

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN VĂN ANH

**NGHIÊN CỨU VÀ XÂY DỰNG ỨNG DỤNG TƯ VẤN TRÊN NỀN
TẢNG HADOOP MAPREDUCE**

CHUYÊN NGÀNH : HỆ THỐNG THÔNG TIN

MÃ SỐ: 60.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. ĐÀO ĐÌNH KHẢ

HÀ NỘI - 2016

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Đào Đình Khả

Phản biện 1:

Phản biện 2:

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại
Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ... giờ ngày tháng năm

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Chúng ta đang chứng kiến sự bùng nổ thông tin khổng lồ chưa từng thấy từ người sử dụng các dịch vụ Internet và các phương tiện truyền thông. Theo ước tính, thị trường chứng khoán New York sinh ra khoảng 1 Terabyte dữ liệu về các giao dịch mỗi ngày, Facebook đang lưu trữ và truyền bá trên 10 tỷ tấm ảnh (tức khoảng một petabyte), trang web Ancestry.com (một trang web cung cấp dịch vụ lưu giữ gia phả dòng họ) đang lưu trữ khoảng 2,5 petabyte dữ liệu, trang web Internet Archive đang lưu trữ khoảng 2 petabytes dữ liệu và có tốc độ gia tăng 20 terabyte/tháng.

Có nhiều thực tế nảy sinh từ việc bùng nổ thông tin, thông tin thay đổi quá nhanh nên phải có hướng tiếp cận linh hoạt. Bài toán tư vấn đối với các sản phẩm mà họ chưa biết là một trong những trọng tâm nghiên cứu của thương mại điện tử. Số lượng khách hàng và số lượng sản phẩm lớn là thách thức cho các phương pháp tư vấn hiện nay trong huấn luyện để đưa ra kết quả tư vấn. Trong quá trình nghiên cứu triển khai ứng dụng, bên cạnh những vấn đề chung của bài toán tư vấn người dùng như tính thừa thớt dữ liệu huấn luyện, xử lý khách hàng mới, hàng hóa mới thì yêu cầu tăng tốc độ của các giải thuật huấn luyện vẫn là chủ đề mở được cộng đồng quan tâm nghiên cứu. Sự ra đời của Hadoop MapReduce là một mô hình phù hợp để nhắm đến nội dung này.

2. Tổng quan về vấn đề nghiên cứu

Sau nhiều năm giữ bí mật với các đối thủ cạnh tranh, Google đã công bố nền tảng MapReduce ra cộng đồng và trở thành đối tượng nghiên cứu và triển khai cho nhiều ứng dụng khác nhau. Đối với các chuyên gia của Google, MapReduce chỉ được xem là một giải pháp xử lý dữ liệu lớn khi họ đang cố gắng mở rộng bộ máy tìm kiếm của mình. Trên thực tế MapReduce là một mô hình lập trình, hay một thuật giải của khoa học máy tính. Thuật giải MapReduce đưa ra nguyên lý chia việc xử lý thành nhiều khối công việc nhỏ, phân tán khắp các nút tính toán (tiêu biểu là các server thông thường), rồi kết hợp kết quả chung cho lớp các bài toán cụ thể. Hiện tại MapReduce cho phép thực hiện trên các phần cứng thông thường (commodity hardware) mà không đòi hỏi các server chạy MapReduce có khả năng tính toán, lưu trữ và truy xuất mạnh mẽ. Do vậy, chi phí triển khai ứng dụng trên MapReduce sẽ thấp hơn nhiều so

với các cách tiếp cận khác. Một lợi thế khác của MapReduce là nó đơn giản hoá các giải thuật tính toán phân tán. Trên MapReduce ta chỉ cần xây dựng hai hàm Map và Reduce phù hợp với thành phần xử lý dữ liệu đầu vào cho từng ứng dụng. Do vậy, các nhà phát triển ứng dụng phân tán có thể trọng tâm nhiều hơn vào phần logic của ứng dụng và bỏ qua các chi tiết phức tạp của xử lý phân tán.

