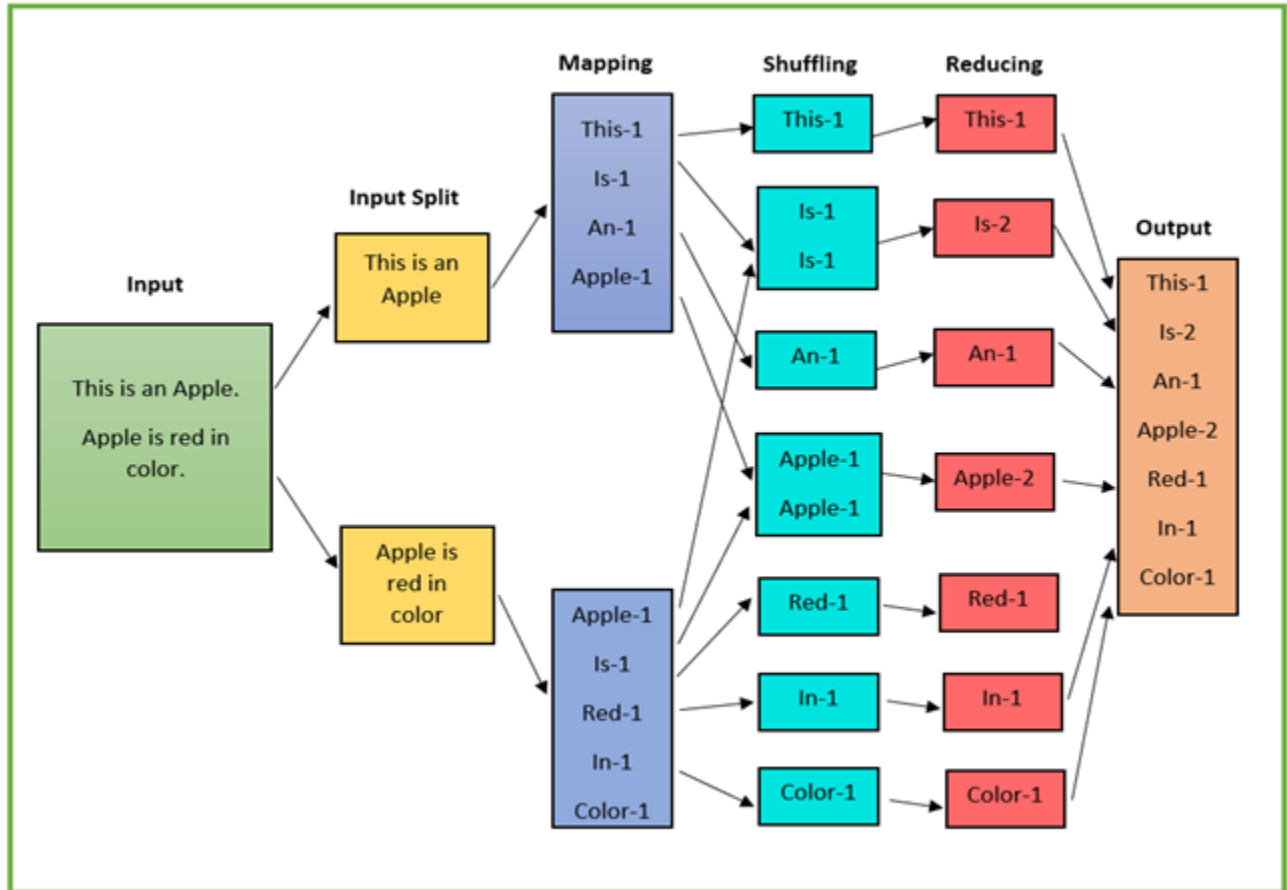


Một File Input có thể được xử lý bởi nhiều Map/Reduce. Map-Reduce sẽ cố gắng cung cấp giao diện lập trình đơn giản trong khi che dấu những xử lý phức tạp đi. Các xử lý chi tiết phức tạp bao gồm:

- Phân chia dữ liệu
- lập lịch chạy các thủ tục Map/Reduce trên các máy tính
- Thực hiện thủ tục Groupby
- Quản lý hỏng hóc (ví dụ tự động khởi động các thủ tục M/R đang chạy dở thì máy hỏng, quản lý dữ liệu khi máy hỏng)
- Quản lý giao tiếp giữa các máy tính.

