

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH



LAB – 02:
Bài Toán Đếm Từ với MapReduce

Lớp: 19_21
Môn: Nhập môn dữ liệu lớn

Niên khóa: 2021-2022

Mục lục

I. Thông tin sinh viên	3
II. Nội dung tìm hiểu	5
1. Level 1	5
2. Level 2	13
3. Link video minh chứng:	17
III. Tài liệu tham khảo	18

I. Thông tin sinh viên

1. Thông tin nhóm:

Tên nhóm: Gaming House.

Danh sách thành viên:

STT	Họ tên	MSSV
1	Đỗ Thái Duy	19120492
2	Huỳnh Quốc Duy	19120494
3	Phạm Đức Huy	19120534
4	Lê Thành Lộc	19120562

2. Bảng phân công công việc:

MSSV	Họ tên	Công việc
19120492	Đỗ Thái Duy	<ul style="list-style-type: none">- Code: Triển khai code WordCount phiên bản 1 và 2.- Viết báo cáo: Giải thích và mô tả mã nguồn đã viết ở level 1.- Quay video demo level 1.
19120494	Huỳnh Quốc Duy	<ul style="list-style-type: none">- Code: Triển khai code WordCount phiên bản 3, Refactor lại code và ghi comment đầy đủ cho cả 3 phiên bản.- Viết báo cáo: Giải thích và mô tả mã nguồn WordCount phiên bản 3 ở level 1.- Chỉnh sửa, thêm phụ đề cho các video demo.
19120534	Phạm Đức Huy	<ul style="list-style-type: none">- Code: Thực hiện cài các package cần thiết và code WordCount phiên bản 1 và 2 bằng ngôn ngữ python.- Viết báo cáo: Giải thích và mô tả mã nguồn WordCount phiên bản 1 và 2 ở level 2.- Quay video demo level 2.

19120562	Lê Thành Lộc	<ul style="list-style-type: none"> - Code: Thực hiện cài các package cần thiết và code WordCount phiên bản 3 bằng ngôn ngữ python. - Viết báo cáo: Giải thích và mô tả mã nguồn WordCount phiên bản 3 ở level 2. - Tìm kiếm và tạo các test case theo yêu cầu của level 1 và level 2.
----------	--------------	--

3. Đánh giá mức độ hoàn thành:

STT	Họ tên	MSSV	Mức độ hoàn thành
1	Đỗ Thái Duy	19120492	100%
2	Huỳnh Quốc Duy	19120494	100%
3	Phạm Đức Huy	19120534	100%
4	Lê Thành Lộc	19120562	100%

II. Nội dung tìm hiểu

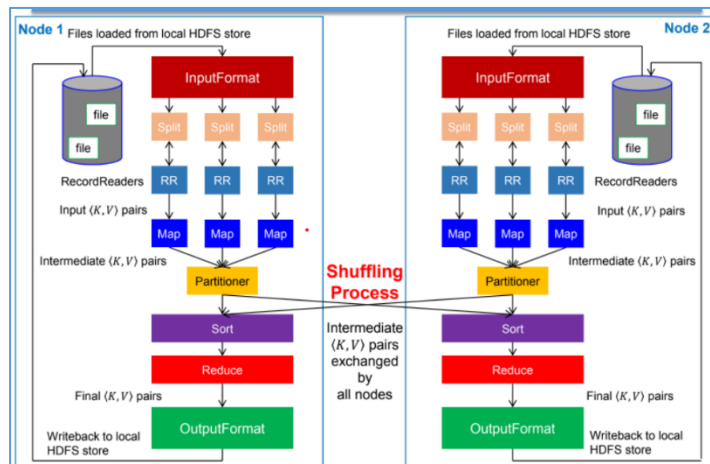
1. Level 1

a) Giải thích mã nguồn:

i) WordCount_v1.java:

Mô tả: chương trình thống kê số lượng của mỗi từ ở mức cơ bản (không tiền xử lý).

Quy trình hoạt động của MapReduce:



Lớp WordCount sẽ bao gồm phương thức main và run để cấu hình và chạy chương trình cùng các lớp bên trong là **Map** và **Reduce**.

```
public class WordCount extends Configured implements Tool {  
  
    private static final Logger LOG = Logger.getLogger(WordCount.class);  
  
    public static void main(String[] args) throws Exception { ...  
    }  
    public int run(String[] args) throws Exception { ...  
    }  
    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> { ...  
    }  
    public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> { ...  
    }  
}
```

Phương thức **main** sẽ gọi **ToolRunner**, công cụ này tạo và chạy một phiên bản WordCount mới.

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

Phương thức **run** sẽ cấu hình công việc, bắt đầu công việc, đợi công việc hoàn thành và sau đó trả về một giá trị số nguyên dưới dạng flag (thành công/thất bại). Ở đây ta sẽ thiết lập file jar để chạy chương trình, đường dẫn input và output cho chương trình, thiết lập lớp map và reduce cho công việc cũng như xác định kiểu output cho cả Map và Reduce.

```

public int run(String[] args) throws Exception {
    Job job = Job.getInstance(getConf(), "wordcount");

    job.setJarByClass(this.getClass());
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    return job.waitForCompletion(true) ? 0 : 1;
}

```

Lớp **Map** (mở rộng từ Mapper) biến đổi các giá trị key/value thành các cặp key/value trung gian để gửi đến Reducer. Ở đây phương thức map sẽ sử dụng biểu thức chính quy để phân tách từng dòng của đoạn văn bản đầu vào, từ đó có được output trung gian, mỗi kí tự (key), nếu không phải khoảng trắng, thì sẽ có giá trị đếm (value) là 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;

    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();
        Text currentWord = new Text();
        for (String word : WORD_BOUNDARY.split(line)) {
            if (word.isEmpty()) {
                continue;
            }
            currentWord = new Text(word);
            context.write(currentWord, one);
        }
    }
}

```

Kiểu dữ liệu của output key-value từ mapper sẽ trùng với kiểu dữ liệu của input key-value ở reducer. Như có thể thấy ở đoạn khai báo hàm bên dưới (Text - IntWritable tương ứng ở 2 hàm).

