

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



## BÀI TẬP 2

### Nhập môn dữ liệu lớn

Sinh viên thực hiện: 21127229 - Dương Trường Bình

Giảng viên hướng dẫn: Lê Ngọc Thành  
Nguyễn Ngọc Thảo  
Đỗ Trọng Lễ  
Bùi Huỳnh Trung Nam

Lớp: 21KHDL

# Mục lục

1	Bài tập 1 . . . . .	2
1.1	Ý tưởng thuật toán với Spark Structured API . . . . .	2
1.2	Mã nguồn Python . . . . .	2
1.3	Cách chạy chương trình trên HDFS . . . . .	2
1.4	Minh chứng chạy chương trình . . . . .	3
2	Bài tập 2 . . . . .	3
2.1	Ý tưởng thuật toán với Spark Structured API . . . . .	3
2.2	Mã nguồn Python . . . . .	4
2.3	Cách chạy chương trình trên HDFS . . . . .	4
2.4	Minh chứng chạy chương trình . . . . .	4
3	Bài tập 3 . . . . .	5
3.1	Ý tưởng thuật toán với Spark Structured API . . . . .	5
3.2	Mã nguồn Python . . . . .	6
3.3	Cách chạy chương trình trên HDFS . . . . .	6
3.4	Minh chứng chạy chương trình . . . . .	7
4	Bài tập 4 . . . . .	8
4.1	Ý tưởng thuật toán với Spark Structured API . . . . .	8
4.2	Mã nguồn Python . . . . .	9
4.3	Cách chạy chương trình trên HDFS . . . . .	9
4.4	Minh chứng chạy chương trình . . . . .	9
	Tài liệu tham khảo . . . . .	11

# 1 Bài tập 1

Xét tập dữ liệu ratings (`u.data`) trong MovieLens, thực hiện chương trình thống kê số lượng người bình chọn ở mỗi mức.

## 1.1 Ý tưởng thuật toán với Spark Structured API

- Sử dụng Spark DataFrame để đọc dữ liệu từ file `u.data`.
- Tạo một DataFrame với các cột tương ứng là `user_id`, `item_id`, `rating`, và `timestamp`.
- Thực hiện nhóm các dữ liệu theo cột `rating`, sau đó tính tổng số lần xuất hiện của mỗi mức `rating`.

## 1.2 Mã nguồn Python

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import count
3
4 # Tạo SparkSession
5 spark = SparkSession.builder \
6     .appName("Movie Ratings Count") \
7     .getOrCreate()
8
9 # Đọc dữ liệu từ file u.data và tạo DataFrame
10 df = spark.read.csv("hdfs:///path/to/u.data", sep='\t', inferSchema=True)
11 df = df.withColumnRenamed("_c0", "user_id") \
12     .withColumnRenamed("_c1", "item_id") \
13     .withColumnRenamed("_c2", "rating") \
14     .withColumnRenamed("_c3", "timestamp")
15
16 # Thực hiện thống kê số lượt bình chọn ở mỗi mức
17 rating_count_df = df.groupBy("rating").agg(count("rating").alias("count"))
18
19 # Hiển thị kết quả
20 rating_count_df.show()
```

## 1.3 Cách chạy chương trình trên HDFS

1. Đảm bảo dữ liệu `u.data` đã được tải lên HDFS.
2. Sử dụng `spark-submit` để chạy chương trình:

```
spark-submit movie_ratings_count.py
```

Sau khi chạy xong, kết quả sẽ được lưu ra HDFS với đường dẫn đã chỉ định.

## 1.4 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ thống Spark với kết quả như sau:

```

>_pws_ Ex02 1ms
>> python .\movie_ratings_count.py
24/09/06 15:01:15 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.h
doop/WindowsProblems
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:01:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
+-----+
|rating|count|
+-----+
|    1| 6110|
|    2|11370|
|    3|27145|
|    4|34174|
|    5|21201|
+-----+