Tutorial

Reference: medium.com



Công việc:

Bước 1:

Đầu tiên, chúng ta phải import dataset vào HDFS. Trong trường hợp này, dataset là 1 file text. Ví dụ, chúng ta có file **word_count.txt** có nội dung sau trong thư mục D:\Hadoop\samples

This is an Apple. Apple is red in color.	
---	--

Bước 2:

Tạo thư mục trên hệ thống file Hadoop (HDFS) sử dụng lệnh sau:

hdfs dfs -mkdir /wc

Bước 3:

Copy file word_count.txt vào hdfs bằng câu lệnh sau

hdfs dfs -put D:/Hadoop/samples/word_count.txt /wc
--

Bước 4:

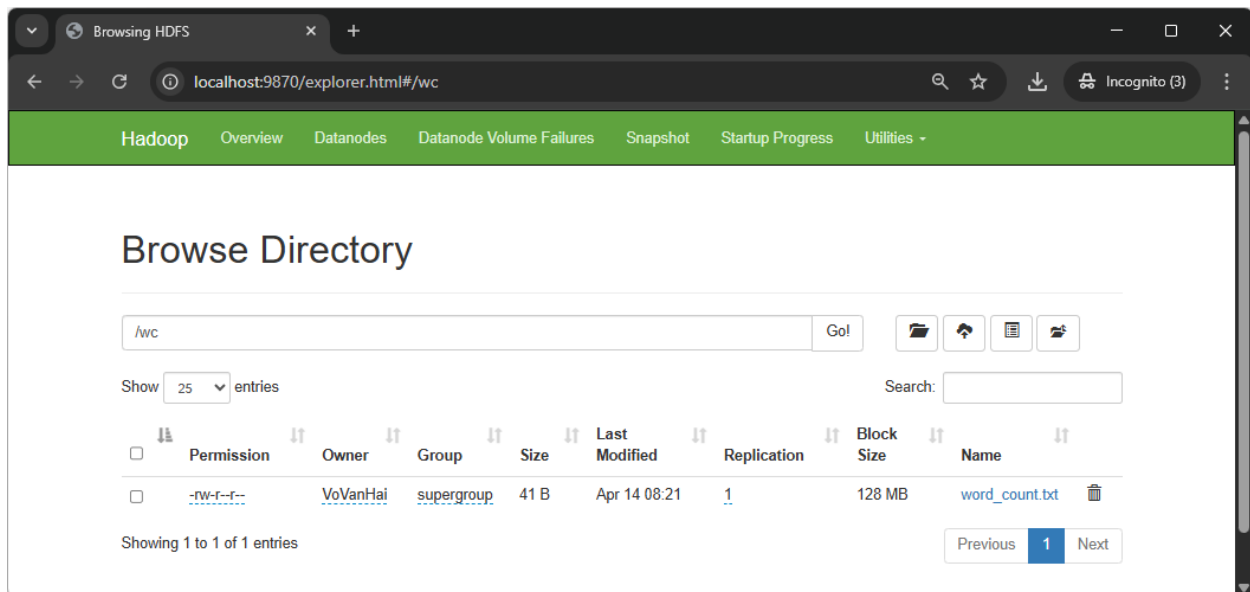
Kiểm tra liệu file đã copy đúng chỗ chưa. DỪng lệnh sau

```
hdfs dfs -ls /wc
```

Kết quả dạng như sau:

```
D:\Hadoop\samples>hdfs dfs -ls /wc
Found 1 items
-rw-r--r--  1 VoVanHai supergroup          41 2025-04-14 08:21 /wc/word_count.txt
```

Có thể vào link <http://localhost:9870/explorer.html#> để xem “Browse Directory”