```

public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> { ...
}
public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> { ...
}

```

Lớp **mapper** tạo một cặp **key/value** cho mỗi từ bao gồm từ và giá trị **IntWritable**. Lớp Reducer xử lý từng cặp, thêm một cho từ hiện tại trong cặp key/value vào tổng số lần xuất hiện của từ đó từ tất cả mappers. Phương thức reduce chạy một lần với mỗi key nhận được từ pha shuffle and sort của MapReduce.

```
public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    public void reduce(Text word, Iterable<IntWritable> counts, Context context)
        throws IOException, InterruptedException {
        int sum = 0;

        for (IntWritable count : counts) {
            sum += count.get();
        }
        context.write(word, new IntWritable(sum));
    }
}
```

Như những gì đã được học ở lớp lý thuyết, 2 dòng lệnh dưới ở hàm run sẽ xác định kiểu dữ liệu của các cặp key-value do Reducer tạo ra để xuất ra file trên HDFS.

```
// Xác định kiểu output cho cả map và reduce
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
```

ii) WordCount_v2.java:

Mô tả: Thực hiện như phiên bản 1.0 nhưng lúc này không đếm các ký tự không phải là từ, không phân biệt hoa thường.

Vẫn sử dụng các phương thức **run**, **main** cùng lớp Reduce kế thừa từ phiên bản 1.0 nhưng thực hiện vài chỉnh sửa ở lớp **Map**.

Thêm phương thức setup để khởi tạo biến hệ thống **wordcount.case.sensitive** giúp bật/tắt tính năng không phân biệt hoa thường cũng như không đếm dấu câu. Giá trị mặc định của **wordcount.case.sensitive** là False.

```
protected void setup(Mapper.Context context)
    throws IOException,
        InterruptedException {
    Configuration config = context.getConfiguration();
    this.caseSensitive = config.getBoolean("wordcount.case.sensitive", false);
}
```

Nếu biến **wordcount.case.sensitive** có giá trị False thì thực hiện quá trình tiền xử lý: loại bỏ các dấu câu trong văn bản cũng như đưa các từ về dạng in thường (lowercase).

```
public void map(LongWritable offset, Text lineText, Context context)
    throws IOException, InterruptedException {
    String line = lineText.toString();
    if (!caseSensitive) {
        line = line.replaceAll("'", "");
        line = line.replaceAll("[^a-zA-Z0-9 ]", " ");
        line = line.toLowerCase();
    }
}
```

Lưu ý: Ta có thể tắt tính năng này bằng cách khai báo **-Dwordcount.case.sensitive=true** trong dòng lệnh để chạy file java. Ví dụ như:

```
hadoop jar wordcount.jar org.myorg.WordCount -Dwordcount.case.sensitive=true
/user/cloudera/wordcount/input /user/cloudera/wordcount/output
```

Lúc này kết quả output sẽ không phân biệt ký tự hoa thường cũng như có đếm các ký tự không phải là từ.

iii) WordCount_v3.java:

Mô tả: Phát triển tiếp từ phiên bản 2.0 và bổ sung việc không đếm các từ nằm trong danh sách stop_words.txt.

Chỉ kế thừa lớp reduce từ phiên bản 2.0 và có nhiều chỉnh sửa ở hàm run và lớp Map.

Chương trình này sẽ sử dụng đến **distributed cache** với mục đích phân phối các dữ liệu cần thiết cho công việc. Ta sẽ thêm một vài dòng lệnh ở hàm **run()** để đưa file stop_words.txt vào làm mẫu từ được **distributed cache** chỉ định bộ đếm bỏ qua.

```
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());
        LOG.info("Added file to the distributed cache: " + args[i]);
    }
}
```

Trong phiên bản 3.0, ngoài **Mapper** và **Reduce** thì ta sử dụng thêm lớp **Combiner** vào cấu hình công việc. **Combiner** được chạy trên mỗi mappers để xử lý thông tin cục bộ trước khi được gửi đến reducer, nó giúp cắt giảm lượng dữ liệu bị xáo trộn giữa các bản đồ và giảm bớt các tác vụ. Ví dụ, gửi <word, 1> và <word,1> đến reducer thì combiner sẽ kết hợp chúng thành <word,2> trước khi chuyển kết quả đến reducer.

```
job.setMapperClass(Map.class);
job.setCombinerClass(Reduce.class);
job.setReducerClass(Reduce.class);
```

Tiếp theo, chỉnh sửa hàm setup() kế thừa từ phiên bản 2.0 ở lớp Map. Nếu trong dòng lệnh chạy chương trình có xuất hiện file chứa mẫu từ cần bỏ qua (stop_words.txt) thì lấy danh sách mẫu từ đó từ **distributed cache** và chuyển đến phương thức **parseSkipFile** được định nghĩa sau đây:

```
if (config.getBoolean("wordcount.skip.patterns", false)) {
    URI[] localPaths = context.getCacheFiles();
    parseSkipFile(localPaths[0]);
}
```

Hàm **parseSkipFile()** lấy danh sách mẫu từ cần bỏ qua vào tập hợp chuỗi **patternToSkip** chứa các dấu câu và các từ thừa cần được bỏ qua.


```
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));
    }
}
```

Cuối cùng thay đổi cách xử lý với từng từ sau khi được phân tách từ câu văn bản. Lúc này nếu biến word trống hoặc nó chứa một trong các mẫu đã được xác định cần bỏ qua trong **patternsToSkip** thì vòng lặp For sẽ tiếp tục mà không ghi giá trị vào biến context.

