

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



BÀI TẬP 1

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 MapReduce	2
1.2	Thuật toán MapReduce	2
1.3	Mã nguồn Python	2
1.4	Cách chạy chương trình	3
1.5	Minh chứng chạy chương trình	3
2	Bài tập 2	3
2.1	Ý tưởng thuật toán MapReduce	3
2.2	Thuật toán MapReduce	4
2.3	Mã nguồn Python	4
2.4	Cách chạy chương trình	5
2.5	Minh chứng chạy chương trình	5
3	Bài tập 3	6
3.1	Ý tưởng thuật toán MapReduce	6
3.2	Phân biệt hoa thường	6
3.3	Thuật toán MapReduce	6
3.4	Mã nguồn Python	6
3.5	Chạy chương trình	7
3.6	Kết quả cho các tệp 4300.txt, 5000.txt và 20417.txt	7
3.7	Minh chứng chạy chương trình	8
4	Bài tập 4	8
4.1	Ý tưởng thuật toán MapReduce	8
4.2	Thuật toán MapReduce	9
4.3	Mã nguồn Python	9
4.4	Cách chạy chương trình	10
4.5	Minh chứng chạy chương trình	10
	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 MapReduce

- **Pha Map:** Mỗi dòng dữ liệu chứa thông tin về một lần đánh giá phim. Từ đó, ta sẽ trích xuất mức đánh giá (rating) và tạo ra các cặp khóa-giá trị với khóa là mức đánh giá và giá trị là 1.

- **Pha Reduce:** Tổng hợp tất cả các cặp khóa-giá trị từ pha Map, tính tổng số lần xuất hiện của mỗi mức đánh giá, và xuất ra kết quả cuối cùng.

1.2 Thuật toán MapReduce

- **Pha Map:** Đọc từng dòng dữ liệu từ tập tin, tách lấy mức đánh giá và xuất ra cặp khóa-giá trị `<rating, 1>`.
- **Pha Reduce:** Gom nhóm các giá trị cùng khóa (mức đánh giá), sau đó tính tổng giá trị cho mỗi khóa (tổng số lượt đánh giá cho mỗi mức).

1.3 Mã nguồn Python

```
1 from mrjob.job import MRJob
2
3 class MRRatingCount(MRJob):
4     def mapper(self, _, line):
5         # Tách dòng dữ liệu theo định dạng 'user_id item_id rating timestamp'
6         _, _, rating, _ = line.split('\t')
7         yield rating, 1
8
9     def reducer(self, rating, counts):
10        # Tính tổng số lượt bình chọn cho mỗi mức đánh giá
11        yield rating, sum(counts)
12
13 if __name__ == '__main__':
14     MRRatingCount.run()
```

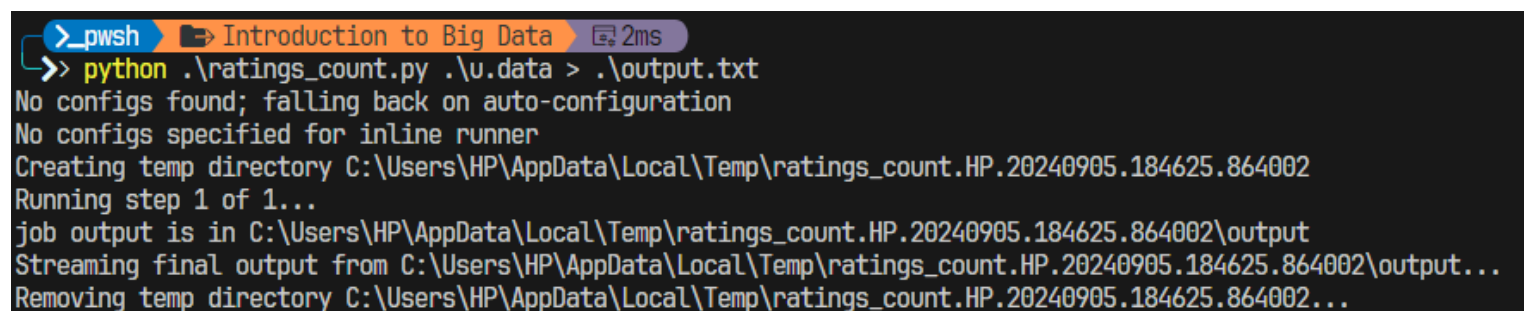
1.4 Cách chạy chương trình

1. Đảm bảo dữ liệu `u.data` đã được tải lên HDFS.
2. Chạy chương trình trên Hadoop với cú pháp sau:

```
python ratings_count.py -r hadoop hdfs:///path/to/u.data
```

1.5 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ điều hành Windows. Kết quả được xuất ra file `output.txt`.



```
>_pwsh Introduction to Big Data 2ms
>> python .\ratings_count.py .\u.data > .\output.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.184625.864002
Running step 1 of 1...
job output is in C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.184625.864002\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.184625.864002\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.184625.864002...
```

File `output.txt`:

```
"1"      6110
"2"      11370
"3"      27145
"4"      34174
"5"      21201
```

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 MapReduce

- **Pha Map:** Mỗi dòng dữ liệu chứa thông tin về một lần đánh giá phim. Từ đó, ta trích xuất ID của bộ phim và tạo ra các cặp khóa-giá trị với khóa là mã phim và giá trị là 1.