Chính vì lý do trên, học viên lựa chọn đề tài “*Nghiên cứu và xây dựng ứng dụng tư vấn trên nền tảng Hadoop MapReduce*” được thực hiện trong khuôn khổ luận văn thạc sĩ chuyên ngành Hệ thống thông tin. Mục tiêu, đối tượng và phương pháp nghiên cứu cụ thể của đề tài được trình bày chi tiết trong các mục tiếp theo.

3. Mục đích nghiên cứu

Nghiên cứu phương pháp xây dựng hệ tư vấn người dùng đối với sản phẩm trên Hadoop MapReduce. Đánh giá hiệu quả của phương pháp xây dựng hệ thống tư vấn trên Hadoop MapReduce so với các phương pháp truyền thống.

4. Đối tượng nghiên cứu

Đối tượng nghiên cứu là mô hình xử lý dữ liệu lớn trên Hadoop MapReduce cho bài toán tư vấn người dùng.

5. Phạm vi nghiên cứu

Phạm vi nghiên cứu là hệ thống tư vấn theo mô hình truyền thống và hệ thống tư vấn cài đặt trên Hadoop MapReduce.

6. Phương pháp nghiên cứu

6.1. Nghiên cứu lý thuyết:

- Dựa trên nguồn tài liệu kỹ thuật đã được công bố, nghiên cứu kỹ thuật và mô hình triển khai ứng dụng trên Hadoop MapReduce.
- Nghiên cứu các phương pháp xây dựng hệ tư vấn dựa vào người dùng và dựa vào sản phẩm trong các hệ thống thương mại điện tử. Thực hiện cài đặt và kiểm nghiệm các mô hình cơ bản.
- So sánh và đánh giá kết quả tư vấn về kết quả và thời gian thực hiện các phương pháp tư vấn cơ bản với phương pháp tư vấn cài đặt trên Hadoop MapReduce.

6.2. Nghiên cứu thực nghiệm:

Phương pháp thực nghiệm được tiến hành trên các bộ dữ liệu về phim được cộng đồng nghiên cứu sử dụng. Dự kiến luận văn sử dụng một trong các bộ dữ liệu sau để tiến hành thử nghiệm kết quả dự đoán các mô hình:

Bộ dữ liệu MovieLens1: Được thu thập bởi Dự án nghiên cứu GroupLens của Đại học Minnesota. Tập dữ liệu MovieLens có ba lựa chọn với kích thước khác nhau lần lượt là: MovieLens 100k, MovieLens 1M và MovieLens 10M. Luận văn sử dụng tập dữ liệu MovieLens 1M.

Tập MovieLens 1M chứa 1,000,209 đánh giá của 6040 người dùng cho khoảng 3900 bộ phim. Tất cả đánh giá được lưu trong file “ratings.dat” theo định dạng: UserID::MovieID::Rating::Timestamp. Trong đó:

- UserID là số nguyên trong khoảng 1 đến 6040.
- MovieID là số nguyên trong khoảng 1 đến 3952.
- Rating là số nguyên trong khoảng 1 đến 5.

Timestamp: là thời gian đánh giá.

Cấu trúc luận văn

Nội dung của luận văn được trình bày trong ba phần chính như sau:

1. Phần mở đầu
2. Phần nội dung: bao gồm ba chương

Chương 1: Tổng quan về Hadoop MapReduce

Chương 2: Phát triển hệ thống tư vấn và cài đặt trên Hadoop MapReduce

Chương 3: Thử nghiệm và đánh giá.

CHƯƠNG I. TỔNG QUAN VỀ BIG DATA VÀ HADOOP-MAPREDUCE

1.1 Giới thiệu về Big Data

1.1.1 Định nghĩa Big Data

Big Data là thuật ngữ dùng để mô tả lượng dữ liệu khổng lồ, có dung lượng lớn (lên đến hàng Terabytes hay Petabytes), tăng trưởng nhanh và có nội dung đa dạng.