```
for (String word : WORD_BOUNDARY.split(line)) {
    if (word.isEmpty() || patternsToSkip.contains(word)) {
        continue;
    }
    currentWord = new Text(word);
    context.write(currentWord, one);
}
```

b) Dẫn chứng các hình ảnh của quá trình thực thi và video demo minh chứng:

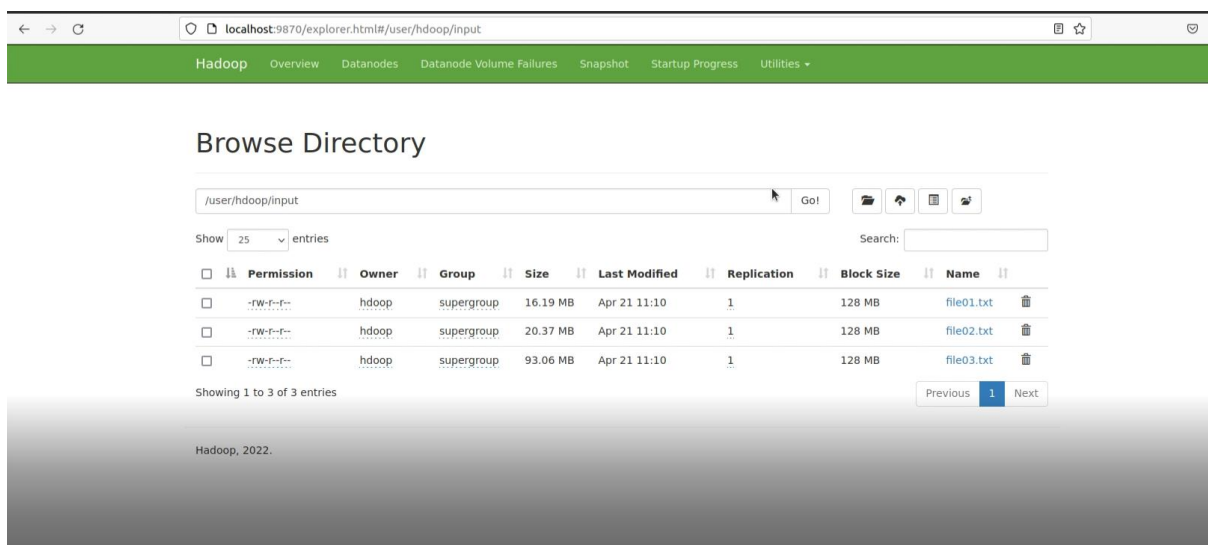
- Khởi động **NameNode**, **DataNode**, **ResourceManager** và **NodeManagers** với `./start-dfs.sh` và `./start-yarn.sh` và chạy lệnh `jps` để liệt kê danh sách các JVM (Java HotSpot) đang có quyền truy cập.

```
bash: export: `HADOOP_OPTS-Djava.library.path=/home/hadoop/hadoop-3.3.2/bin/hadoop/lib/native': not a valid identifier
duy@ubuntu:~$ su - hadoop
Password:
hadoop@ubuntu:~$ cd hadoop-3.3.2/sbin
hadoop@ubuntu:~/hadoop-3.3.2/sbin$ ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
Starting resourcemanager
Starting nodemanagers
hadoop@ubuntu:~/hadoop-3.3.2/sbin$ jps
24451 SecondaryNameNode
24218 DataNode
24650 ResourceManager
25133 Jps
24045 NameNode
24798 NodeManager
```

- Biên dịch 3 file **WordCount.java** ở 3 phiên bản khác nhau (**WordCount_v1**, **WordCount_v2**, **WordCount_v3**) và tạo thành các file **.jar**:

```
hadoop@ubuntu: ~$ ls
Desktop  Downloads  hadoop-3.3.2  hadoop-3.3.2.tar.gz  input  Music  Pictures  Public  stop_words.txt  Templates  tmpdata  Videos  WordCount
hadoop@ubuntu: ~$ mkdir WordCount_v1
hadoop@ubuntu: ~$ mkdir WordCount_v2
hadoop@ubuntu: ~$ mkdir WordCount_v3
hadoop@ubuntu: ~$ ls
Desktop  Documents  hadoop-3.3.2  input  Pictures  stop_words.txt  tmpdata  WordCount  WordCount_v1  WordCount_v2
hadoop@ubuntu: ~$ javac /home/hadoop/WordCount_v1/WordCount.java -cp $(hadoop classpath) -d WordCount_v1
hadoop@ubuntu: ~$ jar -cvf WordCount1.jar -C WordCount_v1 .
added manifest
adding: org/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/WordCount$Map.class(ln = 2165) (out= 955)(deflated 55%)
adding: org/nyorg/WordCount.class(ln = 1976) (out= 989)(deflated 49%)
adding: org/nyorg/WordCount$Reduce.class(ln = 1647) (out= 692)(deflated 57%)
adding: WordCount.java(ln = 9837) (out= 3125)(deflated 65%)
hadoop@ubuntu: ~$ ls
Desktop  Documents  hadoop-3.3.2  input  Pictures  stop_words.txt  tmpdata  WordCount  WordCount_v1  WordCount_v2
hadoop@ubuntu: ~$ javac /home/hadoop/WordCount_v2/WordCount.java -cp $(hadoop classpath) -d WordCount_v2
hadoop@ubuntu: ~$ jar -cvf WordCount2.jar -C WordCount_v2 .
added manifest
adding: org/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/WordCount$Map.class(ln = 2740) (out= 1224)(deflated 55%)
adding: org/nyorg/WordCount.class(ln = 2015) (out= 1083)(deflated 50%)
adding: org/nyorg/WordCount$Reduce.class(ln = 1647) (out= 692)(deflated 57%)
adding: WordCount.java(ln = 4254) (out= 1600)(deflated 62%)
hadoop@ubuntu: ~$ javac /home/hadoop/WordCount_v3/WordCount.java -cp $(hadoop classpath) -d WordCount_v3
hadoop@ubuntu: ~$ jar -cvf WordCount3.jar -C WordCount_v3 .
added manifest
adding: org/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/(ln = 0) (out= 0)(stored 0%)
adding: org/nyorg/WordCount$Map.class(ln = 4559) (out= 2202)(deflated 51%)
adding: org/nyorg/WordCount.class(ln = 2755) (out= 1387)(deflated 49%)
adding: org/nyorg/WordCount$Reduce.class(ln = 1647) (out= 694)(deflated 57%)
adding: WordCount.java(ln = 7208) (out= 2664)(deflated 63%)
hadoop@ubuntu: ~$ ls
Desktop  Documents  hadoop-3.3.2  input  Pictures  stop_words.txt  tmpdata  WordCount  WordCount1.jar  WordCount_v1  WordCount_v2
hadoop@ubuntu: ~$
```

- Tiến hành put 3 file dữ liệu “file01.txt”, “file02.txt” và “file03.txt” với dung lượng lần lượt là 16MB, 20MB và 95MB, nội dung là các cuốn sách được copy liên tiếp ngẫu nhiên từ trang [Free eBooks | Project Gutenberg](#) lên HDFS:



- Tiến hành chạy WordCount.java phiên bản 1 với cùng lúc 3 file dữ liệu và sau đó chạy lần lượt từng file dữ liệu (trong video demo):

