



ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP HCM  
KHOA CÔNG NGHỆ THÔNG TIN

LỚP 18CTT2  
NHẬP MÔN DỮ LIỆU LỚN

# BÁO CÁO

## LAB 02

**BÀI TOÁN ĐẾM TỪ VỚI  
MAPREDUCE**

# Mục lục

1. Thông tin thành viên nhóm: .....	3
2. Phân công công việc: .....	3
3. Nội dung thực hiện: .....	4
3.1. Mức 1: .....	4
a) Viết chương trình Java: .....	4
b) Thực thi các chương trình trên một file test dung lượng nhỏ .....	11
c) Thực thi các chương trình trên các file dung lượng lớn: .....	26
3.2. Mức 2: .....	27
a) Viết chương trình Python: .....	27
b) Chạy trên máy local .....	29
c) Chạy trên Hdfs .....	30
d) Bài tập áp dụng .....	31
4. Tài liệu tham khảo: .....	33

## 1. Thông tin thành viên nhóm:

STT	MSSV	Họ và tên
1	18120023	Nguyễn Huy Hải
2	18120058	Phạm Công Minh
3	18120533	Dương Đoàn Bảo Sơn
4	18120543	Trần Đại Tài

## 2. Phân công công việc:

Họ tên	Công việc	Tự đánh giá
Nguyễn Huy Hải	- Nguyên cứu version 1,2 mức 2 - Quay video thực thi code mức 2	100%
Phạm Công Minh	- Quay video thực thi code mức 1 - Viết báo cáo	100%
Dương Đoàn Bảo Sơn	- Nguyên cứu code mức 1 - Quay video giải thích code mức 1	100%
Trần Đại Tài	- Nguyên cứu version 3 mức 2 - Quay video giải thích code mức 2	100%

### 3. Nội dung thực hiện:

#### 3.1. Mức 1:

##### a) Viết chương trình Java:

**Phiên bản 1.0: chương trình thống kê số lượng của mỗi từ ở mức cơ bản (không tiền xử lý)**

- Sử dụng regex để trích xuất từ trong cuối văn bản

```
import java.util.regex.Pattern;
```

- Các class cấu hình và chạy mapreduce của Hadoop

```
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
```

- Các class dùng để tạo và tùy chỉnh Job và các tiến trình Map và Reduce bằng cách mở rộng các class tương ứng

```
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
```

- Các class giúp truy cập các files trong HDFS, có vai trò của InputFormat và OutputFormat trong MapReduce

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

- Các class tương ứng với các kiểu dữ liệu hỗ trợ cho việc đọc, ghi, so sánh các giá trị trong các tiền trình Map và Reduce

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
```

- Class giúp thông báo lỗi hoặc thành công trong Mapper và Reducer

```
import org.apache.log4j.Logger;
```

- Khởi tạo và cấu hình class WordCount

```
public class WordCount extends Configured implements Tool {
    //khởi tạo logger
    private static final Logger LOG = Logger.getLogger(WordCount.class);
    //tạo và chạy wordcount mới với tham số truyền vào là arguments ở command line
    public static void main(String[] args) throws Exception {
        int res = ToolRunner.run(new WordCount(), args);
        System.exit(res); //trả về System số nguyên res khi chương trình hoàn thành
    }
    //khởi tạo và chạy Job
    public int run(String[] args) throws Exception {
        Job job = Job.getInstance(getConf(), "wordcount");
        job.setJarByClass(this.getClass()); // sử dụng file jar
        //lấy các đường dẫn của input và output
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //đặt lớp map và reduce cho job
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        //xuất ra dưới dạng key value
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        return job.waitForCompletion(true) ? 0 : 1;
    }
}
```

- Mở rộng class Map: tách các từ bằng cách dùng regex và tạo ra cặp key-value theo dạng (word,1)

```
public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    private long numRecords = 0;
    //split các từ có khoảng trắng ở giữa, word boundary này có thể tách được spaces,
    tab và chấm cuối dòng.
```

```
//\b ranh giới của 1 từ, \s là ký tự trắng (xuống dòng,), \s* 0 hoặc nhiều \s
private static final Pattern WORD_BOUNDARY = Pattern.compile("\\s*\\b\\s*");

public void map(LongWritable offset, Text lineText, Context context)
    throws IOException, InterruptedException {
    String line = lineText.toString(); //input split là 1 dòng
    Text currentWord = new Text(); //từ (key)
    for (String word : WORD_BOUNDARY.split(line)) { //tách các từ
        if (word.isEmpty()) {
            continue;
        }
        currentWord = new Text(word);
        context.write(currentWord, one); //được ghi lại dưới dạng(key : 1)
    }
}
```

- Mở rộng class Map: tính tổng các value của mỗi key

```
public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    //chạy trên từng cặp key-value
    public void reduce(Text word, Iterable<IntWritable> counts, Context context)
        throws IOException, InterruptedException {
        int sum = 0;
        //tính tổng các value của key
        for (IntWritable count : counts) {
            sum += count.get();
        }
        //ghi cặp key-value mới vào context object
        context.write(word, new IntWritable(sum));
    }
}
```

**Phiên bản 2.0: thực hiện như phiên bản 1.0 nhưng lúc này không đếm các ký tự không phải là từ, không phân biệt hoa thường. Sử dụng Code ở phiên bản 1.0 với các thay đổi dưới đây:**

**\* Được xem là từ nếu chỉ chứa a-z, A-Z, 0-9**

- Class giúp truy cập các arguments trên command line tại run time

```
import org.apache.hadoop.conf.Configuration;
```

- Class giúp xử lý một phần input file thay vì toàn bộ

```
import org.apache.hadoop.mapreduce.lib.input.FileSplit;
```

- Class dùng để thao tác với chuỗi trong Hadoop

```
import org.apache.hadoop.util.StringUtils;
```

- Thay đổi cấu hình Job so với phiên bản 1: thêm Combiner ở giữa Mapper và Reducer

```
public int run(String[] args) throws Exception {
    Job job = Job.getInstance(getConf(), "wordcount");
    job.setJarByClass(this.getClass());
    // Use TextInputFormat, the default unless job.setInputFormatClass is used
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(Map.class);
    //thêm lớp combiner: lớp này có tác dụng xử lý thông tin trước khi chuyển qua lớp
    //reduce trên từng input split. Nhằm tiết kiệm băng thông và thời gian
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    return job.waitForCompletion(true) ? 0 : 1;
}
```

- Mở rộng class Map:
- + Cũng dùng regex để phân tách từ, tuy nhiên ở đây chỉ chọn các ký tự 0-9, a-z, A-Z để tạo thành từ
- + Cho người dùng thay đổi việc có phân biệt hoa thường hay không bằng cách thêm một đối số caseSensitive ở cmd line

```

public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    //thêm biến caseSensitive vào class Map
    private boolean caseSensitive = false;
    // đổi split thành \W+ :regex ngăn cách bởi các kí tự non-word 1 hoặc nhiều lần
    private static final Pattern WORD_BOUNDARY = Pattern.compile("\\"W+");
    
    protected void setup(Mapper.Context context)
        throws IOException,
               InterruptedException {
        Configuration config = context.getConfiguration();
        //có thể thay đổi giá trị caseSensitive ở cmd line , mặc định là false
        this.caseSensitive = config.getBoolean("wordcount.case.sensitive", false);
    }

    public void map(LongWritable offset, Text lineText, Context context)
        throws IOException, InterruptedException {
        String line = lineText.toString();
        // caseSensitive=false toàn bộ input split thành chữ thường
        if (!caseSensitive) {
            line = line.toLowerCase();
        }
        Text currentWord = new Text();
        for (String word : WORD_BOUNDARY.split(line)) {
            if (word.isEmpty()) {
                continue;
            }
            currentWord = new Text(word);
            context.write(currentWord, one);
        }
    }
}

```

**Phiên bản 3.0: phát triển tiếp từ phiên bản 2.0 và bổ sung việc không đếm các từ nằm trong danh sách stop\_words.txt. Sử dụng Code ở phiên bản 2.0 với các thay đổi dưới đây:**

- Thêm các class giúp xử lý trên file

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.net.URI;
import java.util.HashSet;
import java.util.Set;
```

- Đoạn code bên dưới cho phép thêm vào file stop-words khi gọi lệnh chạy code, trong lệnh thêm -skip “đường dẫn đến file stop-word”. File stop-word sẽ được đưa vào cách và sẽ có thông báo như ở dòng code LOG.info khi đã được đưa vào cache

```
public static void main(String[] args) throws Exception {
    int res = ToolRunner.run(new WordCount(), args);
    System.exit(res);
}

public int run(String[] args) throws Exception {
    Job job = Job.getInstance(getConf(), "wordcount");
    //thêm đoạn code trong phần run nếu trong commandline
    //có truyền -skip thì lấy URI của path phía sau -skip để định vị file stop-word
    //chúng ta đã đưa lên hdfs lừa vào trong cache.
    for (int i = 0; i < args.length; i += 1) {
        if ("-skip".equals(args[i])) {
            job.getConfiguration().setBoolean("wordcount.skip.patterns", true);
            i += 1;
            job.addCacheFile(new Path(args[i]).toUri());
            // this demonstrates logging
            LOG.info("Added file to the distributed cache: " + args[i]);
        }
    }
    job.setJarByClass(this.getClass());
    // Use TextInputFormat, the default unless job.setInputFormatClass is used
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(Map.class);
    job.setCombinerClass(Reduce.class);
    job.setReducerClass(Reduce.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    return job.waitForCompletion(true) ? 0 : 1;
}
```

- Mở rộng class Map:

+ Thêm vào biến patternsToSkip để lưu các stop-word

```
public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    private boolean caseSensitive = false;
    private long numRecords = 0;
    private String input;
    private Set<String> patternsToSkip = new HashSet<String>(); //mảng các từ, dấu câu cần bỏ qua
    private static final Pattern WORD_BOUNDARY = Pattern.compile("\\\\w+");
}
```

+ Chuyển input split thành dạng string, nếu có truyền tham số -skip thì gọi parseSkipFile

```
protected void setup(Mapper.Context context)
    throws IOException,
    InterruptedException {
    //chuyển các input split thành string để xử lý,
    if (context.getInputSplit() instanceof FileSplit) {
        this.input = ((FileSplit) context.getInputSplit()).getPath().toString();
    } else {
        this.input = context.getInputSplit().toString();
    }
    Configuration config = context.getConfiguration();
    this.caseSensitive = config.getBoolean("wordcount.case.sensitive", false);
    //nếu wordcount.skip.pattern = true thì gọi phương thức parseSkipFile với tham số là URI được lưu trên cache
    if (config.getBoolean("wordcount.skip.patterns", false)) {
        URI[] localPaths = context.getCacheFiles();
        parseSkipFile(localPaths[0]);
    }
}
```

+ Thêm các từ đầu câu trong file stop-word từ URI của nó trong cache vào mảng patternsToSkip

```
private void parseSkipFile(URI patternsURI) {
    LOG.info("Added file to the distributed cache: " + patternsURI);
    try {
        BufferedReader fis = new BufferedReader(new FileReader(new
File(patternsURI.getPath()).getName()));
        String pattern;
        while ((pattern = fis.readLine()) != null) {
            patternsToSkip.add(pattern);
        }
    } catch (IOException ioe) {
```

```

        System.err.println("Caught exception while parsing the cached file "
            + patternsURI + " : " + StringUtils.stringifyException(ioe));
    }
}