SUCCESS: The process with PID 10832 (child process of PID 26096) has been terminated.
SUCCESS: The process with PID 26096 (child process of PID 15736) has been terminated.
SUCCESS: The process with PID 15736 (child process of PID 23828) has been terminated.

```

Hình 1.4.1: Kết quả chương trình đếm số lượt bình chọn cho mỗi mức rating.

## 2 Bài tập 2

Xét tập dữ liệu ratings (`u.data`) trong MovieLens, thực hiện chương trình sắp xếp các phim theo số lượt bình chọn.

### 2.1 Ý tưởng thuật toán với Spark Structured API

- Sử dụng Spark DataFrame để đọc dữ liệu từ file `u.data`.
- Tạo một DataFrame với các cột tương ứng là `user_id`, `item_id`, `rating`, và `timestamp`.
- Thực hiện nhóm dữ liệu theo cột `item_id` (mã phim) và đếm tổng số lượt bình chọn cho mỗi phim.
- Sắp xếp kết quả theo tổng số lượt bình chọn từ cao đến thấp.

## 2.2 Mã nguồn Python

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import count
3
4 # Tạo SparkSession
5 spark = SparkSession.builder \
6     .appName("Movie Rating Count") \
7     .getOrCreate()
8
9 # Đọc dữ liệu từ file u.data và tạo DataFrame
10 df = spark.read.csv("hdfs:///path/to/u.data", sep='\t', inferSchema=True)
11 df = df.withColumnRenamed("_c0", "user_id") \
12     .withColumnRenamed("_c1", "item_id") \
13     .withColumnRenamed("_c2", "rating") \
14     .withColumnRenamed("_c3", "timestamp")
15
16 # Thực hiện nhóm theo item_id (mã phim) và đếm số lượt bình chọn
17 movie_count_df = df.groupBy("item_id").agg(count("item_id").alias("count"))
18
19 # Sắp xếp theo tổng số lượt bình chọn (từ cao xuống thấp)
20 sorted_movie_count_df = movie_count_df.orderBy("count", ascending=False)
21
22 # Hiển thị kết quả
23 sorted_movie_count_df.show()
```

## 2.3 Cách chạy chương trình trên HDFS

1. Đảm bảo dữ liệu u.data đã được tải lên HDFS.
2. Sử dụng spark-submit để chạy chương trình:

```
spark-submit movie_count_Ex02.py
```

Sau khi chạy xong, kết quả sẽ được lưu ra HDFS với đường dẫn đã chỉ định.

## 2.4 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ thống Spark với kết quả như sau:

```

>_pwsh Ex02 2ms
>> python .\movie_count_Ex02.py
24/09/06 15:10:40 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.
doop/WindowsProblems
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:10:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
+-----+-----+
|item_id|count|
+-----+-----+
|    50| 583|
|   258| 509|
|   100| 508|
|   181| 507|
|   294| 485|
|   286| 481|
|   288| 478|
|     1| 452|
|   300| 431|
|   121| 429|
|   174| 420|
|   127| 413|
|    56| 394|
|     7| 392|
|    98| 390|
|   237| 384|
|   117| 378|
|   172| 367|
|   222| 365|
|   204| 350|
+-----+-----+
only showing top 20 rows

SUCCESS: The process with PID 25204 (child process of PID 18424) has been terminated.
SUCCESS: The process with PID 18424 (child process of PID 25980) has been terminated.
SUCCESS: The process with PID 25980 (child process of PID 26840) has been terminated.

```

Hình 2.4.2: Kết quả chương trình sắp xếp các phim theo số lượt bình chọn.

## 3 Bài tập 3

Thực hiện chương trình thống kê số lần xuất hiện của mỗi từ trong tài liệu cho trước.

### 3.1 Ý tưởng thuật toán với Spark Structured API

- Sử dụng Spark DataFrame để đọc dữ liệu từ file văn bản.
- Tách từng từ trong tài liệu, sau đó thực hiện thống kê số lần xuất hiện của mỗi từ.
- Thực hiện 2 yêu cầu:
  - **Câu a:** Thống kê số lần xuất hiện của mỗi từ, phân biệt hoa thường.
  - **Câu b:** Thống kê số lần xuất hiện của mỗi từ, không phân biệt hoa thường.

## 3.2 Mã nguồn Python

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import split, explode, col, lower, count
3
4 # Tạo SparkSession
5 spark = SparkSession.builder \
6     .appName("Word Count") \
7     .getOrCreate()
8
9 # Đọc dữ liệu từ file văn bản và tạo DataFrame
10 df = spark.read.text("../pg20417.txt")
11
12 # Tách từng từ trong tài liệu (Câu a: phân biệt hoa thường)
13 df_exploded_a = df.withColumn("word", explode(split(col("value"), "\\s+")))
14
15 # Thống kê số lần xuất hiện của mỗi từ (Câu a)
16 word_count_a = df_exploded_a.groupBy("word").agg(
17     count("word").alias("count")).orderBy("count", ascending=False)
18
19 # Hiển thị kết quả Câu a
20 word_count_a.show(10)
21
22 # Tách từng từ và chuyển về chữ thường (Câu b: không phân biệt hoa thường)
23 df_exploded_b = df.withColumn(
24     "word", explode(split(lower(col("value")), "\\s+")))
25
26 # Thống kê số lần xuất hiện của mỗi từ (Câu b)
27 word_count_b = df_exploded_b.groupBy("word").agg(
28     count("word").alias("count")).orderBy("count", ascending=False)
29
30 # Hiển thị kết quả Câu b
31 word_count_b.show(10)
```

## 3.3 Cách chạy chương trình trên HDFS

1. Đảm bảo dữ liệu văn bản đã được tải lên HDFS.
2. Sử dụng `spark-submit` để chạy chương trình:

```
spark-submit word_count.py
```

Sau khi chạy xong, kết quả sẽ được lưu ra HDFS với đường dẫn đã chỉ định cho cả hai trường hợp.

### 3.4 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ thống Spark với kết quả như sau:

```

python .\word_count_Ex03.py
24/09/06 15:24:13 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.h
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:24:13 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

+-----+
|word|count|
+-----+
| the| 7916|
| of| 5427|
| | 2920|
| and| 2757|
| a| 2422|
| to| 2170|
| is| 2064|
| in| 2052|
| that| 1272|
| are| 925|
+-----+
only showing top 10 rows

+-----+
|word|count|
+-----+
| the| 9167|
| of| 5756|
| | 2920|
| and| 2987|
| a| 2727|
| in| 2351|
| to| 2235|
| is| 2094|
| it| 1322|
| that| 1311|
+-----+
only showing top 10 rows

SUCCESS: The process with PID 25456 (child process of PID 18876) has been terminated.
SUCCESS: The process with PID 18876 (child process of PID 11500) has been terminated.
SUCCESS: The process with PID 11500 (child process of PID 22140) has been terminated.

```

Hình 3.4.3: Kết quả chương trình thống kê số lần xuất hiện của mỗi từ (phân biệt và không phân biệt hoa thường) của file pg20417.txt

```

python .\word_count_Ex03.py
24/09/06 15:25:58 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.h
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:25:58 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

+-----+
|word|count|
+-----+
| the|20680|
| of|10447|
| and| 7633|
| | 7281|
| in| 5260|
| to| 5093|
| is| 4098|
| a| 3825|
| that| 2670|
| which| 2392|
+-----+
only showing top 10 rows

+-----+
|word|count|
+-----+
| the|22807|
| of|11128|
| and| 8309|
| | 7281|
| in| 5623|
| to| 5273|
| a| 4240|
| is| 4139|
| that| 2808|
| it| 2613|
+-----+
only showing top 10 rows

SUCCESS: The process with PID 13708 (child process of PID 27132) has been terminated.
SUCCESS: The process with PID 27132 (child process of PID 8888) has been terminated.
SUCCESS: The process with PID 8888 (child process of PID 25668) has been terminated.

```

Hình 3.4.4: Kết quả chương trình thống kê số lần xuất hiện của mỗi từ (phân biệt và không phân biệt hoa thường) của file pg5000.txt



```

>>> python .\word_count_Ex03.py
24/09/06 15:28:09 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.h
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:28:10 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
+-----+
|word|count|
+-----+
| the|13613|
| of| 8136|
|   | 7927|
| and| 6541|
| a| 5839|
| to| 4788|
| in| 4612|
| his| 3034|
| he| 2712|
| I| 2429|
+-----+
only showing top 10 rows

+-----+
|word|count|
+-----+
| the|14870|
| of| 8221|
|   | 7927|
| and| 7053|
| a| 6599|
| to| 4913|
| in| 4803|
| he| 3620|
| his| 3272|
| with| 2487|
+-----+
only showing top 10 rows

SUCCESS: The process with PID 17692 (child process of PID 11232) has been terminated.
SUCCESS: The process with PID 11232 (child process of PID 11272) has been terminated.
SUCCESS: The process with PID 11272 (child process of PID 2796) has been terminated.

```

Hình 3.4.5: Kết quả chương trình thống kê số lần xuất hiện của mỗi từ (phân biệt và không phân biệt hoa thường) của file pg4300.txt

## 4 Bài tập 4

Thực hiện chương trình tìm từ xuất hiện nhiều nhất trong tài liệu (không phân biệt hoa thường).

### 4.1 Ý tưởng thuật toán với Spark Structured API

- Sử dụng Spark DataFrame để đọc dữ liệu từ file văn bản.
- Tách từng từ trong tài liệu và chuyển tất cả về dạng chữ thường để không phân biệt hoa thường.
- Thực hiện thống kê số lần xuất hiện của mỗi từ.
- Sử dụng hàm `orderBy` để sắp xếp từ theo số lần xuất hiện và lấy từ xuất hiện nhiều nhất.

## 4.2 Mã nguồn Python