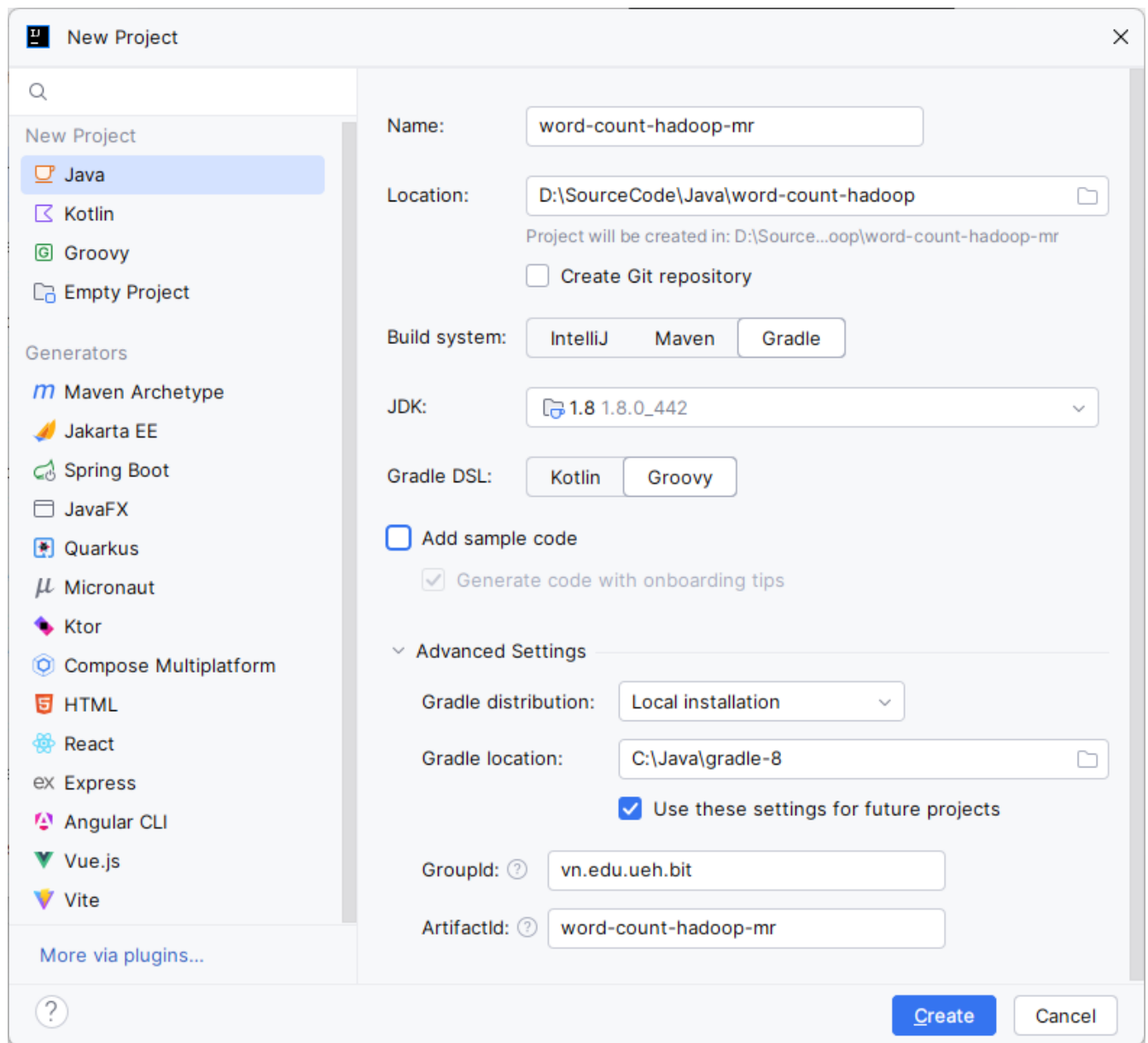


Bước 5:

Tiếp theo, ta chuẩn bị viết code cho ứng dụng. Có thể dùng bất cứ IDE nào. Trong ví dụ này, tôi dùng JetBrains IntelliJ IDEA for education.

Trong IntelliJ tạo project mới.

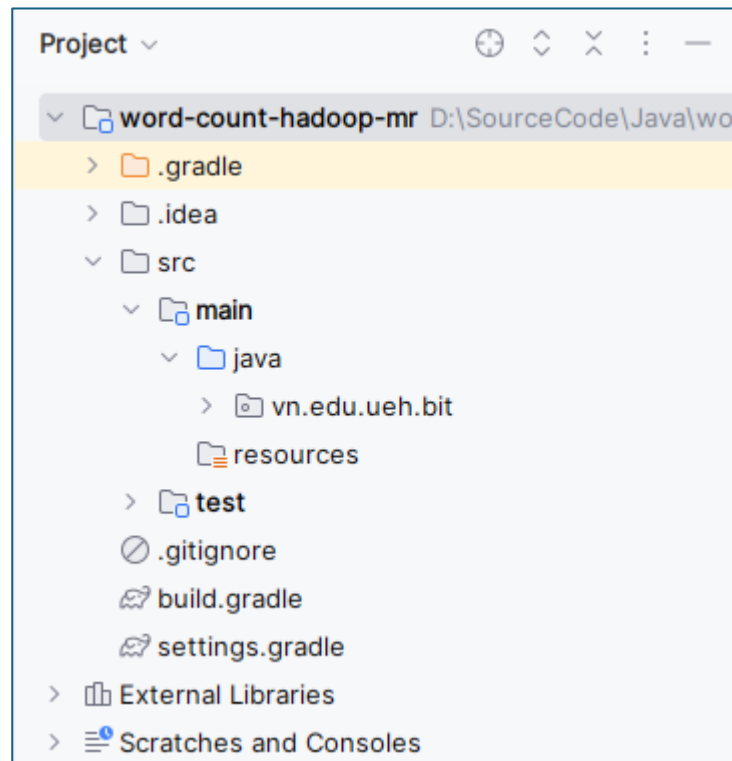
Lưu ý về việc chọn build tool (Gradle hoặc Maven)