```

+ Ở class Map chỉ đơn giản thêm vào dòng code bỏ qua các từ trong mảng patternsToSkip

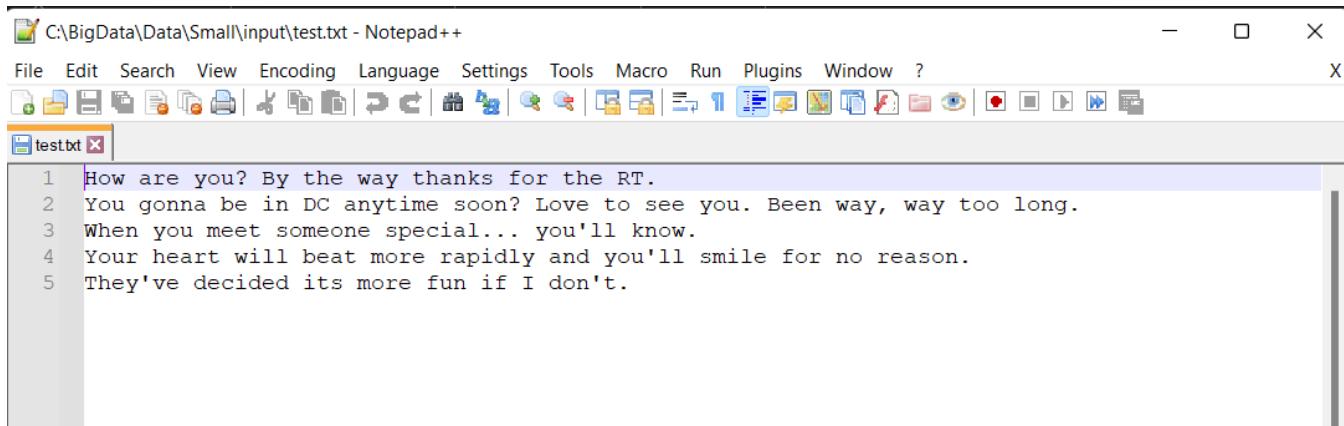
```

public void map(LongWritable offset, Text lineText, Context context)
    throws IOException, InterruptedException {
    String line = lineText.toString();
    if (!caseSensitive) {
        line = line.toLowerCase();
    }
    Text currentWord = new Text();
    for (String word : WORD_BOUNDARY.split(line)) {
        //thay đổi câu lệnh if trong map nếu các từ nằm trong patternsToSkip thì bỏ qua
        if (word.isEmpty() || patternsToSkip.contains(word)) {
            continue;
        }
        currentWord = new Text(word);
        context.write(currentWord, one);
    }
}
}

```

### b) Thực thi các chương trình trên một file test dung lượng nhỏ

\* Nội dung file test.txt:

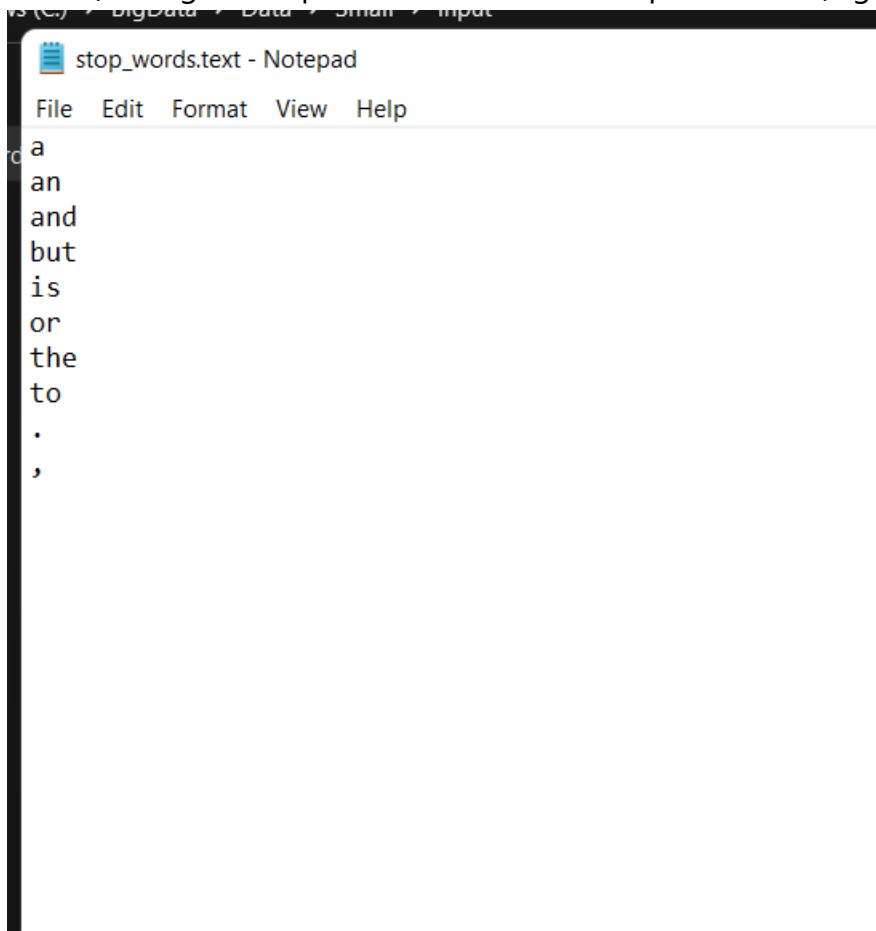


```

C:\BigData\Data\Small\input\test.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
test.txt [x]
1 How are you? By the way thanks for the RT.
2 You gonna be in DC anytime soon? Love to see you. Been way, way too long.
3 When you meet someone special... you'll know.
4 Your heart will beat more rapidly and you'll smile for no reason.
5 They've decided its more fun if I don't.

```

\* Nội dung file stop-word.text chứa các stop word sử dụng trong phiên bản word count 3.0



The screenshot shows a Notepad window titled "stop\_words.txt - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main content area contains the following text:

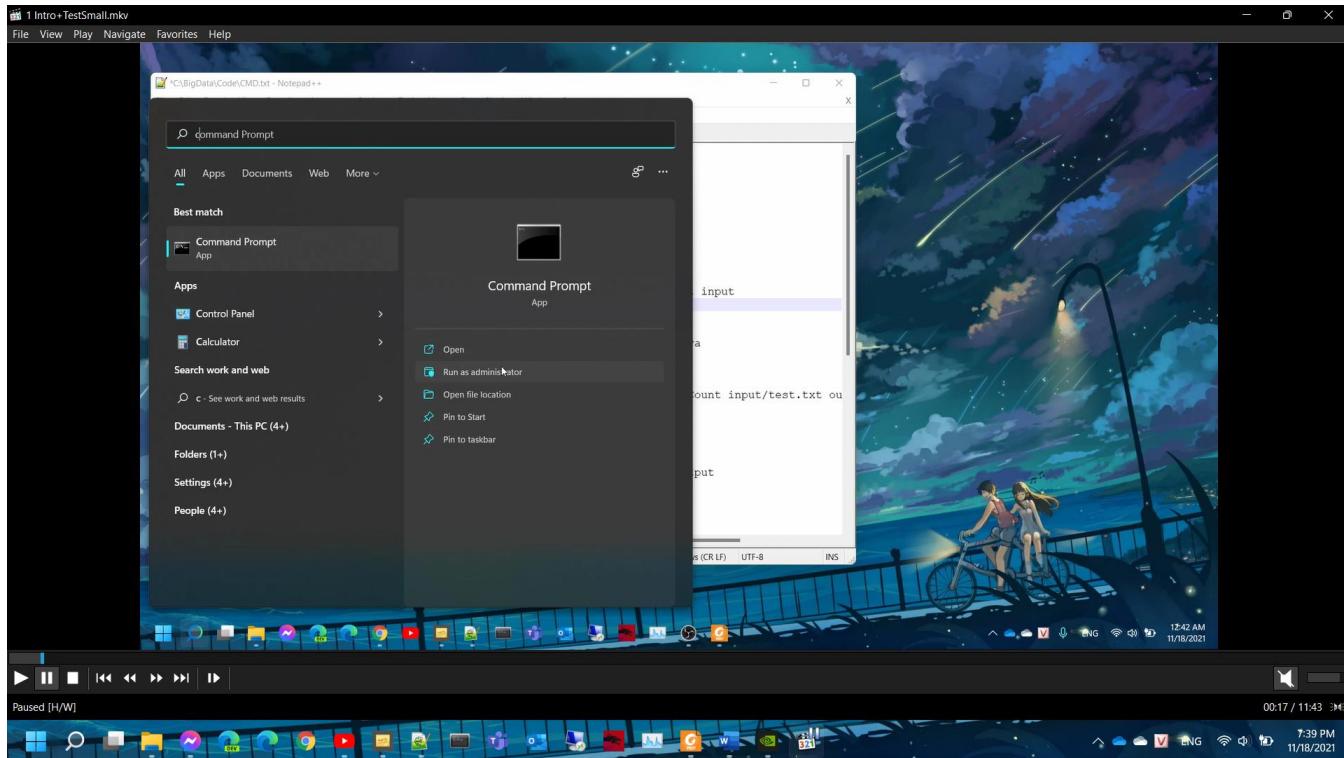
```
a  
an  
and  
but  
is  
or  
the  
to  
. .  
,
```

\* 2 file test.txt và stop-word.text được đặt trong thư mục Data\Level1\Test trong thư mục bài nộp

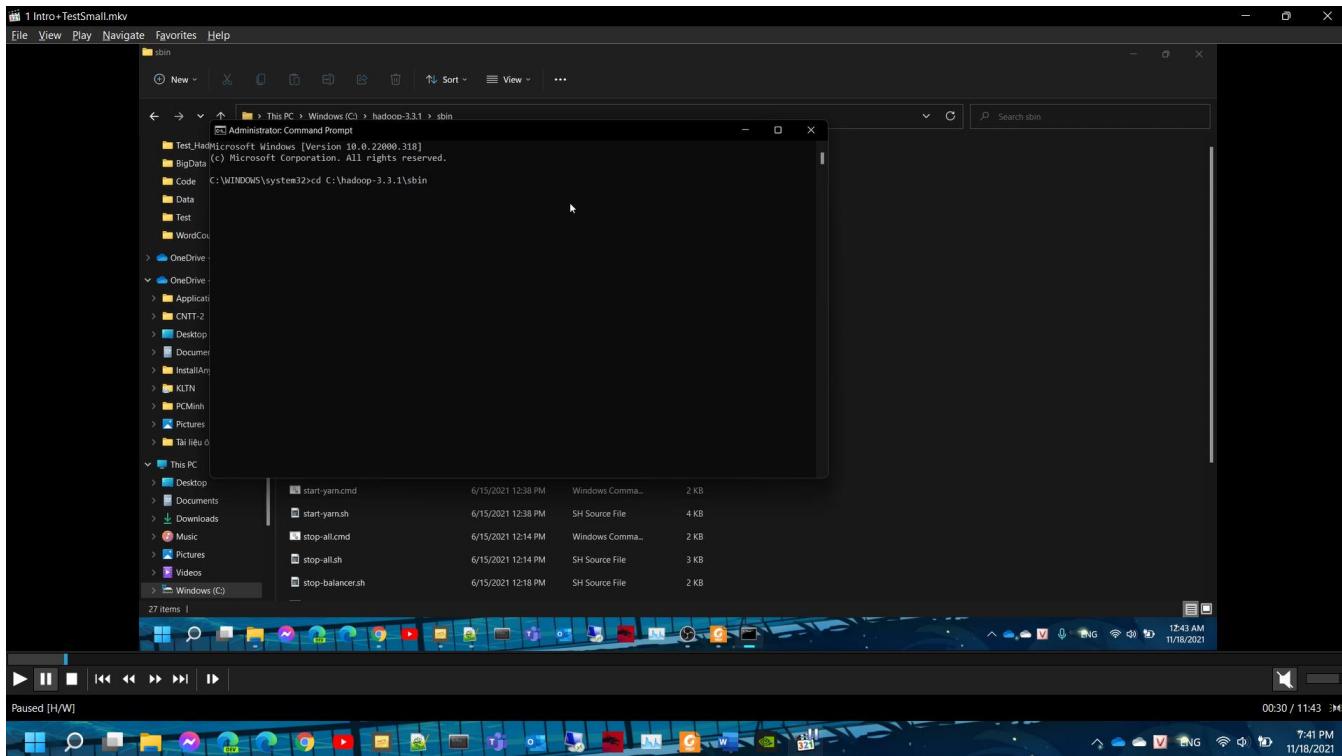
\* Các bước chạy các chương trình wordcount MapReduce với file này:

- **Bước 1:** Trước tiên khởi động Hadoop

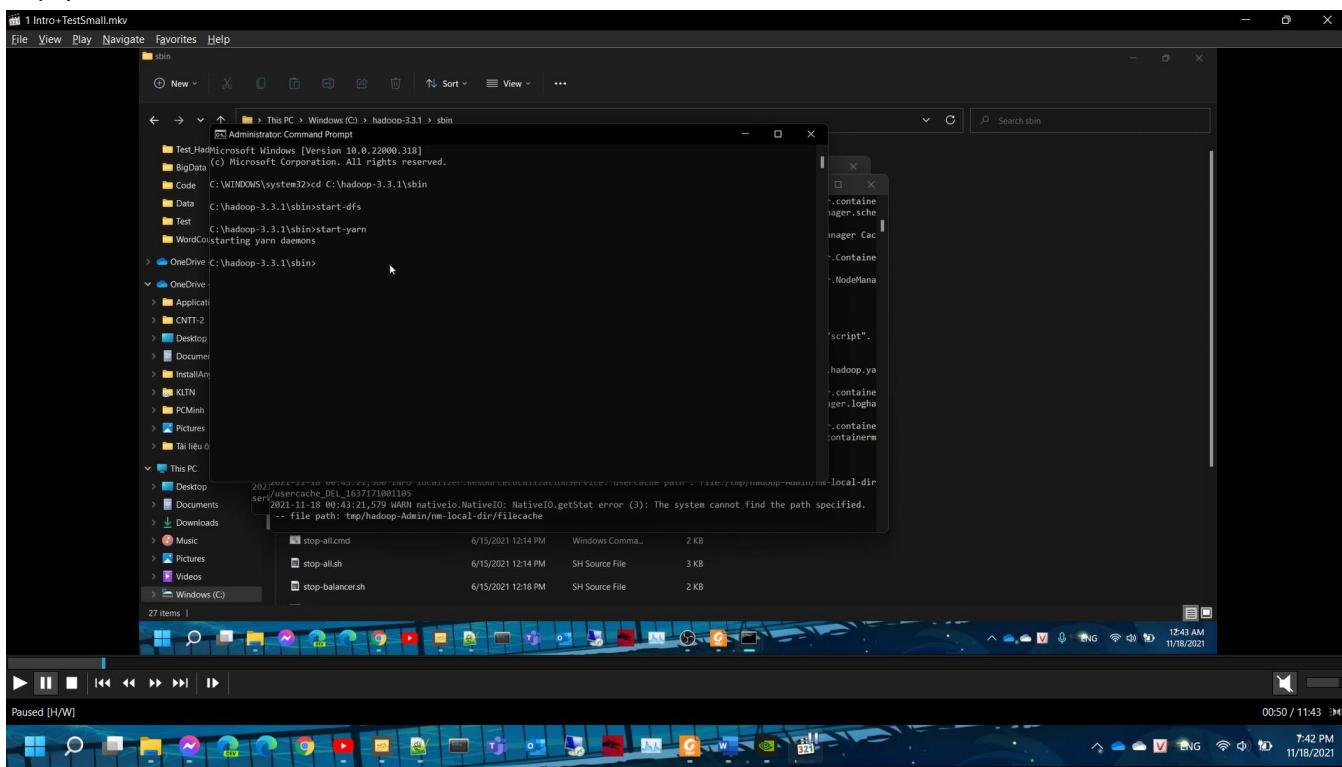
+ Mở Window Command Prompt và Run as administrator:



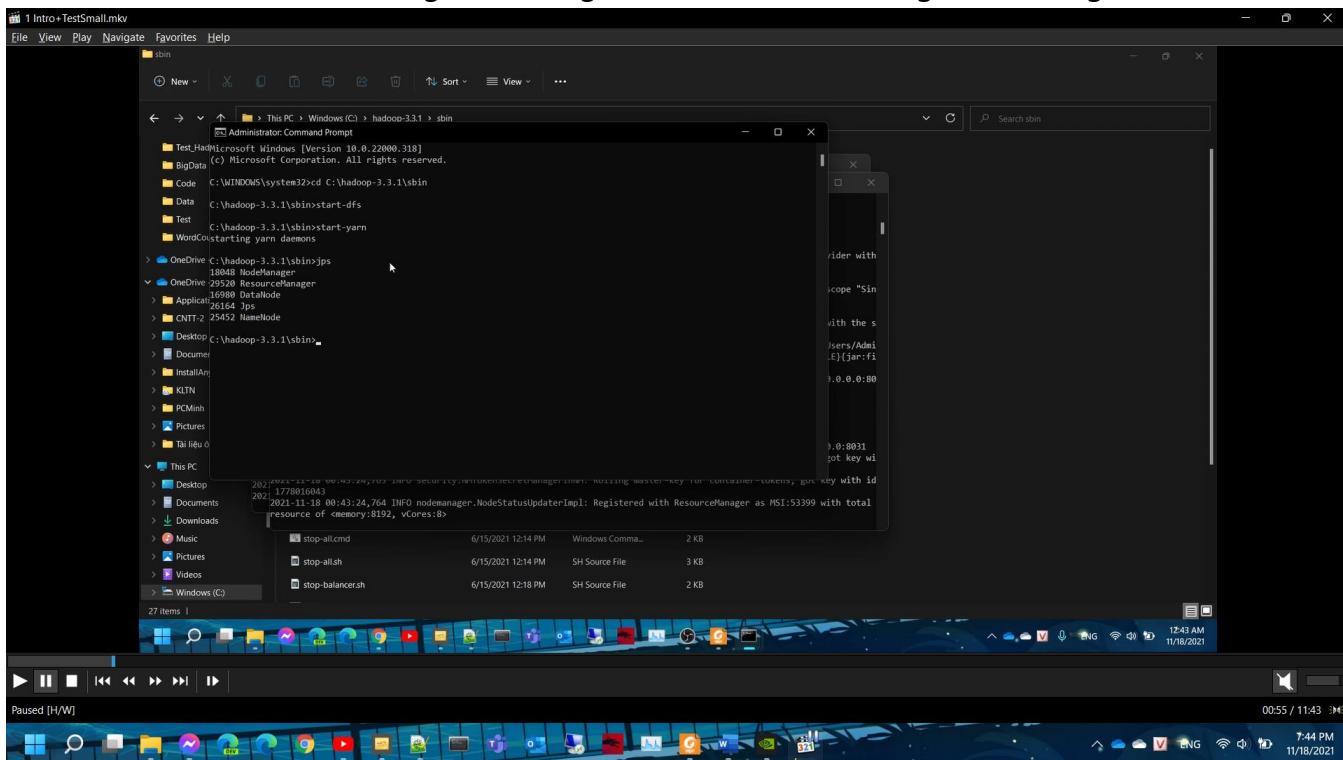
+Đi đến file sbin của Hadoop:



+ Chạy 2 lệnh start-dfs và start-yarn để khởi động các daemon (sẽ có 4 hộp thoại cmd xuất hiện)