- **Pha Reduce:** Tổng hợp các cặp khóa-giá trị từ pha Map, tính tổng số lượt bình chọn cho mỗi bộ phim, và xuất ra kết quả cuối cùng với danh sách các phim và số lượt bình chọn tương ứng.
- **Pha Reduce thứ 2:** Sắp xếp danh sách phim dựa trên tổng số lượt bình chọn, sau đó xuất ra danh sách phim theo thứ tự giảm dần của số lượt bình chọn.

2.2 Thuật toán MapReduce

- **Pha Map:** Đọc từng dòng dữ liệu từ tập tin, tách lấy mã phim và xuất ra cặp khóa-giá trị <movie_id, 1>.
- **Pha Reduce:** Gom nhóm các giá trị cùng khóa (mã phim), sau đó tính tổng giá trị cho mỗi khóa (tổng số lượt bình chọn cho mỗi phim).
- **Pha Reduce thứ 2:** Sắp xếp các phim dựa trên số lượt bình chọn, trả về danh sách phim theo thứ tự giảm dần.

2.3 Mã nguồn Python

```
1 from mrjob.job import MRJob
2 from mrjob.step import MRStep
3
4 class MRMovieRatingsCount(MRJob):
5
6     def steps(self):
7         return [
8             MRStep mapper=self.mapper_get_ratings,
9                   reducer=self.reducer_count_ratings),
10            MRStep(reducer=self.reducer_sort_by_ratings)
11        ]
12
13     def mapper_get_ratings(self, _, line):
14         # Tách dòng dữ liệu theo định dạng 'user_id item_id rating timestamp'
15         _, movie_id, _, _ = line.split('\t')
16         yield movie_id, 1
17
18     def reducer_count_ratings(self, movie_id, counts):
19         # Tính tổng số lượt bình chọn cho mỗi bộ phim
20         yield None, (sum(counts), movie_id)
21
22     def reducer_sort_by_ratings(self, _, movie_counts):
```

```

23     # Sắp xếp phim theo số lượt bình chọn và xuất kết quả
24     for count, movie_id in sorted(movie_counts, reverse=True):
25         yield movie_id, count
26
27 if __name__ == '__main__':
28     MRMovieRatingsCount.run()

```

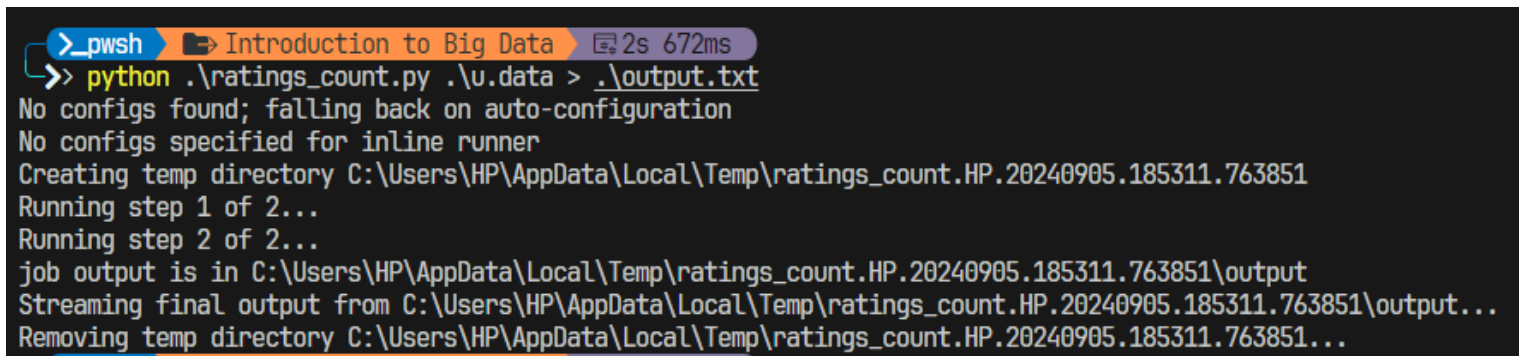
2.4 Cách chạy chương trình

1. Đảm bảo dữ liệu u.data đã được tải lên HDFS.
2. Chạy chương trình trên Hadoop với cú pháp sau:

```
python ratings_count.py -r hadoop hdfs:///path/to/u.data
```

2.5 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ điều hành Windows. Kết quả được xuất ra file output.txt.



```

>_pwsh Introduction to Big Data 2s 672ms
>> python .\ratings_count.py .\u.data > .\output.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.185311.763851
Running step 1 of 2...
Running step 2 of 2...
job output is in C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.185311.763851\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.185311.763851\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\ratings_count.HP.20240905.185311.763851...