Trong trường hợp này, tôi chọn Gradle và trong mục distribution chọn “Local instalation”. (Để làm được bước này, ta cần downalod gradle (<https://docs.gradle.org/current/userguide/installation.html>) rồi giải nén ra).

Nhấn nút create để tạo project.

Kết quả dạng giống như thế này. (có thể phải chờ cho việc tạo project hoàn tất)

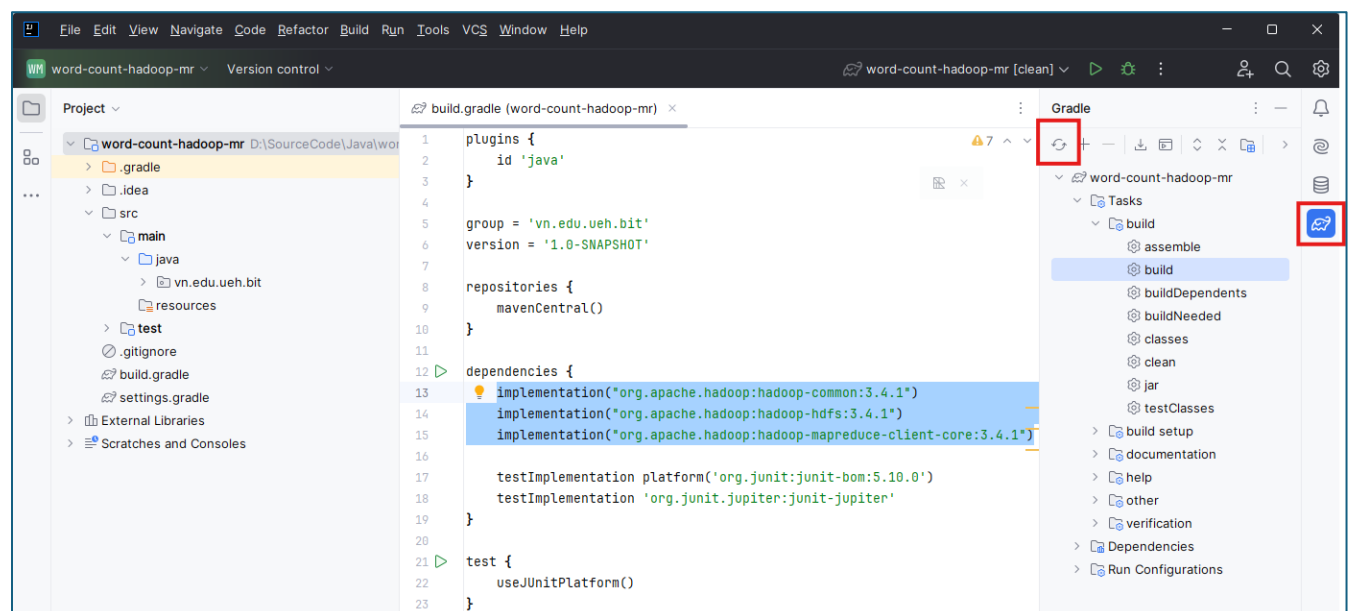


Nhấn mở file “build.gradle” sau đó thêm vào 3 dependencies sau

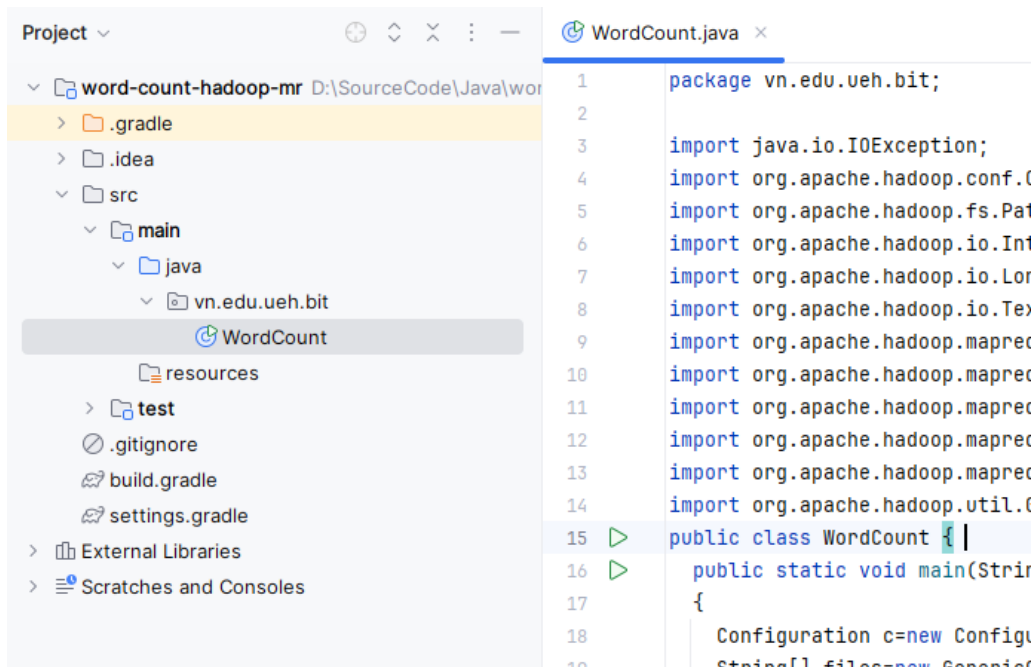
```
implementation("org.apache.hadoop:hadoop-common:3.4.1")  
implementation("org.apache.hadoop:hadoop-hdfs:3.4.1")  
implementation("org.apache.hadoop:hadoop-mapreduce-client-core:3.4.1")
```

```
build.gradle (word-count-hadoop-mr) x
1  plugins {
2      id 'java'
3  }
4
5  group = 'vn.edu.ueh.bit'
6  version = '1.0-SNAPSHOT'
7
8  repositories {
9      mavenCentral()
10 }
11
12 dependencies {
13     implementation("org.apache.hadoop:hadoop-common:3.4.1")
14     implementation("org.apache.hadoop:hadoop-hdfs:3.4.1")
15     implementation("org.apache.hadoop:hadoop-mapreduce-client-core:3.4.1")
16
17     testImplementation platform('org.junit:junit-bom:5.10.0')
18     testImplementation 'org.junit.jupiter:junit-jupiter'
19 }
20
21 test {
22     useJUnitPlatform()
23 }
```

Refresh lại Gradle, đợi cho quá trình build



Tạo 1 file mới trong gói vn.edu.ueh.bit với tên WordCount như hình



Copy và dán nội dung sau vào

```
package vn.edu.ueh.bit;

import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
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 void main(String [] args) throws Exception
    {
        Configuration c=new Configuration();
        String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
        Path input=new Path(files[0]);
        Path output=new Path(files[1]);
        Job j=new Job(c,"wordcount");
        j.setJarByClass(WordCount.class);
        j.setMapperClass(MapForWordCount.class);
        j.setReducerClass(ReduceForWordCount.class);
        j.setOutputKeyClass(Text.class);
        j.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(j, input);
        FileOutputFormat.setOutputPath(j, output);
        System.exit(j.waitForCompletion(true)?0:1);
    }
}
```