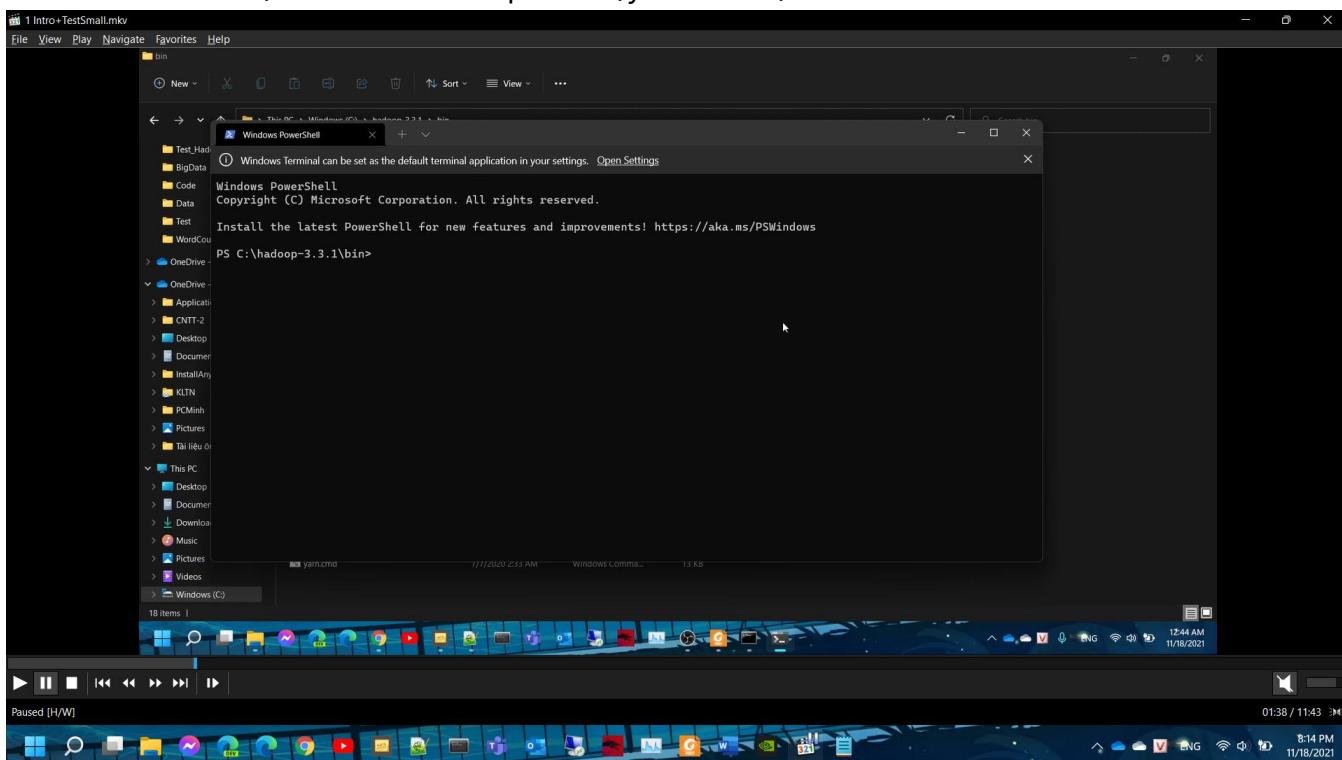


+ Kiểm tra các daemon đang hoạt động để biết là đã khởi động thành công

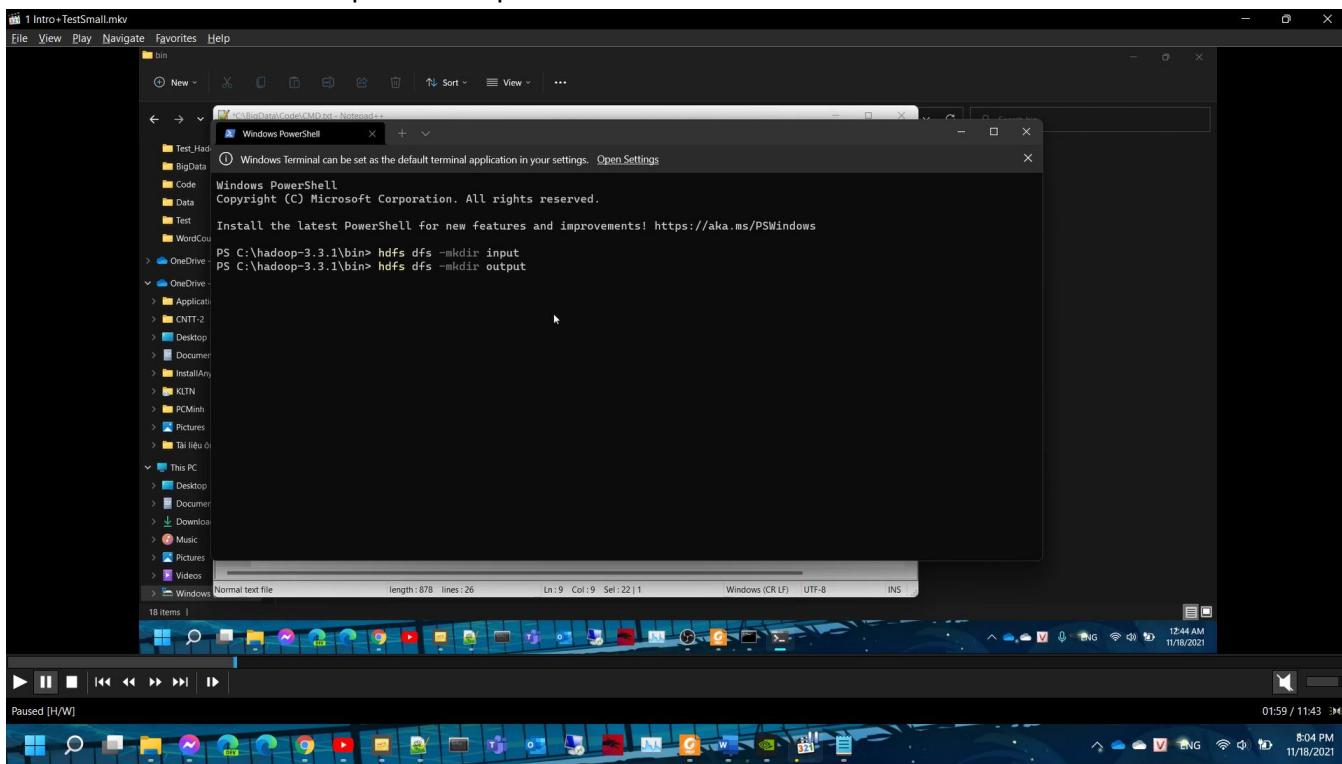


- **Bước 2:** Đưa các file input lên HDFS gồm file test.txt và file stop-words.text để chạy wordcount phiên bản 3.0

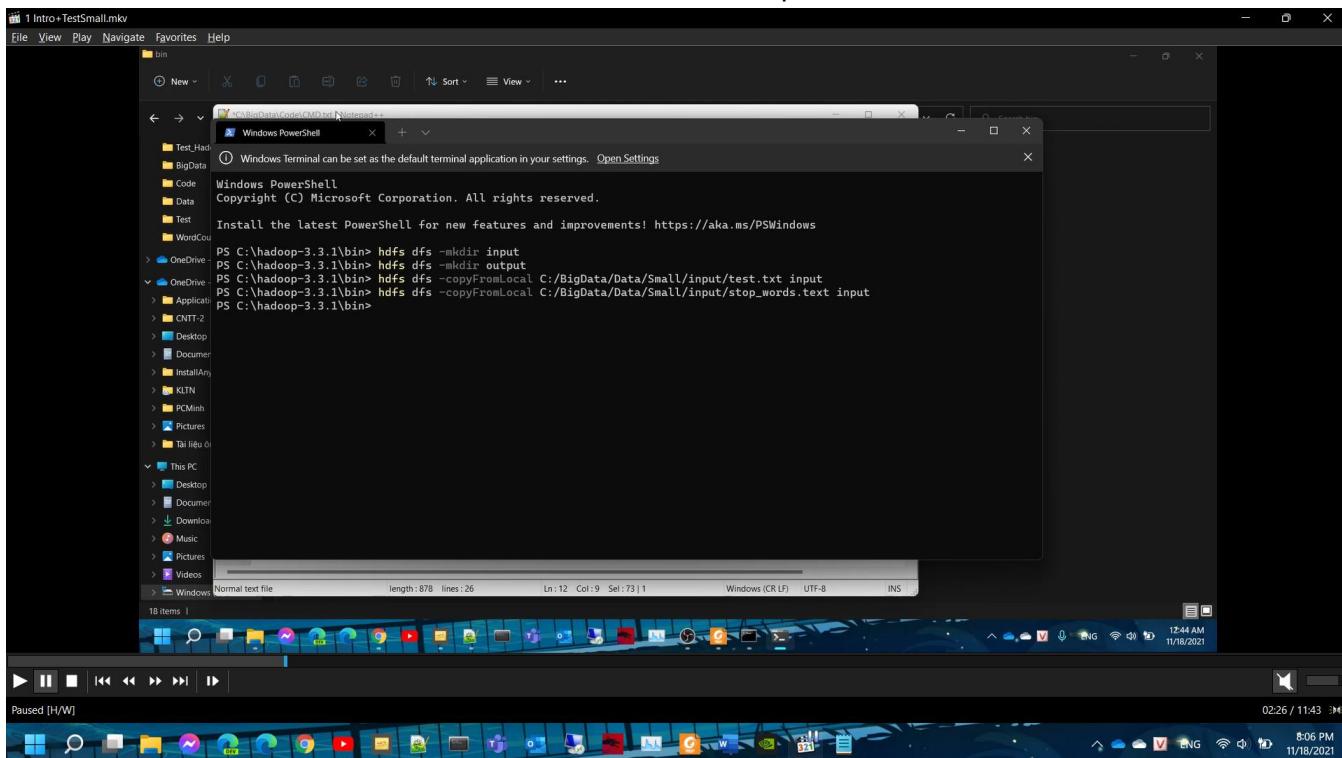
+ Đi đến thư mục bin của Hadoop để chạy các câu lệnh HDFS



+ Tạo các thư mục input và output trên HDFS



+ Lần lượt viết các lệnh để đưa 2 file test.txt và stop-words.text vào HDFS:



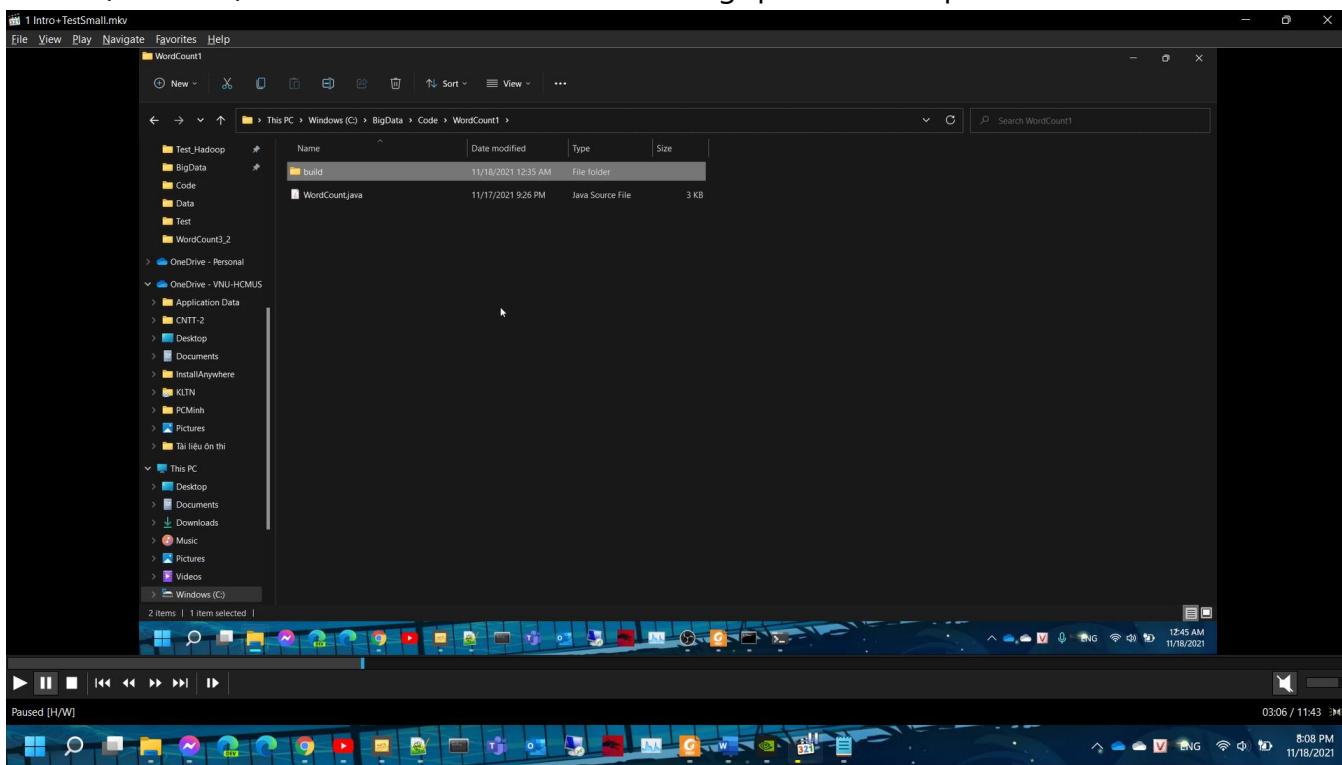
```

PS C:\hadoop-3.3.1\bin> hdfs dfs -mkdir input
PS C:\hadoop-3.3.1\bin> hdfs dfs -mkdir output
PS C:\hadoop-3.3.1\bin> hdfs dfs -copyFromLocal C:/BigData/Data/Small/input/test.txt input
PS C:\hadoop-3.3.1\bin> hdfs dfs -copyFromLocal C:/BigData/Data/Small/input/stop_words.txt input
PS C:\hadoop-3.3.1\bin>

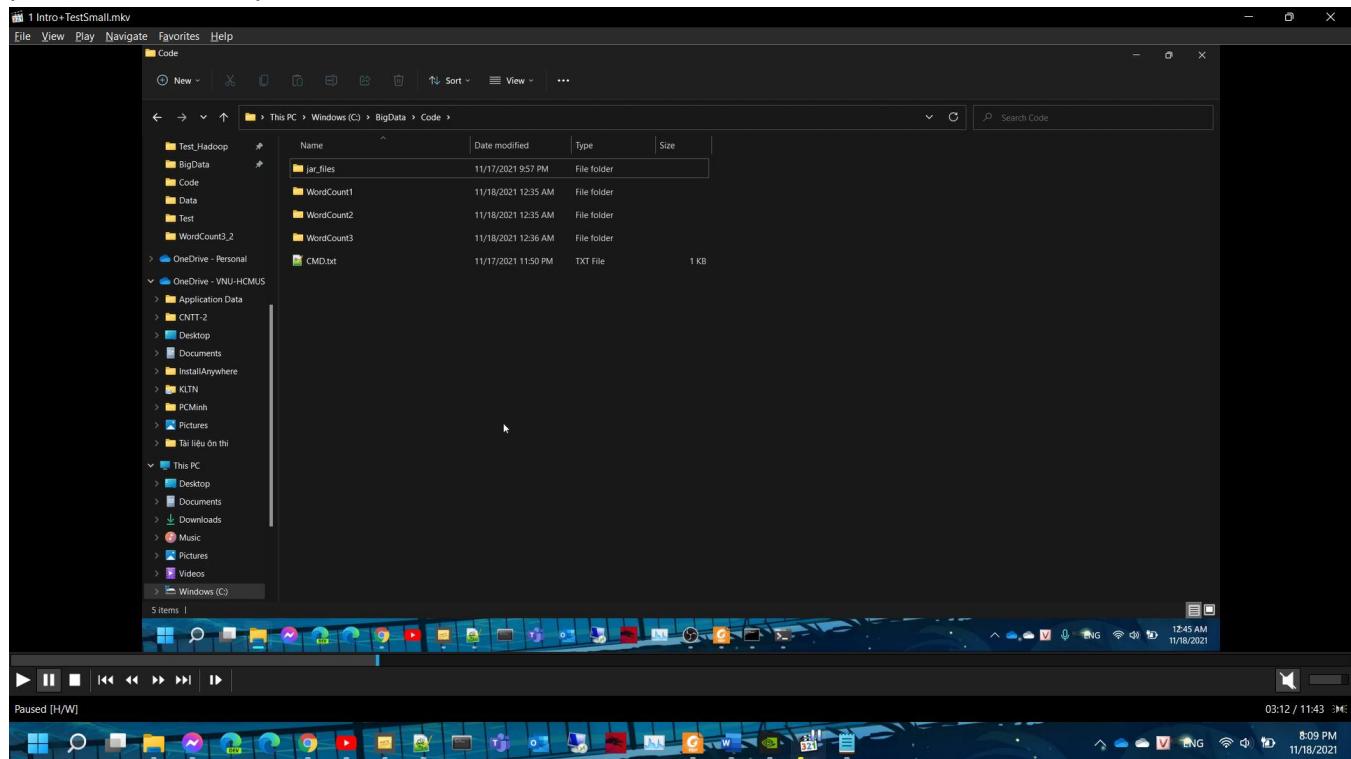
```