Big Data thường liên quan đến một số loại dữ liệu [1]:

- Các loại dữ liệu truyền thống của doanh nghiệp: thông tin người dùng, dữ liệu giao dịch, các dữ liệu kế toán nói chung...
- Dữ liệu do máy tự sinh, dữ liệu cảm biến: dữ liệu cảm biến của các thiết bị, các file log sinh ra khi chạy các thiết bị phần cứng hoặc chạy các ứng dụng,...
- Dữ liệu xã hội: dữ liệu người dùng, phản hồi, hoạt động của người dùng, tin nhắn, bài đăng, bình luận trên các mạng xã hội như Facebook, Twitter,...

1.1.2 Tầm quan trọng của Big Data

Big Data là công nghệ thu thập thông tin quy mô lớn từ các website. Các doanh nghiệp thường vận dụng công cụ này nhằm phục vụ công việc dự đoán xu hướng thị trường, nâng cao chất lượng sản phẩm hoặc dịch vụ hiện có, tạo ra sản phẩm mới hoặc tìm hiểu về hành vi khách hàng. Phân tích dữ liệu cũng có thể giúp các doanh nghiệp thích nghi, tạo ra nội dung website thu hút nhiều khách hàng hơn, có được cái nhìn sâu sắc vào hành vi mua hàng. Dữ liệu càng nhiều thì càng tốt cho công ty. Để làm được như vậy, doanh nghiệp nên cung cấp nội dung trên nhiều nền tảng social media, nhằm thu thập được nhiều thông tin từ những điểm tiếp xúc với khách hàng bằng cách tìm hiểu qua hệ thống cơ sở dữ liệu, công ty có thể tạo ra nội dung có liên quan hơn với người đọc.

Khi Big Data được lưu trữ, xử lý và phân tích một cách chuẩn xác, chúng ta có thể có được nhiều thông tin hữu ích để hiểu hơn về công việc của mình qua đó giúp cho công việc đạt hiệu quả cao hơn. Trong hệ thống thương mại điện tử, trên mạng xã hội giúp cho nhà quản lý hiểu rõ hơn về khách hàng và xu hướng của họ, đồng thời giúp các hệ thống tư vấn đưa ra những gợi ý chính xác cho người dùng.

1.2 Tổng quan về Hadoop

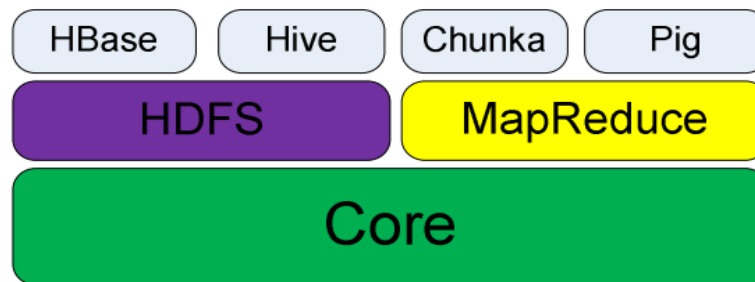
1.2.1 Giới thiệu Hadoop và các thành phần của Hadoop Ecosystem

1.2.1.1 Giới thiệu Hadoop

Ta có thể khái quát như sau:

- Hadoop là một framework cho phép phát triển các ứng dụng phân tán.
- Hadoop viết bằng Java. Tuy nhiên, nhờ cơ chế streaming, Hadoop cho phép phát triển các ứng dụng phân tán bằng cả java lẫn một số ngôn ngữ lập trình khác như C++, Python, Pearl.
- Hadoop cung cấp một phương tiện lưu trữ dữ liệu phân tán trên nhiều node, hỗ trợ tối ưu hoá lưu lượng mạng, đó là HDFS. HDFS che giấu tất cả các thành phần phân tán, các nhà phát triển ứng dụng phân tán sẽ chỉ nhìn thấy HDFS như một hệ thống file cục bộ bình thường.
- Hadoop giúp các nhà phát triển ứng dụng phân tán tập trung tối đa vào phần logic của ứng dụng, bỏ qua được một số phần chi tiết kỹ thuật phân tán bên dưới (phần này do Hadoop tự động quản lý).
- Hadoop là Linux-based. Tức Hadoop chỉ chạy trên môi trường Linux 2.