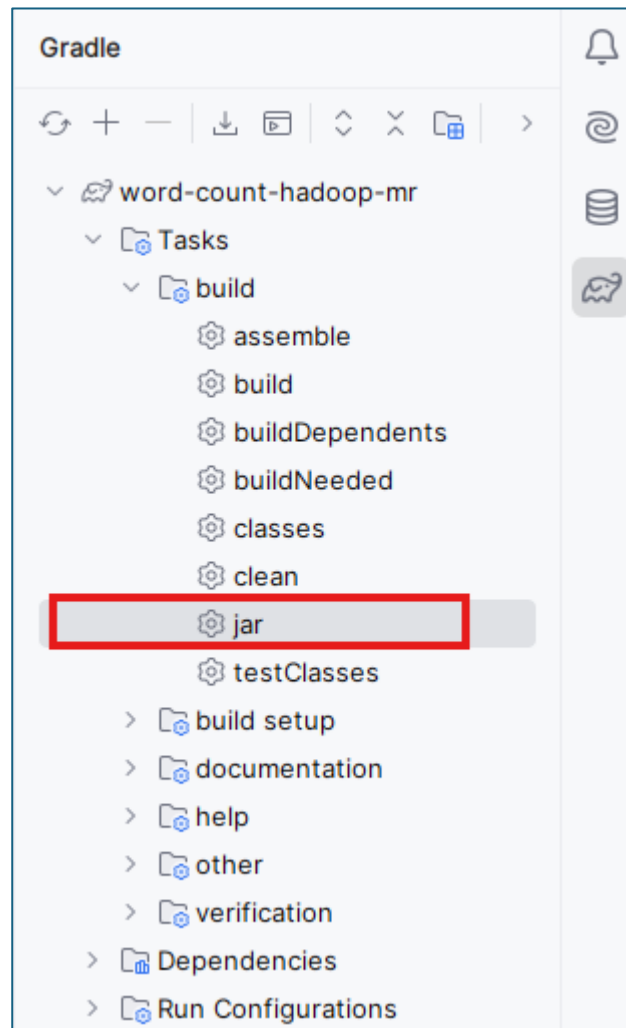
```

public static class MapForWordCount extends Mapper<LongWritable, Text, Text,
IntWritable>{
    public void map(LongWritable key, Text value, Context con) throws
IOException, InterruptedException
    {
        String line = value.toString();
        String[] words=line.split(" ");
        for(String word: words )
        {
            Text outputKey = new Text(word.toUpperCase().trim());
            IntWritable outputValue = new IntWritable(1);
            con.write(outputKey, outputValue);
        }
    }
}

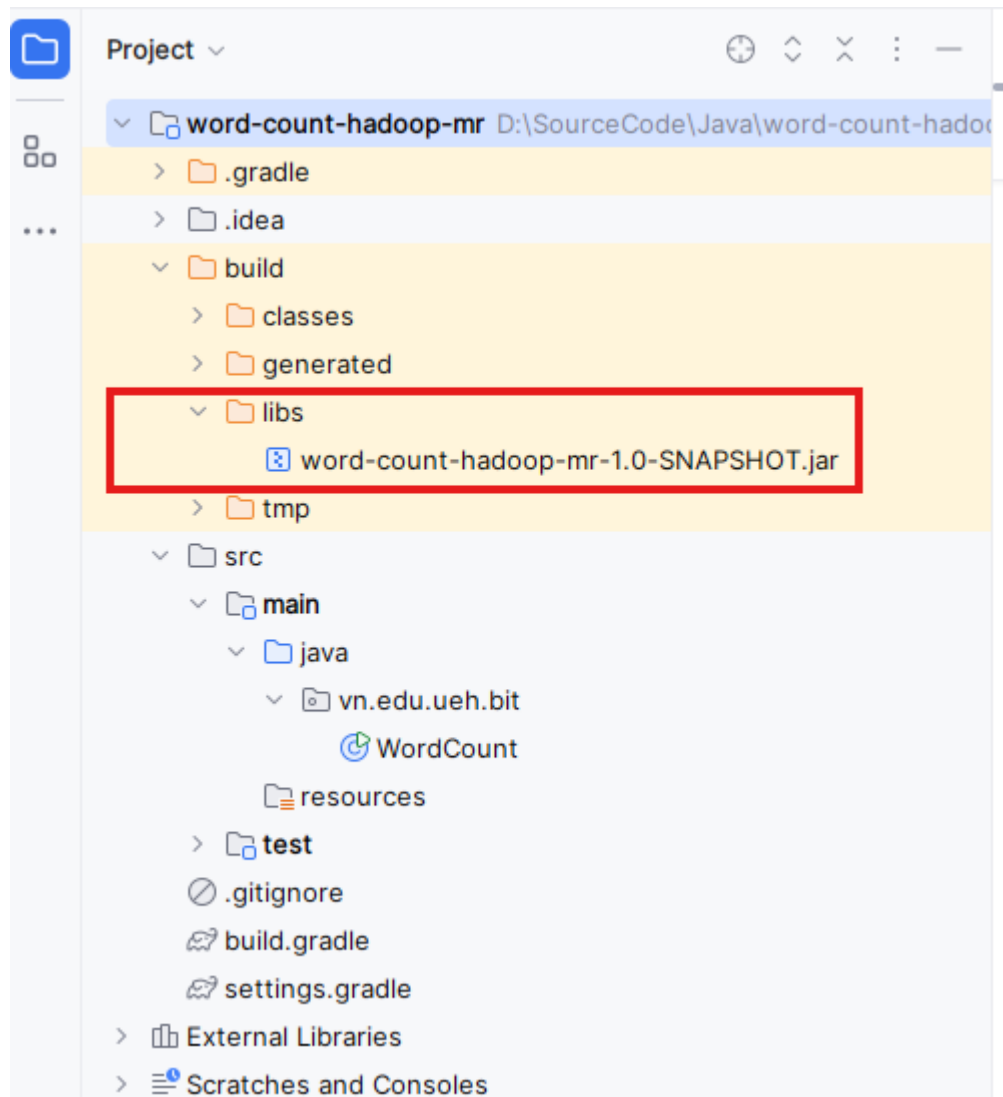
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text,
IntWritable>
{
    public void reduce(Text word, Iterable<IntWritable> values, Context con)
throws IOException, InterruptedException
    {
        int sum = 0;
        for(IntWritable value : values)
        {
            sum += value.get();
        }
        con.write(word, new IntWritable(sum));
    }
}