```
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import split, explode, col, lower, count
3
4 # Tạo SparkSession
5 spark = SparkSession.builder \
6     .appName("Most Frequent Word") \
7     .getOrCreate()
8
9 # Đọc dữ liệu từ file văn bản và tạo DataFrame
10 df = spark.read.text("hdfs:///path/to/input.txt")
11
12 # Tách từng từ và chuyển về chữ thường
13 df_exploded = df.withColumn("word", explode(
14     split(lower(col("value")), "\\s+")))
15
16 # Thống kê số lần xuất hiện của mỗi từ
17 word_count_df = df_exploded.groupBy("word").agg(count("word").alias("count"))
18
19 # Sắp xếp theo số lần xuất hiện từ cao đến thấp và lấy từ xuất hiện nhiều nhất
20 most_frequent_word = word_count_df.orderBy("count", ascending=False).limit(1)
21
22 # Hiển thị kết quả
23 most_frequent_word.show()
```

## 4.3 Cách chạy chương trình trên HDFS

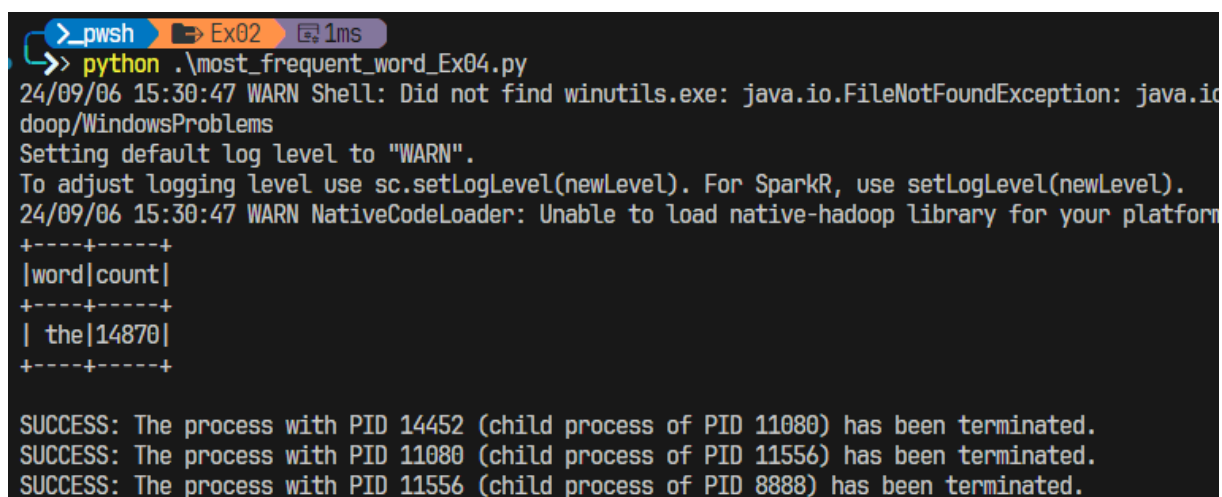
1. Đảm bảo dữ liệu văn bản đã được tải lên HDFS.
2. Sử dụng spark-submit để chạy chương trình:

```
spark-submit most_frequent_word.py
```

Sau khi chạy xong, kết quả sẽ được lưu ra HDFS với đường dẫn đã chỉ định.

## 4.4 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ thống Spark với kết quả như sau:



```
>_pwsh Ex02 1ms
>> python .\most_frequent_word_Ex04.py
24/09/06 15:30:47 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io
doop/WindowsProblems
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/09/06 15:30:47 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform
+----+-----+
|word|count|
+----+-----+
| the|14870|
+----+-----+

SUCCESS: The process with PID 14452 (child process of PID 11080) has been terminated.
SUCCESS: The process with PID 11080 (child process of PID 11556) has been terminated.
SUCCESS: The process with PID 11556 (child process of PID 8888) has been terminated.
```

Hình 4.4.6: Kết quả chương trình tìm từ xuất hiện nhiều nhất trong tài liệu.

# Tài liệu tham khảo

- [1] Slide của thầy Lê Ngọc Thành, *Module 7 - Spark Structured APIs*, Trường Đại học Khoa học Tự nhiên, 2024