Lab
Bài Toán Đếm Từ với MapReduce

1. Mô tả

Trong bài lab này, SV thực hiện viết chương trình để đếm từ trong một tập tin và chạy chương trình trên hệ thống MapReduce. Qua đó, phân tích quá trình chạy trên hệ thống, so sánh với việc chạy trên hệ thống thông thường. Rút ra các nhận xét quan trọng.

2. Yêu cầu

SV hoàn thiện các mức yêu cầu sau đây.

2.1. Mức 1

Tài liệu tham khảo [1] ở mục 5 thể hiện ba phiên bản khác nhau của chương trình đếm từ (word count) được viết bằng ngôn ngữ Java sử dụng MapReduce. Cụ thể:

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

SV hoàn thành các yêu cầu sau:

- Viết lại đầy đủ chương trình WordCount đầy đủ ở 3 cấp độ trên.
- Giải thích mã nguồn đã viết (cần ghi rõ những gì thừa kế, những gì thay đổi, bổ sung, v.v…). Liên hệ mã nguồn với các nội dung lý thuyết đã được học.
- Chạy thực thi trên các máy cài đặt Hadoop và thể hiện quá trình chạy.
    - Lưu ý cần thiết kế quá trình thực thi thể hiện việc chạy phân tán trên hệ thống Hadoop (ví dụ như file kích thước lớn, nằm trên trên node khác khau, v.v...).
    - Quay video để minh chứng quá trình thực thi. Trích các screenshot để đưa vào báo cáo. Tất cả các hình cần có đoạn văn mô tả hình đó là gì, mục tiêu đưa vào để nói ý gì.

2.2. Mức 2

2.2.1. Cài đặt PIP, MRJob, Nano

Việc viết bằng ngôn ngữ Java (ngôn ngữ chính được hỗ trợ bởi MapReduce) thường phức tạp. Việc streaming qua ngôn ngữ Python giúp cho việc lập trình trở nên dễ dàng hơn. Có nhiều công cụ để hỗ trợ streaming như Hadoop Streaming, MRJob, Dumbo, Hadoopy. Trong khuôn khổ bài lab này, chúng ta sử dụng thư viện MRJob được phát triển bởi Yelp (Amazon Web Services). Lưu ý, tùy vào môi trường đã cài Hadoop trước đây, SV tìm hiểu để thay đổi cách thức khi không tương thích phiên bản.

SV thực hiện cài đặt thư viện streaming python để có thể sử dụng Python để lập trình Map Reduce. Ví dụ:

- Cài đặt pip: `yum install hadoop-pip`
- Cập nhật pip: `python -m pip install --upgrade pip`
- Cài đặt MRJob: `pip install mrjob==0.5.11`

(Lưu ý mỗi phiên bản MRJob sẽ phù hợp với phiên bản python có trên hệ thống)

2.2.2. Cách chạy MRJob trên Local

Trước khi chạy code thực sự trên HDFS của Hadoop, ta có thể test trên hệ thống local với kích thước dữ liệu nhỏ trước để đảm bảo không lỗi:

Cú pháp: `python yourcode.py data`

Nếu muốn xuất kết quả ra file thay vì ra màn hình ta dùng dấu “>”. Ví dụ:

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

2.2.3. Cách chạy MRJob trên HDFS

Có thể vừa tải dữ liệu lên HDFS vừa chạy code thực thi trên đó.

Cú pháp:

```
python yourcode.py -r hadoop --hadoop-streaming-jar
/usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
localData
```

Hoặc có thể chạy trên dữ liệu đã được upload sẵn trên HDFS. Để làm được cách này ta cần biết hostname và port để kết nối để HDFS.

Việc xem hostname có thể thực hiện một trong hai cách:

- Cách 1: xem trong tập tin /etc/hadoop/conf/hdfs-site.xml, thẻ có name là dfs.namenode.rpc-address
- Cách 2: xem trong trình quản lý Ambari nếu nhóm có cài đặt Ambari.

```
Services>HDFS>Configs>Advanced> Advanced hdfs-
site>dfs.namenode.rpc-address
```

Cách chạy:

```
python yourcode.py -r hadoop --hadoop-streaming-jar
/usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
hdfs://sandbox-hdp.hortonworks.com:8020/user/root/data
```

Một cách viết ngắn gọn hơn mà không cần phải xác định host nếu đang chạy trên host mặc định:

```
python yourcode.py -r hadoop --hadoop-streaming-jar
/usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
hdfs:////user/root/data
```

Đọc thêm MRJob trong tài liệu:

https://mrjob.readthedocs.io/en/latest/guides/writing-mrjobs.html#input-and-output-formats

2.2.4. Bài tập áp dụng

Dựa trên hướng dẫn MRJob trên, thực hiện thống kê mỗi từ xuất hiện trong các cuốn sách bên dưới bằng ngôn ngữ Python theo từng trường hợp sau:

- Trường hợp phân biệt hoa thường
- Trường hợp không phân biệt hoa thường
- Tìm từ xuất hiện nhiều nhất trong tài liệu (không phân biệt hoa thường).

Nên: tự tạo dữ liệu là một tập tin văn bản ngắn để test trên local.

Khi thành công, thực hiện tải dữ liệu sau lên HDFS và thực hiện đếm. Dữ liệu gồm 3 cuốn sách:

http://www.gutenberg.org/ebooks/20417

http://www.gutenberg.org/ebooks/5000

http://www.gutenberg.org/ebooks/4300

3. Thang điểm

No.	Criteria	Scores
1	Hoàn thành mức 1 thành công	25%
2	Hoàn thành mức 2 thành công	25%
3	Quay video chứng minh quá trình chạy và thực thi thành công. Nếu không có video thì không xem xét mức 1 và 2 là thành công.	10%

4	Chạy ít nhất trên 3 bộ test khác nhau của mỗi yêu cầu. Các bộ test cần thể hiện khác nhau đặc trưng như kích thước (100 MB, 200MB, 500MB), nội dung, v.v…Đối với level 2 thì test trên các tài liệu đã cung cấp.	10%
5	Báo cáo đầy đủ và chi tiết quá trình thực hiện, mã nguồn. Trình bày rõ ràng, bố cục hợp lý. Tổ chức các tập tin thành từng thư mục thể hiện ý nghĩa của mỗi nhóm như thư mục mã nguồn, báo cáo, test, v.v…	30%
Total		100%

4. Lưu ý

- Bài lab được thực hiện dưới dạng nhóm.
- Hạn chót theo thông tin trên Moodle.
- Ngoài các yêu cầu về nội dung, báo cáo cần có các thông tin cơ bản sau:
  - Thông tin về các thành viên
  - Kế hoạch và phân công
  - Tự đánh giá mức độ hoàn thành của mỗi thành viên cho các công việc phụ trách.
  - Tài liệu tham khảo (nếu có)
- Đạo văn, gian lận trong quá trình làm bài sẽ 0 điểm môn học.

5. References

[1] WordCount, https://docs.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_overview.html