```

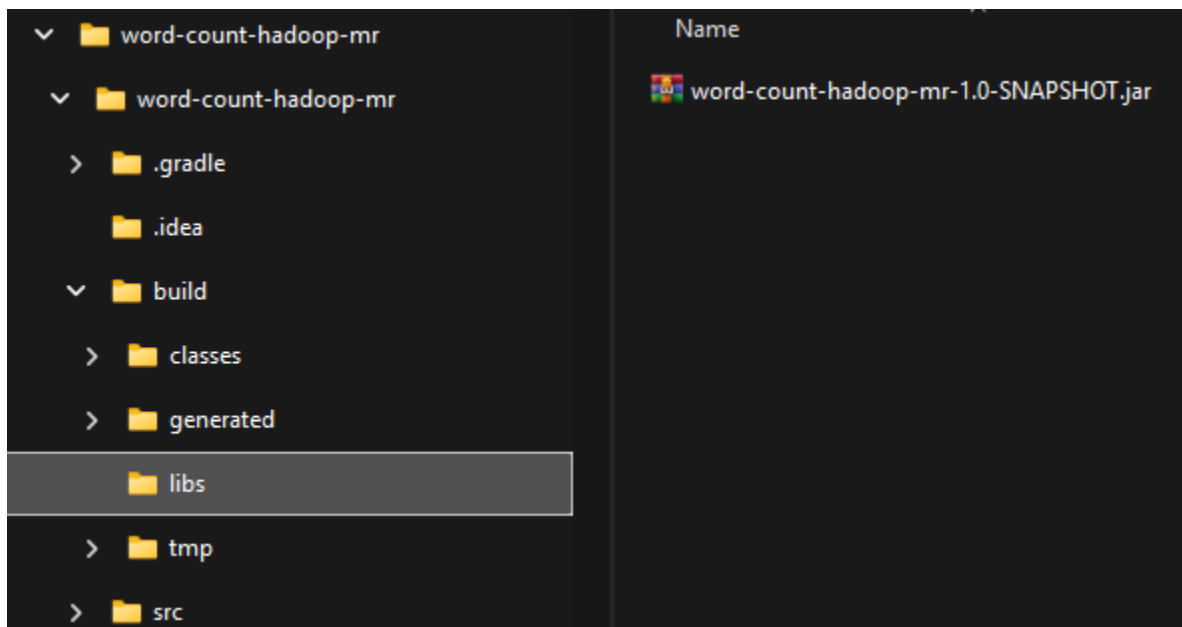
Đảm bảo không bị lỗi gì sau đó build file jar



Kết quả như sau



Nhấn chuột phải lên thư mục libs, chọn Open In → Explore. Windows Explorer sẽ mở và cho Kết quả như sau



Bước 6:

Chúng ta cần chạy file JAR với dữ liệu đầu vào của chúng ta sẽ là file txt của tập dữ liệu và đầu ra sẽ được lưu trữ trong một file khác. (chạy chương trình được lưu trong file JAR với dữ liệu đầu vào có trong HDFS.)