- **Bước 3:** Compile các file wordcount.java thành các file .jar

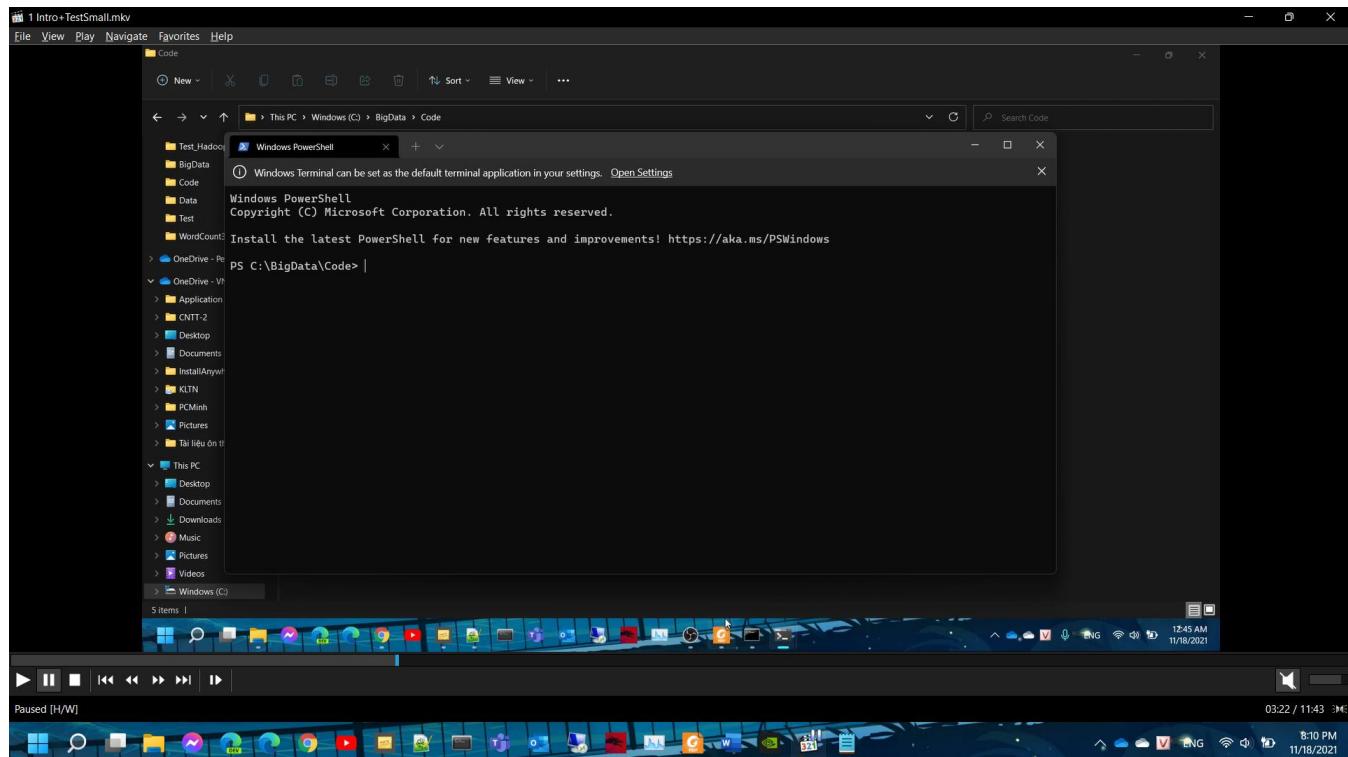
+ Tạo thư mục build để chứa các file .class trong quá trình compile



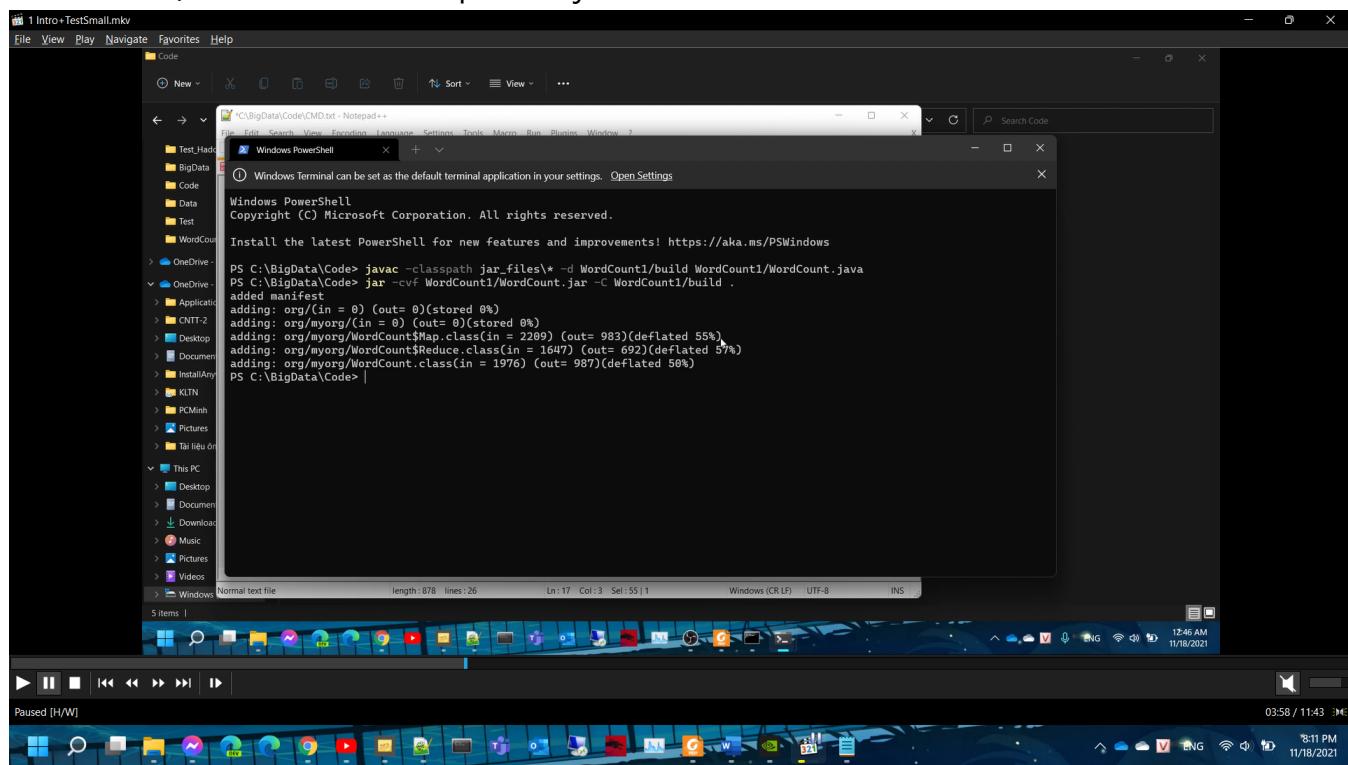
+ Ở đây compile bằng cách sử dụng thư mục jar\_files chứa các file .jar đã được tổng hợp để phục vụ việc compile



- + Mở hộp thoại cmd tại nơi chứa các file này (thư mục jar\_files và các chương trình wordcount)