```
hadoop@ubuntu: ~$ hadoop fs -put /home/duy/Downloads/file*.txt /user/hadoop/input
hadoop@ubuntu: ~$ hadoop jar WordCount1.jar org.myorg.WordCount /user/hadoop/input /user/hadoop/output/test1
2022-04-21 11:12:19,620 INFO client.DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-21 11:12:20,436 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1650513605518_0001
2022-04-21 11:12:20,738 INFO Input.FileInputFormat: Total input files to process : 3
2022-04-21 11:12:21,253 INFO mapreduce.JobSubmitter: number of splits:3
2022-04-21 11:12:21,417 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1650513605518_0001
2022-04-21 11:12:21,417 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-21 11:12:21,795 INFO conf.Configuration: resource-types.xml not found
2022-04-21 11:12:21,795 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-21 11:12:22,696 INFO impl.YarnClientImpl: Submitted application application_1650513605518_0001
2022-04-21 11:12:22,805 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1650513605518_0001/
2022-04-21 11:12:22,807 INFO mapreduce.Job: Running job: job_1650513605518_0001
2022-04-21 11:12:32,130 INFO mapreduce.Job: Job job_1650513605518_0001 running in uber mode : false
2022-04-21 11:12:32,133 INFO mapreduce.Job: map 0% reduce 0%
2022-04-21 11:12:52,403 INFO mapreduce.Job: map 47% reduce 0%
```

```
hadoop@ubuntu: ~
duy@ubuntu: ~/Downloads

hadoop@ubuntu:~$ hadoop jar WordCount1.jar org.myorg.WordCount /user/hadoop/input/file01.txt /user/hadoop/output/test
2022-04-21 11:15:39,472 INFO client.DefaultHARFalloverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-21 11:15:39,883 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1650513605518_0002
2022-04-21 11:15:40,025 INFO input.FileInputFormat: Total input files to process : 1
2022-04-21 11:15:40,157 INFO mapreduce.JobSubmitter: number of splits:1
2022-04-21 11:15:40,265 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1650513605518_0002
2022-04-21 11:15:40,265 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-21 11:15:40,429 INFO conf.Configuration: resource-types.xml not found
2022-04-21 11:15:40,429 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-21 11:15:40,489 INFO impl.VarnClientImpl: Submitted application application_1650513605518_0002
2022-04-21 11:15:40,525 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1650513605518_0002/
2022-04-21 11:15:40,526 INFO mapreduce.Job: Running job: job_1650513605518_0002
2022-04-21 11:15:46,623 INFO mapreduce.Job: Job job_1650513605518_0002 running in uber mode : false
2022-04-21 11:15:46,627 INFO mapreduce.Job: map 0% reduce 0%
2022-04-21 11:15:57,758 INFO mapreduce.Job: map 100% reduce 0%
2022-04-21 11:16:03,808 INFO mapreduce.Job: map 100% reduce 100%
```

- Tiến hành chạy WordCount.java phiên bản 2 với cùng lúc 3 file dữ liệu và sau đó chạy lần lượt từng file dữ liệu (trong video demo):

```
hadoop@ubuntu: ~
duy@ubuntu: ~/Downloads

hadoop@ubuntu:~$ hadoop jar WordCount2.jar org.myorg.WordCount /user/hadoop/input /user/hadoop/output/test4
2022-04-21 11:22:17,328 INFO client.DefaultHARFalloverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-21 11:22:17,652 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1650513605518_0005
2022-04-21 11:22:17,855 INFO input.FileInputFormat: Total input files to process : 3
2022-04-21 11:22:18,317 INFO mapreduce.JobSubmitter: number of splits:3
2022-04-21 11:22:18,860 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1650513605518_0005
2022-04-21 11:22:18,860 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-21 11:22:19,027 INFO conf.Configuration: resource-types.xml not found
2022-04-21 11:22:19,028 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-21 11:22:19,087 INFO impl.VarnClientImpl: Submitted application application_1650513605518_0005
2022-04-21 11:22:19,126 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1650513605518_0005/
2022-04-21 11:22:19,127 INFO mapreduce.Job: Running job: job_1650513605518_0005
2022-04-21 11:22:25,228 INFO mapreduce.Job: Job job_1650513605518_0005 running in uber mode : false
2022-04-21 11:22:25,236 INFO mapreduce.Job: map 0% reduce 0%
2022-04-21 11:22:43,417 INFO mapreduce.Job: map 50% reduce 0%
2022-04-21 11:22:44,483 INFO mapreduce.Job: map 61% reduce 0%
2022-04-21 11:22:46,499 INFO mapreduce.Job: map 72% reduce 0%
2022-04-21 11:22:49,522 INFO mapreduce.Job: map 76% reduce 0%
2022-04-21 11:22:55,556 INFO mapreduce.Job: map 79% reduce 0%
2022-04-21 11:23:01,587 INFO mapreduce.Job: map 84% reduce 0%
2022-04-21 11:23:03,604 INFO mapreduce.Job: map 84% reduce 22%
2022-04-21 11:23:07,626 INFO mapreduce.Job: map 89% reduce 22%
2022-04-21 11:23:11,652 INFO mapreduce.Job: map 100% reduce 22%
2022-04-21 11:23:12,665 INFO mapreduce.Job: map 100% reduce 100%
```

- Tiến hành chạy WordCount.java phiên bản 3 với cùng lúc 3 file dữ liệu và bỏ qua các từ trong file “stop_words.txt”:

```
hadoop@ubuntu: ~
duy@ubuntu: ~

hadoop@ubuntu:~$ hadoop fs -rm -r -f /user/hadoop/stop_words.txt
Deleted /user/hadoop/stop_words.txt
hadoop@ubuntu:~$ hadoop fs -put /home/hadoop/stop_words.txt /user/hadoop
hadoop@ubuntu:~$ hadoop jar WordCount3.jar org.myorg.WordCount /user/hadoop/input /user/hadoop/output/test8 -skip /user/hadoop/stop_words.txt
2022-04-21 11:33:02,149 INFO myorg.WordCount: Added file to the distributed cache: /user/hadoop/stop_words.txt
2022-04-21 11:33:02,648 INFO client.DefaultHARFalloverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-21 11:33:03,978 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1650513605518_0009
2022-04-21 11:33:03,193 INFO input.FileInputFormat: Total input files to process : 3
2022-04-21 11:33:03,238 INFO mapreduce.JobSubmitter: number of splits:3
2022-04-21 11:33:03,339 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1650513605518_0009
2022-04-21 11:33:03,339 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-21 11:33:03,471 INFO conf.Configuration: resource-types.xml not found
2022-04-21 11:33:03,471 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-21 11:33:03,537 INFO impl.VarnClientImpl: Submitted application application_1650513605518_0009
2022-04-21 11:33:03,567 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1650513605518_0009/
2022-04-21 11:33:03,568 INFO mapreduce.Job: Running job: job_1650513605518_0009
2022-04-21 11:33:09,659 INFO mapreduce.Job: Job job_1650513605518_0009 running in uber mode : false
2022-04-21 11:33:09,662 INFO mapreduce.Job: map 0% reduce 0%
2022-04-21 11:33:27,918 INFO mapreduce.Job: map 51% reduce 0%
2022-04-21 11:33:28,960 INFO mapreduce.Job: map 62% reduce 0%
2022-04-21 11:33:29,971 INFO mapreduce.Job: map 73% reduce 0%
2022-04-21 11:33:33,994 INFO mapreduce.Job: map 75% reduce 0%
2022-04-21 11:33:40,036 INFO mapreduce.Job: map 78% reduce 0%
2022-04-21 11:33:46,063 INFO mapreduce.Job: map 83% reduce 0%
2022-04-21 11:33:48,075 INFO mapreduce.Job: map 83% reduce 22%
2022-04-21 11:33:52,102 INFO mapreduce.Job: map 88% reduce 22%
2022-04-21 11:33:56,133 INFO mapreduce.Job: map 100% reduce 22%
```

- Kết quả thu được sau khi chạy WordCount.java ở phiên bản 1:

```

hadoop@ubuntu: ~
duy@ubuntu: ~/Downloads

zookeeper_mt 1
zookeeper_st 1
zoological 25
zoologically 1
zoologique 2
zoologist 3
zoologists 7
zoology 17
zoom 12
zoomed 53
zooming 26
zoophyte 1
zoophytes 2
zosterops 1
zoological 1
zucchini 1
zxld 3
zygomatic 27
zygomatice 4
zygomatikus 3
zygema 1
zéro 1
zôlin 1
zôon 1
{ 385
{ " 4
{ } 2
{" 18
{( 2
{: 1
{ \ 1
{ }, 1
| 2381
| 12
| $ 1
| ( 5
| - 11
| ... 9
| : 1
| [ 25
| \ 5
| {( 1
| 7

```

- Kết quả thu được sau khi chạy WordCount.java ở phiên bản 2:

```

hadoop@ubuntu: ~
duy@ubuntu: ~/Downloads

zustands 4
zustandsbild 2
zutreffen 2
zutreffend 2
zutreffende 2
zutritt 2
zuviel 6
zuwachs 2
zuwenden 2
zuwenig 2
zuyder 2
zuydersee 2
zuzuf 2
zwaardenaker 24
zwaardenakers 2
zwang 2
zwar 12
zweck 2
zwecke 2
zwecken 2
zweckn 10
zwet 28
zweifache 2
zweifelh 2
zweite 2
zweittens 2
zweiter 2
zwetschen 6
zwey 4
zweiback 8
zwischen 28
zwischenstufen 2
zxld 6
zy 8
zyg 26
zygomatic 56
zygomatice 18
zygomatikus 14
zynotic 2
zynga 2
zyxomma 2
hadoop@ubuntu:~$

```

- Kết quả thu được sau khi chạy WordCount.java ở phiên bản 3 và bỏ qua các từ có trong file **stop_words.txt**:

So sánh kết quả với hình ảnh của WordCount_v2, trong file stop_words tại em đã thêm các từ như zyxomma, zynga, zymotic,... để có cái nhìn trực quan hơn về kết quả đã thực hiện được ở WordCount_v3 này.

```

zusammenhangs 2
zusammenzuhalten 2
zusprechen 2
zuspricht 2
zust 26
zustand 16
zustands 4
zustandsbild 2
zutreffen 2
zutreffende 2
zutreffende 2
zutritts 2
zuviel 6
zuwachs 2
zuwenden 2
zuwenig 2
zuyder 2
zuydersee 2
zuzuf 2
zwaardenaker 24
zwaardenakers 2
zwang 2
zwar 12
zweck 2
zwecke 2
zwecken 2
zweckn 10
zwei 28
zweifache 2
zweifelh 2
zweite 2
zweitens 2
zweiter 2
zwetschen 6
zwey 4
zwieback 8
zwischen 28
zwischenstufen 2
zxid 6
zy 8
zyg 26

```

2. Level 2

a) Trường hợp phân biệt hoa thường. File name **WordCount_v1.py**.