Cú pháp

```
hadoop jar <jar path> <class path> <input file path in hdfs> <output file path in hdfs>
```

Trường hợp này, cho dễ làm việc, copy file jar đã build được vào thư mục

```
hadoop jar word-count-hadoop-mr-1.0-SNAPSHOT.jar vn.edu.ueh.bit.WordCount /wc /output_dir
```

Kết quả sẽ có dạng như sau:

```
D:\Hadoop\samples>hadoop jar word-count-hadoop-mr-1.0-SNAPSHOT.jar vn.edu.ueh.bit.WordCount /wc /output_dir
2025-04-14 08:23:16,119 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2025-04-14 08:23:16,654 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/VoVanHai/.staging/job_1744593535775_0002
2025-04-14 08:23:16,852 INFO input.FileInputFormat: Total input files to process : 1
2025-04-14 08:23:17,333 INFO mapreduce.JobSubmitter: number of splits:1
2025-04-14 08:23:17,406 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744593535775_0002
2025-04-14 08:23:17,406 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-14 08:23:17,523 INFO conf.Configuration: resource-types.xml not found
2025-04-14 08:23:17,523 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-14 08:23:17,726 INFO impl.YarnClientImpl: Submitted application application_1744593535775_0002
2025-04-14 08:23:17,757 INFO mapreduce.Job: The url to track the job: http://VVH-Precision:8088/proxy/application_1744593535775_0002/
2025-04-14 08:23:17,758 INFO mapreduce.Job: Running job: job_1744593535775_0002
2025-04-14 08:23:24,897 INFO mapreduce.Job: Job job_1744593535775_0002 running in uber mode : false
2025-04-14 08:23:24,898 INFO mapreduce.Job: map 0% reduce 0%
2025-04-14 08:23:30,989 INFO mapreduce.Job: map 100% reduce 0%
2025-04-14 08:23:36,041 INFO mapreduce.Job: map 100% reduce 100%
2025-04-14 08:23:36,046 INFO mapreduce.Job: Job job_1744593535775_0002 completed successfully
2025-04-14 08:23:36,123 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=101
    FILE: Number of bytes written=554835
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=145
    HDFS: Number of bytes written=54
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=2053
    Total time spent by all reduces in occupied slots (ms)=2555
    Total time spent by all map tasks (ms)=2053
    Total time spent by all reduce tasks (ms)=2555
    Total vcore-milliseconds taken by all map tasks=2053
    Total vcore-milliseconds taken by all reduce tasks=2555
    Total megabyte-milliseconds taken by all map tasks=2102272
    Total megabyte-milliseconds taken by all reduce tasks=2616320
  Map-Reduce Framework
```

```

Map-Reduce Framework
  Map input records=2
  Map output records=9
  Map output bytes=77
  Map output materialized bytes=101
  Input split bytes=104
  Combine input records=0
  Combine output records=0
  Reduce input groups=8
  Reduce shuffle bytes=101
  Reduce input records=9
  Reduce output records=8
  Spilled Records=18
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=53
  CPU time spent (ms)=888
  Physical memory (bytes) snapshot=699961344
  Virtual memory (bytes) snapshot=1717534720
  Total committed heap usage (bytes)=1265631232
  Peak Map Physical memory (bytes)=346497024
  Peak Map Virtual memory (bytes)=855384064
  Peak Reduce Physical memory (bytes)=353464320
  Peak Reduce Virtual memory (bytes)=862187520
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=41
File Output Format Counters
  Bytes Written=54

```

Chạy lệnh sau để xem output

```
hdfs dfs -cat /output_dir/*
```

Kết quả như sau

```

D:\Hadoop\samples>hdfs dfs -cat /output_dir/*
AN      1
APPLE   1
APPLE.  1
COLOR.  1
IN      1
IS      2
RED     1
THIS    1


```

Truy xuất <http://localhost:8088/cluster> để xem thêm thông tin

All Applications

localhost:8088/cluster

Incognito (2)



All Appli

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
1	0	0	1	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
application_1744563535775_0002	VoVanHai	wordcount	MAPREDUCE		default	0	Mon Apr 14 08:23:17 +0700 2025	Mon Apr 14 08:23:18 +0700 2025	Mon Apr 14 08:23:34 +0700 2025	FINISHED

Showing 1 to 1 of 1 entries

Course tham khảo

<https://stg-tud.github.io/ctbd/2017/>

<https://juheck.gitbooks.io/hadoop-and-big-data/content/hdfs-mapreduce/exercise-hdfs.html>

<https://vda-lab.github.io/2016/04/hadoop-tutorial>