- + Gõ 2 lệnh như sau để compile file .java



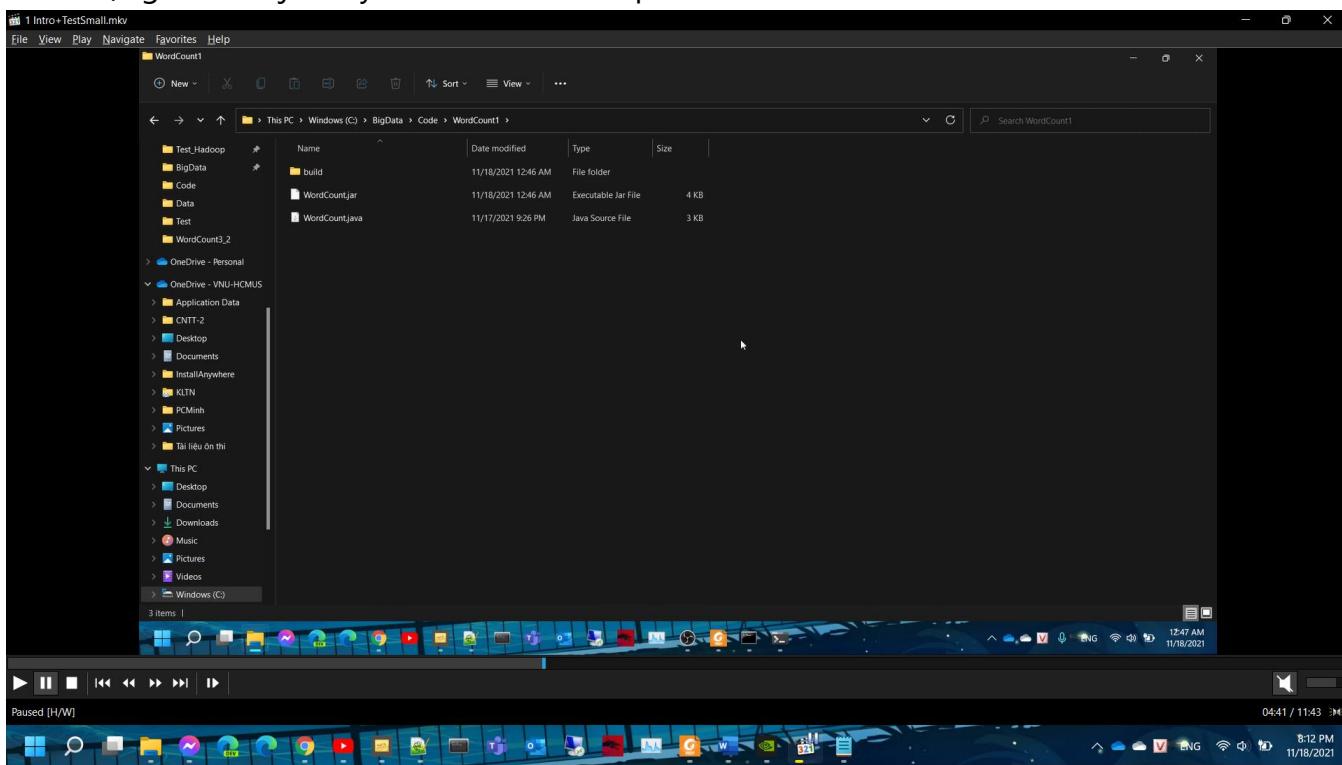
+ Tiếp tục thực với cho đến khi compile hết 3 chương trình wordcount

```

PS C:\BigData\Code> javac -classpath jar_files* -d WordCount1/build WordCount1/WordCount.java
PS C:\BigData\Code> jar -cvf WordCount1/WordCount.jar -C WordCount1/build .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%
adding: org/myorg/(in = 0) (out= 0)(stored 0%
adding: org/myorg/WordCount$Map.class(in = 2209) (out= 983)(deflated 55%
adding: org/myorg/WordCount$Reduce.class(in = 1647) (out= 692)(deflated 57%
PS C:\BigData\Code> javac -classpath jar_files* -d WordCount2/build WordCount2/WordCount.java
PS C:\BigData\Code> jar -cvf WordCount2/WordCount.jar -C WordCount2/build .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%
adding: org/myorg/(in = 0) (out= 0)(stored 0%
adding: org/myorg/WordCount$Map.class(in = 2584) (out= 1340)(deflated 55%
adding: org/myorg/WordCount$Reduce.class(in = 1647) (out= 693)(deflated 57%
PS C:\BigData\Code> javac -classpath jar_files* -d WordCount3/build WordCount3/WordCount.java
PS C:\BigData\Code> jar -cvf WordCount3/WordCount.jar -C WordCount3/build .
added manifest
adding: org/(in = 0) (out= 0)(stored 0%
adding: org/myorg/(in = 0) (out= 0)(stored 0%
adding: org/myorg/WordCount$Map.class(in = 4483) (out= 2111)(deflated 52%
adding: org/myorg/WordCount$Reduce.class(in = 1647) (out= 693)(deflated 57%
adding: org/myorg/WordCount.class(in = 2755) (out= 1388)(deflated 49%
PS C:\BigData\Code> |

```

+ Khi thành công ta sẽ thấy file wordcount.jar nằm cùng thư mục với file wordcount.java, ta sẽ sử dụng các file .jar này để đếm từ với MapReduce



- **Bước4:** Quay lại của sổ cmd ở thư mục bin của Hadoop và tiến hành chạy chương trình,
- + Sau chữ jar chính là đường dẫn đến file wordcount.jar ứng với từng phiên bản wordcount
  - + Đầu vào sẽ là input/test.txt tức là file test.txt mà ta đã đưa lên HDFS ở bước trên
  - + Ở đây kết quả sẽ được ghi vào thư mục output/testv\*\_out (tương ứng với 3 phiên bản ta thay \* bằng 1, 2, 3) trên HDFS:
- + Chạy phiên bản 1:

```
PS C:\hadoop-3.3.1\bin> hdfs dfs -mkdir input
PS C:\hadoop-3.3.1\bin> hdfs dfs -mkdir output
PS C:\hadoop-3.3.1\bin> hdfs dfs -copyFromLocal C:/BigData/Data/Small/input/test.txt input
PS C:\hadoop-3.3.1\bin> hdfs dfs -copyFromLocal C:/BigData/Data/Small/input/stop_words.text input
PS C:\hadoop-3.3.1\bin> hdfs dfs -ls input
Found 2 items
-rw-r--r--- 1 Admin supergroup      29 2021-11-18 00:44 input/stop_words.text
-rw-r--r--- 1 Admin supergroup    275 2021-11-18 00:44 input/test.txt
PS C:\hadoop-3.3.1\bin> hadoop jar C:/BigData/Code/WordCount/wordcount.jar org.myorg.WordCount input/test output/testv1_out
```

## + Chạy phiên bản 2:

```

PS C:\hadoop-3.3.1\bin> hadoop jar C:/BigData/Code/WordCount2/wordcount.jar org.myorg.WordCount input/test.txt output/testv2_out
2021-11-18 00:49:22,694 INFO client.DefaultHttpAmmFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-18 00:49:23,158 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Admin/.staging/job_1637171003367_0002
2
2021-11-18 00:49:23,329 INFO input.FileInputFormat: Total input files to process : 1
2021-11-18 00:49:23,395 INFO mapreduce: number of splits:1
2021-11-18 00:49:23,402 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1637171003367_0002
2021-11-18 00:49:23,403 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-18 00:49:23,420 INFO conf.Configuration: resource-types.xml not found
2021-11-18 00:49:23,438 INFO resource.ResourceUtil: Unable to find resource-types.xml
2021-11-18 00:49:23,699 INFO resource.YarnClientImpl: Submitted application application_1637171003367_0002
2021-11-18 00:49:23,739 INFO mapreduce.Job: The url to track the job: http://MSI-8088/proxy/application_1637171003367_0002/
2021-11-18 00:49:23,748 INFO mapreduce.Job: Running job: job_1637171003367_0002

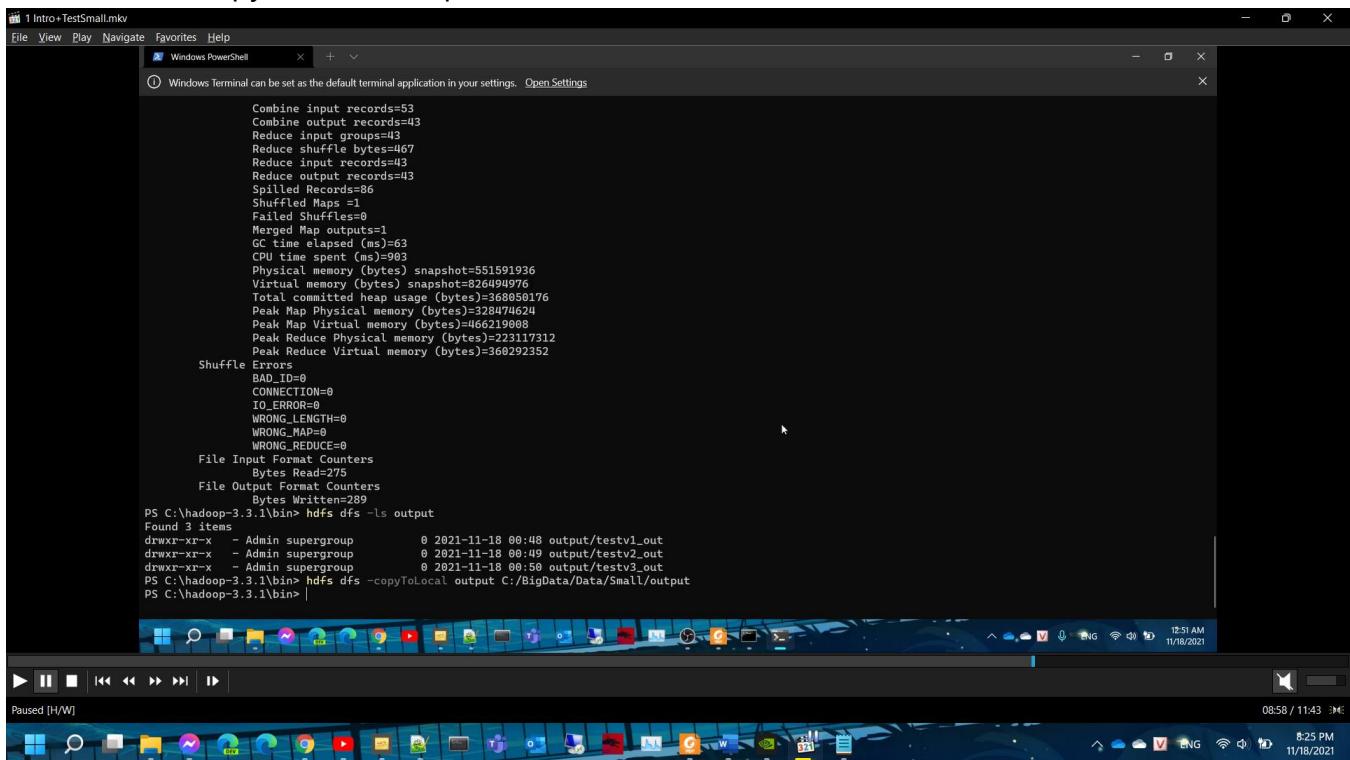
```

## + Chạy phiên bản 3: cần thêm vào -skip input/stop\_words.text là đường dẫn đến file chứa các stop word

```

PS C:\hadoop-3.3.1\bin> hadoop jar C:/BigData/Code/WordCount3/wordcount.jar org.myorg.WordCount input/test.txt output/testv3_out -skip input/stop_words.text
2
Map input records=5
Map output records=57
Map output bytes=486
Map output materialized bytes=496
Input split bytes=112
Combine input records=57
Combine output records=46
Reduce input groups=6
Reduce output bytes=96
Reduce input records=6
Reduce output records=46
Spilled Records=92
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=61
CPU time spent (ms)=655
Physical memory (bytes) snapshot=555220992
Virtual memory (bytes) snapshot=827564032
Total committed heap usage (bytes)=366477312
Peak Map Physical memory (bytes)=328159232
Peak Map Virtual memory (bytes)=465018880
Peak Reduce Physical memory (bytes)=227061760
Peak Reduce Virtual memory (bytes)=362651648