```

1  #!/usr/bin/env python3
2  from mrjob.job import MRJob
3  import re
4
5  WORD_RE = re.compile(r"[\w']+")
6
7
8  class MRWordFreqCount(MRJob):
9
10     def mapper(self, _, line):
11         for word in WORD_RE.findall(line):
12             yield (word, 1)
13
14     def combiner(self, word, counts):
15         yield (word, sum(counts))
16
17     def reducer(self, word, counts):
18         yield (word, sum(counts))
19
20
21 if __name__ == '__main__':
22     MRWordFreqCount.run()

```

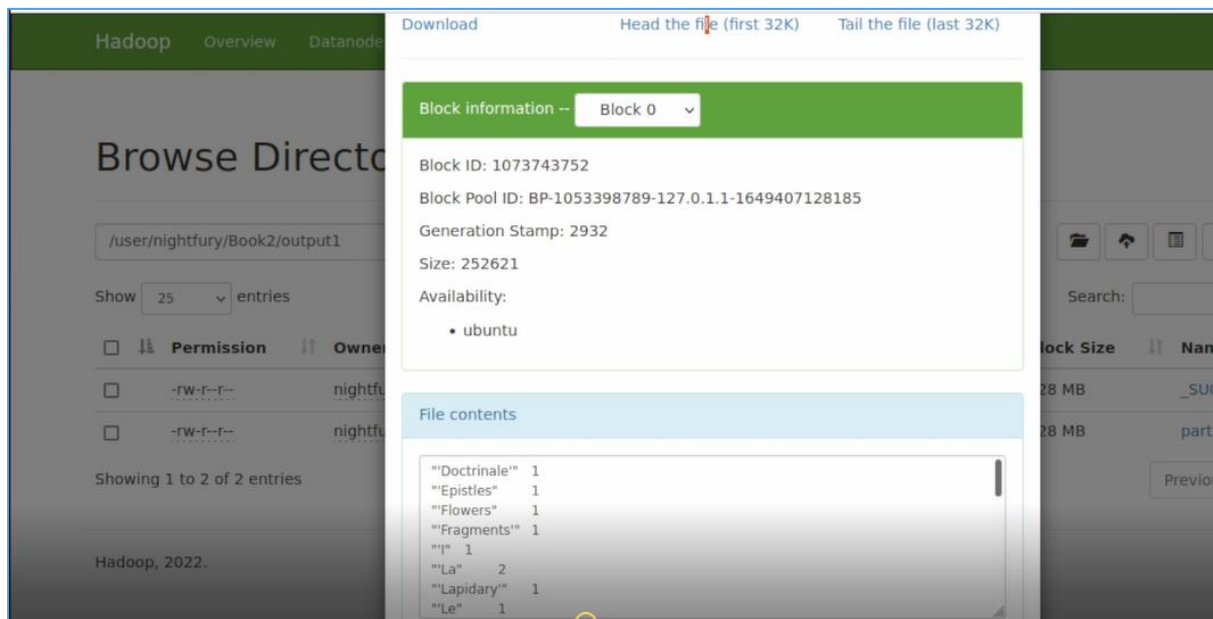
Chạy thực thi file **WordCount_v1.py** với trường hợp phân biệt chữ hoa thường.

```

nightfury@ubuntu: ~/python
nightfury@ubuntu:~$ jps
113685 ResourceManager
113401 SecondaryNameNode
113081 NameNode
113210 DataNode
113821 NodeManager
116510 Jps
nightfury@ubuntu:~$ cd python/
nightfury@ubuntu:~/python$ python3 test1.py -r hadoop --hadoop-streaming-jar /home/nightfury/hadoop-3.3.2/share/hadoop/tools/lib/hadoop-streaming-3.3.2.jar input/book1.txt --output Book1/output1

```

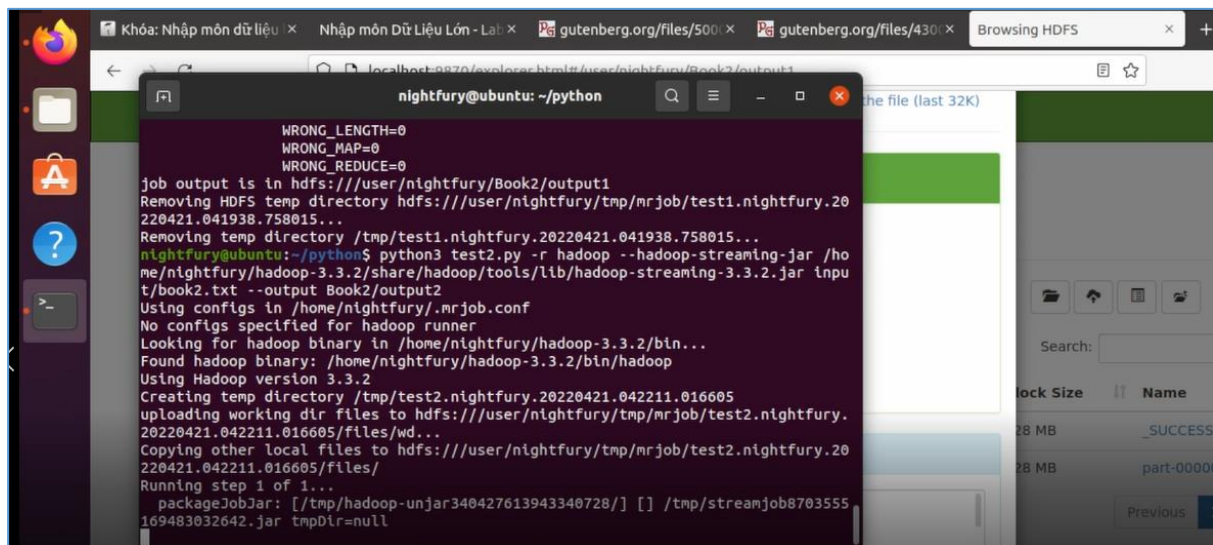

Kết quả thực thi file chương trình có phân biệt hoa thường.



b) Trường hợp không phân biệt hoa thường. File name **WordCount_v2.py**.

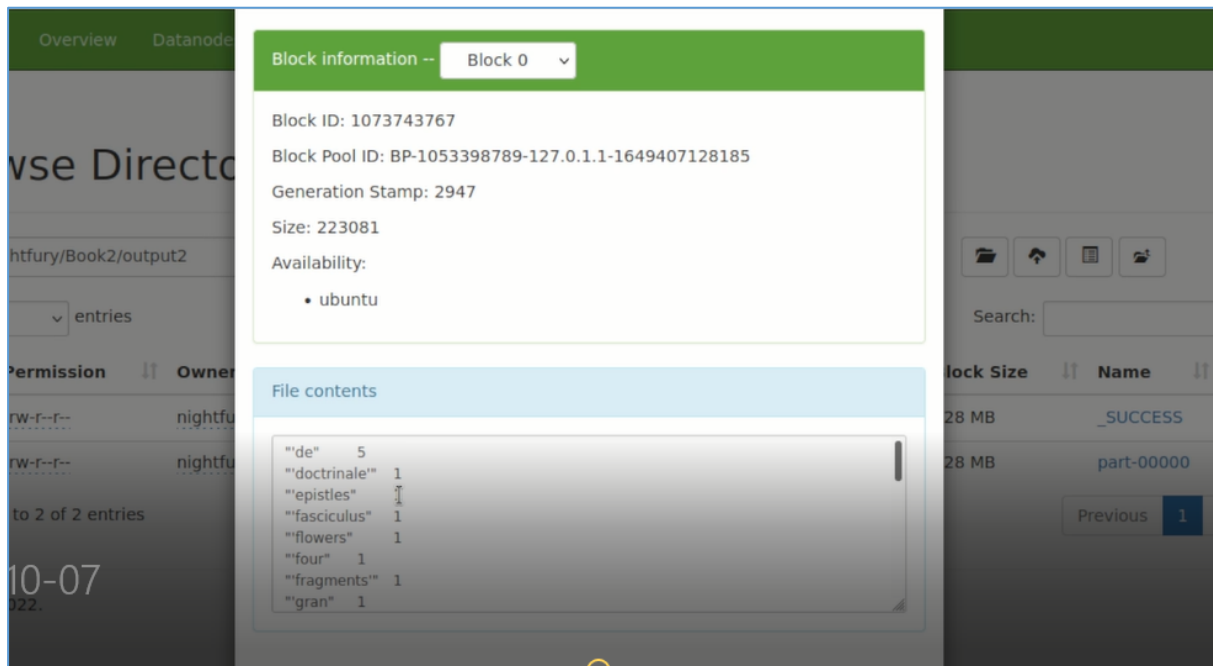
```
1  #!/usr/bin/env python3
2  from mrjob.job import MRJob
3  import re
4
5  WORD_RE = re.compile(r"[\w']+")
6
7
8  class MRWordFreqCount(MRJob):
9
10     def mapper(self, _, line):
11         for word in WORD_RE.findall(line):
12             yield (word, 1)
13
14     def combiner(self, word, counts):
15         yield (word, sum(counts))
16
17     def reducer(self, word, counts):
18         yield (word, sum(counts))
19
20
21 if __name__ == '__main__':
22     MRWordFreqCount.run()
```

Chạy thực thi file WordCount_v2.py không phân biệt hoa thường.