```

File output.txt:

```

"50" 583
"258" 509
"100" 508
"181" 507
"294" 485
"286" 481
"288" 478
...

```

3 Bài tập 3

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

3.1 Ý tưởng thuật toán MapReduce

- **Pha Map:** Mỗi dòng trong tài liệu sẽ được tách thành các từ, và mỗi từ sẽ được tạo thành một cặp khóa-giá trị với từ là khóa và giá trị là 1.
- **Pha Reduce:** Gom nhóm các từ giống nhau từ pha Map và tính tổng số lần xuất hiện của mỗi từ. Kết quả cuối cùng sẽ là danh sách các từ và số lần xuất hiện tương ứng.

3.2 Phân biệt hoa thường

- **Câu a:** Trường hợp phân biệt hoa thường. Các từ được tính riêng biệt dựa trên cách viết hoa/thường.
- **Câu b:** Trường hợp không phân biệt hoa thường. Các từ được chuyển về dạng chữ thường trước khi tính số lần xuất hiện.

3.3 Thuật toán MapReduce

- **Pha Map:** Đọc từng dòng dữ liệu từ tập tin, tách các từ và xuất ra cặp khóa-giá trị `<word, 1>`.
- **Pha Reduce:** Gom nhóm các từ giống nhau, sau đó tính tổng số lần xuất hiện của mỗi từ.

3.4 Mã nguồn Python

```
1 from mrjob.job import MRJob
2 import re
3
4 WORD_RE = re.compile(r"[\w']+")
5
6 class MRWordCount(MRJob):
7
8     def mapper(self, _, line):
9         # Tách các từ trong dòng và phân biệt hoa thường
10        for word in WORD_RE.findall(line):
```

```

11         yield word, 1 # Trường hợp phân biệt hoa thường (Câu a)
12         yield word.lower(), 1 # Không phân biệt hoa thường (Câu b)
13
14     def reducer(self, word, counts):
15         # Tính tổng số lần xuất hiện của mỗi từ
16         yield word, sum(counts)
17
18 if __name__ == '__main__':
19     MRWordCount.run()

```

3.5 Chạy chương trình

1. Đảm bảo dữ liệu văn bản đã được tải lên HDFS. 2. Chạy chương trình với cú pháp sau:

```
python wordcount.py -r hadoop hdfs:///path/to/textfile
```

3.6 Kết quả cho các tệp 4300.txt, 5000.txt và 20417.txt

4300.txt	5000.txt	20417.txt
"0" 41	"'" 4	"'" 1
"1" 140	"'_channa_' 1	"'accidental'" 1
"10" 24	"'_codex'" 1	"'irretraceable'" 1
"100" 6	"'_come'" 1	"'possum'" 2
...

3.7 Minh chứng chạy chương trình

```
>_pwsh Introduction to Big Data 2ms
>> python .\word_count.py .\pg20417.txt > 20417.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192751.782946
Running step 1 of 1...
job output is in C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192751.782946\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192751.782946\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192751.782946...

>_pwsh Introduction to Big Data 2s 885ms
>> python .\word_count.py .\pg5000.txt > 5000.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192802.562674
Running step 1 of 1...
job output is in C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192802.562674\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192802.562674\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192802.562674...

>_pwsh Introduction to Big Data 4s 338ms
>> python .\word_count.py .\pg4300.txt > 4300.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192819.051666
Running step 1 of 1...
job output is in C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192819.051666\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192819.051666\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.192819.051666...
```

4 Bài tập 4

Thực hiện chương trình tìm từ xuất hiện nhiều nhất trong tài liệu.

4.1 Ý tưởng thuật toán MapReduce

- **Pha Map:** Mỗi dòng trong tài liệu sẽ được tách thành các từ, và mỗi từ sẽ được tạo thành một cặp khóa-giá trị với từ là khóa và giá trị là 1.
- **Pha Reduce:** Gom nhóm các từ giống nhau từ pha Map và tính tổng số lần xuất hiện của mỗi từ. Kết quả cuối cùng sẽ là danh sách các từ và số lần xuất hiện tương ứng.
- **Pha Reduce thứ 2:** Sử dụng hàm `max()` để tìm từ xuất hiện nhiều nhất trong danh sách đã tổng hợp từ pha Reduce.