```

**- Bước 5:** copy các file kết quả về Local

A screenshot of a Windows desktop environment. At the top, there is a taskbar with various pinned icons. Below the taskbar, a Windows PowerShell window is open, titled '1 Intro + TestSmall.mkv'. The window displays the output of a Hadoop job, including counters for Combine input records, Reduce input groups, and various memory and shuffle metrics. The command used was 'hdfs dfs -copyToLocal output C:/BigData/Data/Small/output'. The desktop background shows a blue and white abstract pattern.

```
Combine input records=53
Combine output records=43
Reduce input groups=43
Reduce shuffle bytes=467
Reduce input records=43
Reduce output records=43
Spilled Records=86
Shuffled Maps=8
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=62
CPU time spent (ms)=903
Physical memory (bytes) snapshot=551591936
Virtual memory (bytes) snapshot=826494976
Total committed heap usage (bytes)=368050176
Peak Map Physical memory (bytes)=328474624
Peak Map Virtual memory (bytes)=466219008
Peak Reduce Physical memory (bytes)=223117312
Peak Reduce Virtual memory (bytes)=360292352

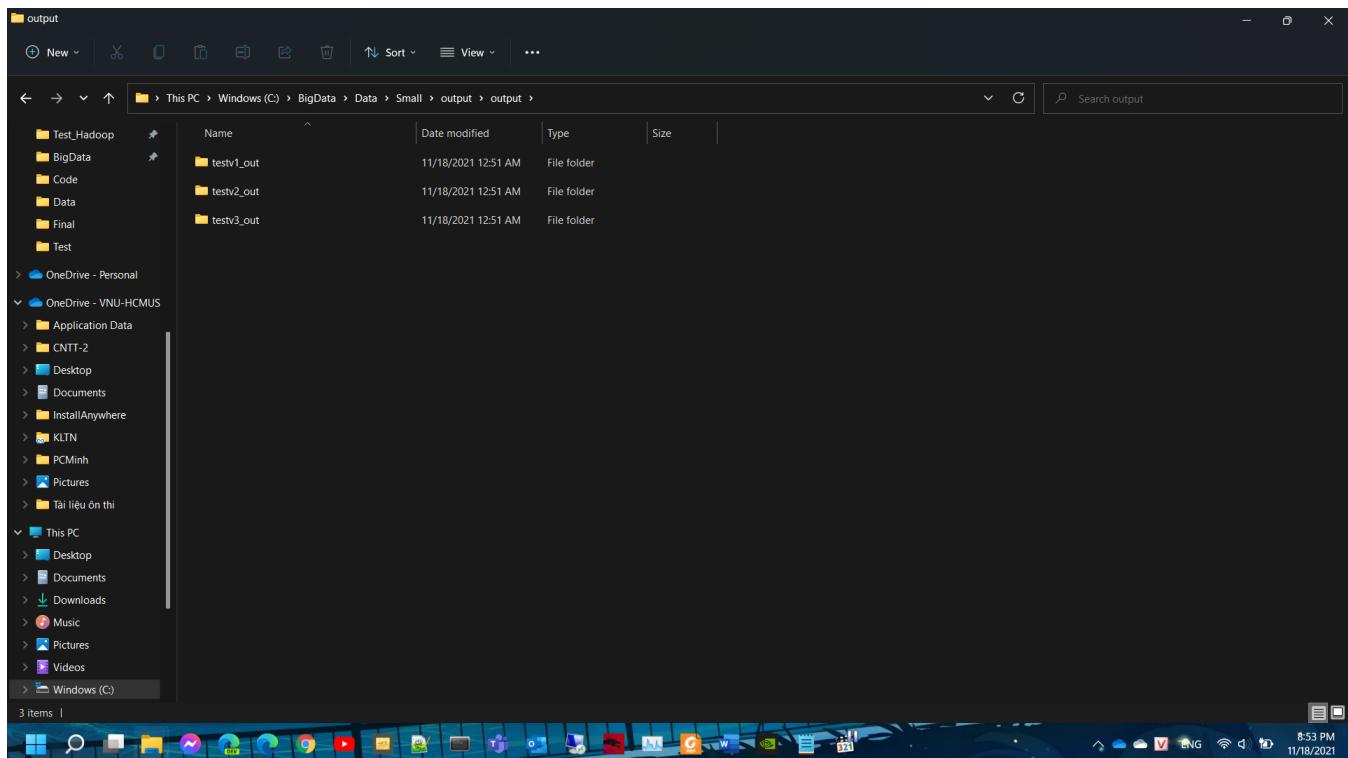
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=275
File Output Format Counters
Bytes Written=289

PS C:\hadoop-3.3.1\bin> hdfs dfs -ls output
Found 3 items
drwxr-xr-x  - Admin supergroup          0 2021-11-18 00:48 output/testv1.out
drwxr-xr-x  - Admin supergroup          0 2021-11-18 00:49 output/testv2.out
drwxr-xr-x  - Admin supergroup          0 2021-11-18 00:50 output/testv3.out
PS C:\hadoop-3.3.1\bin> hdfs dfs -copyToLocal output C:/BigData/Data/Small/output
PS C:\hadoop-3.3.1\bin>
```

**- Bước 6:** Kiểm tra kết quả, xem video để dễ dàng thấy các so sánh ở đây em sẽ chụp một phần của 3 file kết quả. Các file kết quả (3 file như hình dưới đây) em sẽ để trong thư mục

## Results\Level1\Test



## + Kết quả phiên bản 1

The screenshot shows a Notepad++ window displaying the contents of the file "testv1\_out/part-r-00000". The file contains a list of words and their counts, such as "I 4", "you 1", "and 1", etc., up to line 40. The status bar at the bottom indicates the file is a "Normal text file" with 339 length, 54 lines, and the current position at Ln:1 Col:1 Pos:1. The encoding is set to "UTF-8".

```
1 ' 4
2 , 1
3 . 4
4 . 2
5 ... 1
6 ? 2
7 Been 1
8 By 1
9 DC 1
10 How 1
11 I 1
12 Love 1
13 RT 1
14 They 1
15 When 1
16 You 1
17 Your 1
18 and 1
19 anytime 1
20 are 1
21 be 1
22 Beat 1
23 decided 1
24 don 1
25 for 2
26 fun 1
27 gonna 1
28 heart 1
29 if 1
30 in 1
31 its 1
32 know 1
33 ll 2
34 long 1
35 meet 1
36 more 2
37 no 1
38 rapidly 1
39 reason 1
40 see 1
41 smile 1
42 ... 1
```

## + Kết quả phiên bản 2

```

1 and 1
2 anytime 1
3 are 1
4 be 1
5 beat 1
6 been 1
7 by 1
8 dc 1
9 decided 1
10 don 1
11 for 2
12 fun 1
13 gonna 1
14 heart 1
15 how 1
16 i 1
17 if 1
18 in 1
19 its 1
20 know 1
21 ll 2
22 long 1
23 love 1
24 meet 1
25 more 2
26 no 1
27 rapidly 1
28 reason 1
29 rt 1
30 see 1
31 smile 1
32 someone 1
33 soon 1
34 special 1
35 t 1
36 thanks 1
37 the 2
38 they 1
39 to 1
40 too 1
41 ve 1
...

```

Normal text file length : 306 lines : 47 Ln:1 Col:1 Pos:1 Unix (LF) UTF-8 INS

## + Kết quả phiên bản 3

```

1 anytime 1
2 are 1
3 be 1
4 beat 1
5 been 1
6 by 1
7 do 1
8 decided 1
9 don 1
10 for 2
11 fun 1
12 gonna 1
13 heart 1
14 how 1
15 i 1
16 if 1
17 in 1
18 its 1
19 know 1
20 ll 2
21 long 1
22 love 1
23 meet 1
24 more 2
25 no 1
26 rapidly 1
27 reason 1
28 rt 1
29 see 1
30 smile 1
31 someone 1
32 soon 1
33 special 1
34 t 1
35 thanks 1
36 they 1
37 too 1
38 ve 1
39 way 3
40 when 1
41 will 1
...

```

Normal text file length : 289 lines : 44 Ln:1 Col:7 Pos:7 Unix (LF) UTF-8 INS

**c) Thực thi các chương trình trên các file dung lượng lớn:**

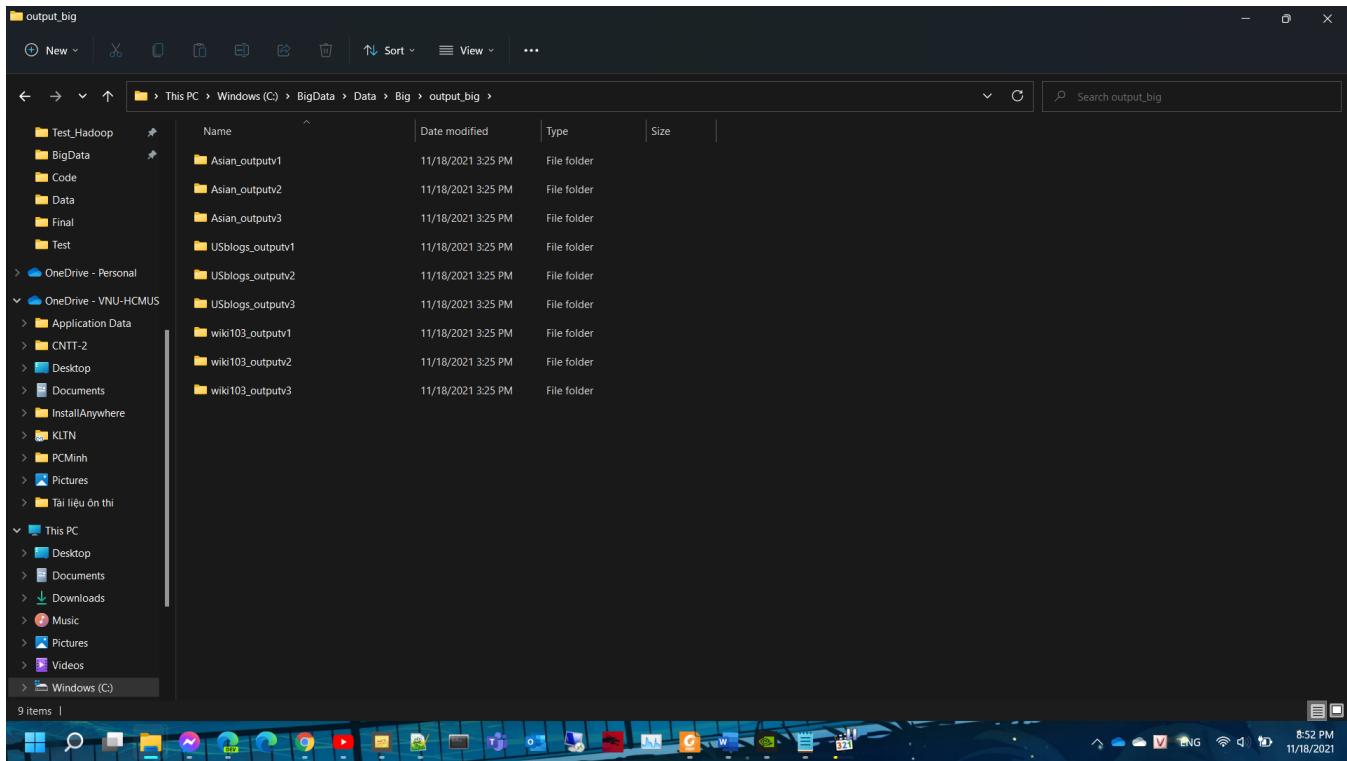
\* Các file sử dụng: đều là các file văn bản với ngôn ngữ chính là tiếng Anh, và một phần nhỏ các ngôn ngữ khác