```
nightfury@ubuntu: ~/python
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
job output is in hdfs:///user/nightfury/Book2/output1
Removing HDFS temp directory hdfs:///user/nightfury/tmp/mrjob/test1.nightfury.20220421.041938.758015...
Removing temp directory /tmp/test1.nightfury.20220421.041938.758015...
nightfury@ubuntu:~/python$ python3 test2.py -r hadoop --hadoop-streaming-jar /home/nightfury/hadoop-3.3.2/share/hadoop/tools/lib/hadoop-streaming-3.3.2.jar input/book2.txt --output Book2/output2
Using configs in /home/nightfury/.mrjob.conf
No configs specified for hadoop runner
Looking for hadoop binary in /home/nightfury/hadoop-3.3.2/bin...
Found hadoop binary: /home/nightfury/hadoop-3.3.2/bin/hadoop
Using Hadoop version 3.3.2
Creating temp directory /tmp/test2.nightfury.20220421.042211.016605
uploading working dir files to hdfs:///user/nightfury/tmp/mrjob/test2.nightfury.20220421.042211.016605/files/wd...
Copying other local files to hdfs:///user/nightfury/tmp/mrjob/test2.nightfury.20220421.042211.016605/files/
Running step 1 of 1...
packageJobJar: [/tmp/hadoop-unjar340427613943340728/] [] /tmp/streamjob8703555169483032642.jar tmpDir=null
```

Kết quả thực thi.



Block information -- Block 0

Block ID: 1073743767
Block Pool ID: BP-1053398789-127.0.1.1-1649407128185
Generation Stamp: 2947
Size: 223081
Availability:
• ubuntu

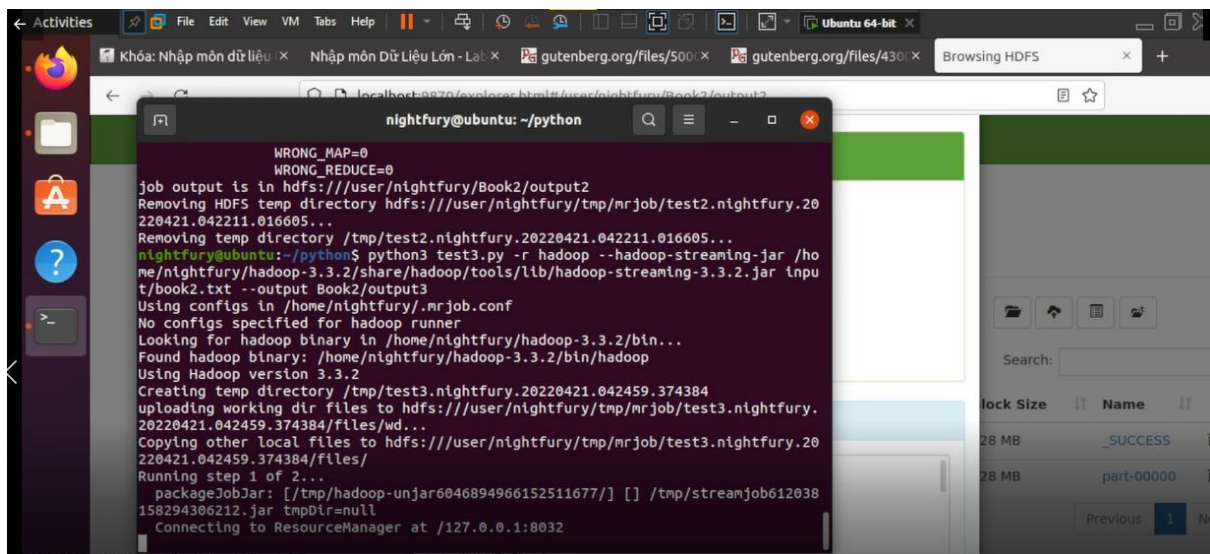
File contents