4.2 Thuật toán MapReduce

- **Pha Map:** Đọc từng dòng dữ liệu từ tập tin, tách các từ và xuất ra cặp khóa-giá trị `<word, 1>`.
- **Pha Reduce:** Gom nhóm các giá trị cùng khóa (từ), sau đó tính tổng số lần xuất hiện của mỗi từ.
- **Pha Reduce thứ 2:** Tìm từ xuất hiện nhiều nhất trong danh sách các từ và số lần xuất hiện.

4.3 Mã nguồn Python

```
1 from mrjob.job import MRJob
2 from mrjob.step import MRStep
3 import re
4
5 WORD_RE = re.compile(r"[\w']+")
6
7 class MRMostUsedWord(MRJob):
8
9     def steps(self):
10         return [
11             MRStep(mapper=self.mapper_get_words,
12                   reducer=self.reducer_count_words),
13             MRStep(reducer=self.reducer_find_max_word)
14         ]
15
16     def mapper_get_words(self, _, line):
17         # Tách từ trong dòng
18         for word in WORD_RE.findall(line):
19             yield word.lower(), 1
20
21     def reducer_count_words(self, word, counts):
22         # Đếm số lần xuất hiện của từ
23         yield None, (sum(counts), word)
24
25     def reducer_find_max_word(self, _, word_count_pairs):
26         # Tìm từ xuất hiện nhiều nhất
27         yield max(word_count_pairs)
28
29 if __name__ == '__main__':
30     MRMostUsedWord.run()
```

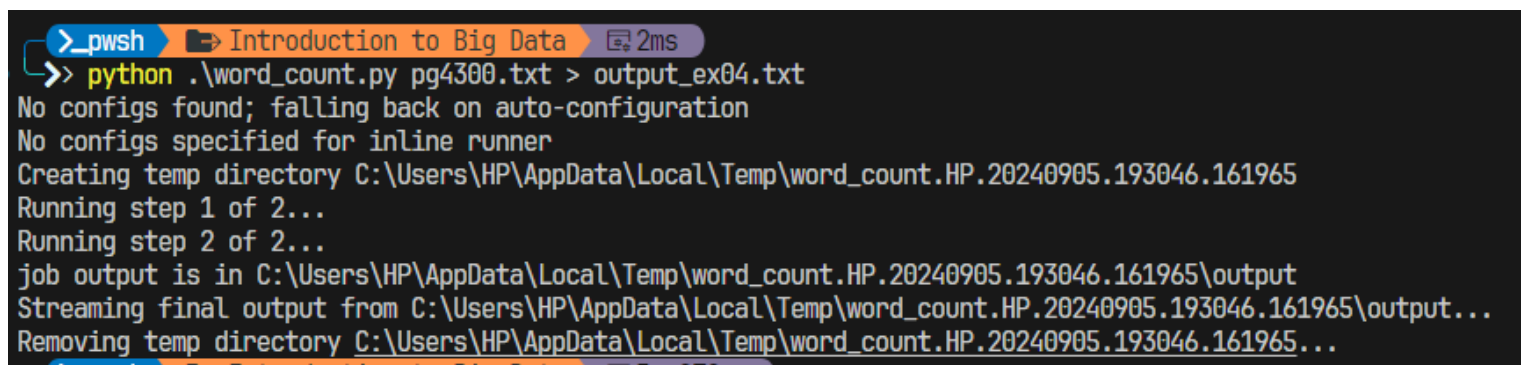
4.4 Cách chạy chương trình

1. Đảm bảo dữ liệu văn bản đã được tải lên HDFS.
2. Chạy chương trình trên Hadoop với cú pháp sau:

```
python most_used_word.py -r hadoop hdfs:///path/to/textfile
```

4.5 Minh chứng chạy chương trình

Chương trình đã được chạy thành công trên hệ điều hành Windows. Kết quả được xuất ra file `output.txt`.



```
>_pwwsh Introduction to Big Data 2ms
>> python .\word_count.py pg4300.txt > output_ex04.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.193046.161965
Running step 1 of 2...
Running step 2 of 2...
job output is in C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.193046.161965\output
Streaming final output from C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.193046.161965\output...
Removing temp directory C:\Users\HP\AppData\Local\Temp\word_count.HP.20240905.193046.161965...
```

File `output.txt` chứa từ xuất hiện nhiều nhất trong tập tin:

```
15092 "the"
```

Tài liệu tham khảo

- [1] Slide của thầy Lê Ngọc Thành, *Module 5 - Hadoop MapReduce*, Trường Đại học Khoa học Tự nhiên, 2024