- wiki103.txt: Nguồn: [variety-text-corpus | Kaggle](#). Trên 100MB
- USblogs.txt: Nguồn: [Tweets Blogs News - Swiftkey Dataset 4million | Kaggle](#). Trên 200MB
- Asian.txt: Nguồn: [Yelp Reviews Grouped by Cuisine | Kaggle](#). Trên 500MB

Name	Date modified	Type	Size
Asian.txt	5/26/2021 6:24 AM	TXT File	108,744 KB
USblogs.txt	9/25/2019 3:27 AM	TXT File	205,235 KB
wiki103.txt	11/10/2021 11:46 AM	TXT File	527,899 KB

\* Kết quả thực thi: (cách thức chạy tương tự như với file test.txt đã nêu ở phần b/)

- Các kết quả in ra màn hình được em copy ra trong lúc chạy và lưu lại trong file screen\_results.txt ở thư mục Results\Level1
- Các file kết quả (9 file như hình dưới đây) được lưu trên GGDrive - link được để trong file Results\Level1\link\_results.txt



### 3.2. Mức 2:

#### a) Viết chương trình Python:

- Sử dụng 2 package là Mrjob và Regex
- Mrjob: cho phép viết các chương trình MapReduce trên python
- Regex: Công cụ xử lý chuỗi
- File wordcount1.py: Trường hợp phân biệt hoa thường

```
# import thư viện
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+") # lọc ra các từ

class MRWordFreqCount(MRJob):
# map task
    def mapper(self, _, line):
        for word in WORD_RE.findall(line): # ngắt các từ
            yield (word, 1)
# reduce task
    def reducer(self, word, counts):
        yield(word, sum(counts)) # đếm số lượng mỗi từ
if __name__ == '__main__':
    MRWordFreqCount.run() # chạy job
```

- File wordcount2.py: Trường hợp không phân biệt hoa thường
- Bổ sung thêm phương thức lower() ở pha map

```
# import thư viện
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+") # lọc ra các từ

class MRWordFreqCount(MRJob):
# map task
    def mapper(self, _, line):
        for word in WORD_RE.findall(line): # ngắt các từ
            yield (word.lower(), 1) # chuyển chữ hoa thành chữ thường
# reduce task
    def reducer(self, word, counts):
        yield(word, sum(counts)) # đếm số lượng mỗi từ
if __name__ == '__main__':
    MRWordFreqCount.run() # chạy job
```

- File wordcount3.py: Tìm từ xuất hiện nhiều nhất trong tài liệu (không phân biệt hoa thường)

- Gồm 2 bước:

- + **Bước 1:** chia từ và đếm từ
- + **Bước 2:** đếm từ xuất hiện nhiều nhất

```
# import thư viện
from mrjob.job import MRJob
from mrjob.step import MRStep
import re

WORD_RE = re.compile(r"[\w']+") # giúp ngắt 1 dòng thành các từ

class MRWordFreqCount(MRJob):
# map task
    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
            yield (word.lower(), 1) # chuyển chữ hoa thành chữ thường
# reduce task

    def reducer_count_words(self, word, counts):
        yield None, (sum(counts), word) # đếm số lượng mỗi từ
# tìm ra từ xuất hiện nhiều nhất
    def reducer_find_max_word(self, _, word_count_pairs):
        try:
            yield max(word_count_pairs)
        except ValueError:
            pass
# trình bày các bước xử lý
    def steps(self):
        return [
```

```
    MRStep(mapper=self.mapper,
            reducer=self.reducer_count_words),
    MRStep(reducer=self.reducer_find_max_word)
]
if __name__ == '__main__':
    MRWordFreqCount.run() # chạy job
```

- Nội dung các file code đã được đính kèm trong thư mục Code\Level2

### b) Chạy trên máy local

- Cú pháp sử dụng: chạy thử nghiệm trên file apple.txt được đính kèm trong thư mục Data\Level2

```
>python yourcode.py data.txt
```

- Kết quả minh họa version 2:

```
D:\>python wordcount2.py apple.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\TAI\AppData\Local\Temp\wordcount2.TAI.20211117.193916.998535
Running step 1 of 1...
job output is in C:\Users\TAI\AppData\Local\Temp\wordcount2.TAI.20211117.193916.998535\output
Streaming final output from C:\Users\TAI\AppData\Local\Temp\wordcount2.TAI.20211117.193916.998535\output...
"a"      3
"ah"     2
"an"     1
"apple"   2
"have"   4
"i"       4
"pen"     4
"pineapple" 2
Removing temp directory C:\Users\TAI\AppData\Local\Temp\wordcount2.TAI.20211117.193916.998535...
```

### c) Chạy trên Hdfs

- Chạy trên dữ liệu đã được upload sẵn trên Hdfs

```
> python yourcode.py -r hadoop --hadoop-streaming-jar <đường dẫn chứa file hadoop-streaming.jar > hdfs://input/data.txt
```

- Chạy wordcount trên HDFS

```
hadoop@hai-ThinkPad-L13-Gen-2:~/hadoop/sbin$ python3 /home/hadoop/hadoop_python/wordcount/wordcount3.py -r hadoop --hadoop-streaming-jar /home/hadoop/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.1.jar hdfs://Input/book/books.txt
No config found; falling back on auto-configuration
No configs specified for hadoop runner
Looking for hadoop binary in /home/hadoop/hadoop/bin...
Found hadoop binary: /home/hadoop/hadoop/bin/hadoop
Using Hadoop version 3.3.1
Creating temp directory /tmp/wordcount3.hadoop_20211118.115401.103976
uploading working dir files to hdfs://user/hadoop/tmp/mrJob/wordcount3.hadoop_20211118.115401.103976/files/...
Copying other local files to hdfs://user/hadoop/tmp/mrJob/wordcount3.hadoop_20211118.115401.103976/files/
Running step 1 of 2...
  packageJobJar: [/tmp/hadoop-unjar14547623149733238535/] [] /tmp/streamjob11488584185256412336.jar tmpDir=null
  Connecting to ResourceManager at /0.0.0.0:8032
  Connecting to ResourceManager at /0.0.0.0:8032
  Disabling Error Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1637235733614_0011
  Total number of files to process : 1
  number of splits:2
  Submitting tokens for job: job_1637235733614_0011
  Executing with tokens: []
  resource-types.xml not found
  Unable to find 'resource-types.xml'.
  Submitted application application_1637235733614_0011
  The url to track the job: http://hai-ThinkPad-L13-Gen-2:8088/proxy/application_1637235733614_0011/
  Running job: job_1637235733614_0011
  Job job_1637235733614_0011 running in uber mode : false
  map 0% reduce 0%
  map 50% reduce 0%
  map 100% reduce 0%
  map 100% reduce 100%
  Job job_1637235733614_0011 completed successfully
  Output directory: hdfs://user/hadoop/tmp/mrJob/wordcount3.hadoop_20211118.115401.103976/step-output/0000
Counters: 54
  File Input Format Counters
    Bytes Read=1596432
  File Output Format Counters
    Bytes Written=619020
  System Counters
    FILE: Number of bytes read=3133537
    FILE: Number of bytes written=7098889
    FILE: Number of large read operations=0
    FILE: Number of read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1596430
    HDFS: Number of bytes written=619020
    HDFS: Number of bytes written=619020
    HDFS: Number of large read operations=0
    HDFS: Number of read operations=11
    HDFS: Number of write operations=2
  Job Counters
    Data-local map tasks=2
    Launched map tasks=2
    Launched reduce tasks=1
    Total megabyte-milliseconds taken by all map tasks=6809600
    Total megabyte-milliseconds taken by all reduce tasks=2775040
```

- Kết quả minh họa:

- + Wordcount version 2:

```

yf11120913 - 
"yrs"    2
"yu"     1
"yulelog"      1
"yuletide"     2
"yum"     3
"yummyum"      1
"yumyun"      3
"yung"     1
"yup"     1
"yvonne"      1
"ywimpled"     1
"zamatejch_"   1
"zarathustra"   3
"zaretsky"      1
"zeal"      3
"zealous"      2
"zebra"     1
"zenith"      3
"zephyrs"      1
"zermatt"      1
"zero"       1
"zest"       1
"zigzag"      2
"zigzagging"   1
"zigzags"      1
"zinfandel"    4
"zingari"      1
"zion"       5
"zip"        1
"zivio"      1
"zmellz"      1
"zodiac"      2
"zodiacal"    2
"zoe"        105
"zones"      1

```

+ Wordcount version 3: từ xuất hiện nhiều nhất là từ "the" với 15092 số lần xuất hiện

```

WRONG_REDUCE=0
job output is in hdfs://user/hadoop/tmp/mrjob/wordcount3.hadoop.20211118.115401.103976/output
Streaming final output from hdfs://user/hadoop/tmp/mrjob/wordcount3.hadoop.20211118.115401.103976/output...
15092   "the"

```

#### d) Bài tập áp dụng

- Dữ liệu được lấy từ 3 cuốn dưới đây (file.txt)

Book 1: <https://www.gutenberg.org/ebooks/20417>

Book 2: <https://www.gutenberg.org/ebooks/5000>

Book 3: <https://www.gutenberg.org/ebooks/4300>

- File kết quả đã được đính kèm trong thư mục Results\Level2:

Output của các cuốn sách tương ứng với từng version:

 wc1_book1.txt	11/18/2021 8:36 PM	Text Document	144 KB
 wc1_book2.txt	11/18/2021 8:37 PM	Text Document	247 KB
 wc1_book3.txt	11/18/2021 8:37 PM	Text Document	418 KB
 wc2_book1.txt	11/18/2021 8:37 PM	Text Document	123 KB
 wc2_book2.txt	11/18/2021 8:37 PM	Text Document	218 KB
 wc2_book3.txt	11/18/2021 8:37 PM	Text Document	370 KB
 wc3_book1.txt	11/18/2021 8:37 PM	Text Document	1 KB
 wc3_book2.txt	11/18/2021 8:37 PM	Text Document	1 KB
 wc3_book3.txt	11/18/2021 8:38 PM	Text Document	1 KB

## 4. Tài liệu tham khảo:

- [HDFS Commands - GeeksforGeeks](#)
- [Example: WordCount v1.0 \(cloudera.com\)](#)
- [https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html](#)
- [https://mrjob.readthedocs.io/en/latest/](#)
- [https://docs.python.org/3/library/re.html](#)
- [https://mrjob.readthedocs.io/en/latest/\\_sources/guides/writing-mrjobs.txt](#)