```
"de" 5
"doctrinale" 1
"epistles" 1
"fasciculus" 1
"flowers" 1
"four" 1
"fragments" 1
"gran" 1
```

- c) Tìm từ xuất hiện nhiều nhất trong tài liệu (không phân biệt hoa thường). File name **WordCount_v3.py**.

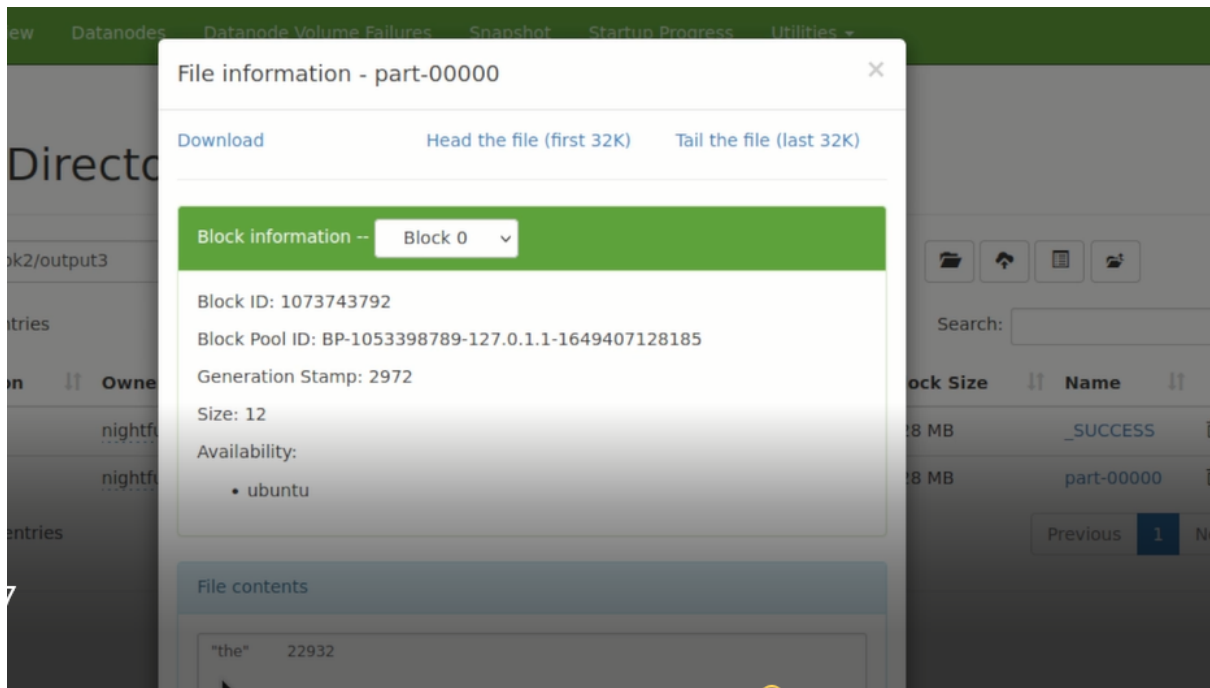
```
1 # import more itertools
2 from mrjob.job import MRJob
3 from mrjob.step import MRStep
4 import re
5
6 WORD_REGEX = re.compile(r"[lw]+")
7 class MRMaxFreq(MRJob):
8     def steps(self):
9         return [
10             MRStep(mapper=self.mapper,
11                   reducer=self.reducer),
12             MRStep(mapper=self.mapper_post,
13                   reducer=self.reducer_post)
14         ]
15     def mapper(self, _, line):
16         for word in WORD_REGEX.findall(line):
17             yield (word.lower(), 1)
18
19     def reducer(self, word, counts):
20         yield (word, sum(counts))
21 # keys: None, values: (word, word_count)
22     def mapper_post(self, word, word_count):
23         yield (None, (word, word_count))
24
25 # sort list of (word, word_count) by word_count
26     def reducer_post(self, _, word_count_pairs):
27
28         from operator import itemgetter
29         yield(max(word_count_pairs, key=itemgetter(1)))
30
31 if __name__ == "__main__":
32     MRMaxFreq().run()
```

Chạy thực thi file **WordCount_v3.py** chứa mã nguồn tìm từ xuất hiện nhiều nhất.



```
nightfury@ubuntu: ~/python
WRONG_MAP=0
WRONG_REDUCE=0
job output is in hdfs:///user/nightfury/Book2/output2
Removing HDFS temp directory hdfs:///user/nightfury/tmp/mrjob/test2.nightfury.20
220421.042211.016605...
Removing temp directory /tmp/test2.nightfury.20220421.042211.016605...
nightfury@ubuntu:~/python$ python3 test3.py -r hadoop --hadoop-streaming-jar /ho
me/nightfury/hadoop-3.3.2/share/hadoop/tools/lib/hadoop-streaming-3.3.2.jar inpu
t/book2.txt --output Book2/output3
Using configs in /home/nightfury/.mrjob.conf
No configs specified for hadoop runner
Looking for hadoop binary in /home/nightfury/hadoop-3.3.2/bin...
Found hadoop binary: /home/nightfury/hadoop-3.3.2/bin/hadoop
Using Hadoop version 3.3.2
Creating temp directory /tmp/test3.nightfury.20220421.042459.374384
uploading working dir files to hdfs:///user/nightfury/tmp/mrjob/test3.nightfury.
20220421.042459.374384/files/wd...
Copying other local files to hdfs:///user/nightfury/tmp/mrjob/test3.nightfury.20
220421.042459.374384/files/
Running step 1 of 2...
packageJobJar: [/tmp/hadoop-unjar6046894966152511677/] [] /tmp/streamjob612038
158294306212.jar tmpDir=null
Connecting to ResourceManager at /127.0.0.1:8032
```


Kết quả thực thi file tìm từ xuất hiện nhiều nhất.



3. Link video minh chứng:

Level 1: <https://www.youtube.com/watch?v=rvDGJee9jg0>

Level 2: https://www.youtube.com/watch?v=ggc_igysJPo

III. Tài liệu tham khảo

[1] Slide lý thuyết phần MapReduce của môn Nhập môn Dữ liệu lớn.

[2]

[https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_overview.htm](https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_overview.html)

1