1.2.1.2 Các thành phần của Hadoop Ecosystem



Hình 1-1: Cấu trúc các thành phần của Hadoop

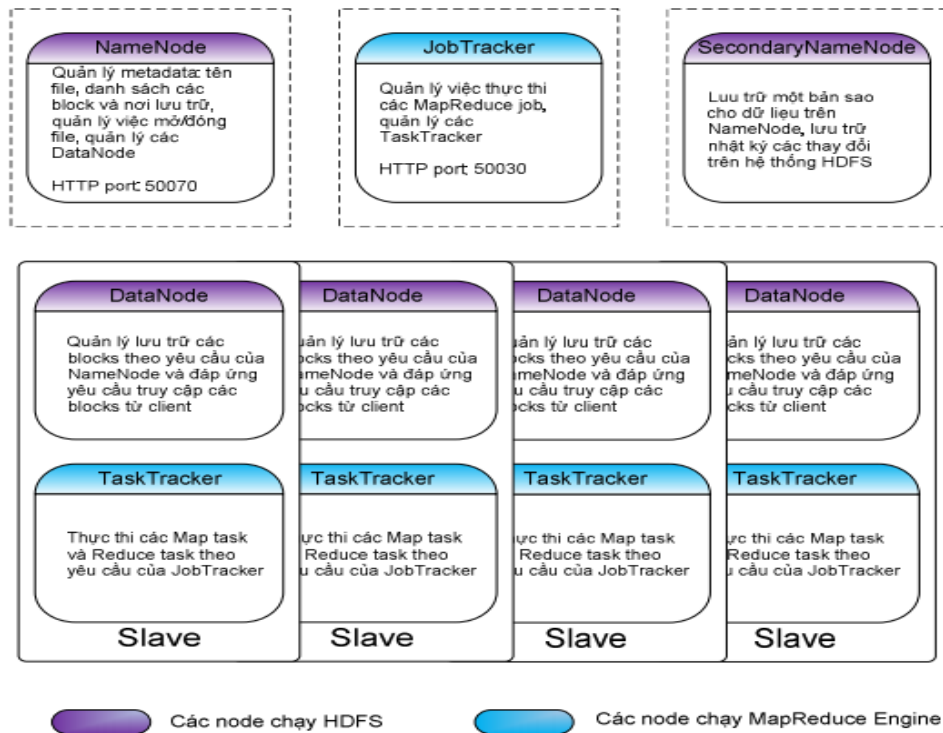
Trong khuôn khổ của luận văn này, em chỉ nghiên cứu hai phần quan trọng nhất của Hadoop, đó là HDFS và MapReduce.

1.2.2 Lưu trữ dữ liệu lớn trên HDFS

1.2.2.1 Tổng quan của một Hadoop Cluster

Một Hadoop cluster có HDFS và MapReduce là hai thành phần chính. Kiến trúc của Hadoop là kiến trúc master-slave, và cả hai thành phần HDFS và MapReduce đều tuân theo kiến trúc master-slave này.

Kiến trúc một Hadoop cluster như sau:



Hình 1-2: Tổng quan một Hadoop cluster

HDFS – Hadoop distributed file system (Hệ thống file phân tán Hadoop) là một dự án con của Apache. HDFS được thiết kế để lưu trữ dữ liệu lớn và file phân tán một cách đáng tin cậy. Kiến trúc của HDFS là kiến trúc dựa trên mô hình chủ - tớ (master – slave), nó được thiết kế để giảm thiểu việc tiêu tốn tài nguyên phần cứng [2,6,7].

1.2.2.2 Khái niệm Block trong HDFS

Tương tự như block trong hệ thống file thông thường, block trong HDFS cũng là lượng dữ liệu tối thiểu của mỗi lần đọc, ghi dữ liệu.

1.2.2.3 Kiến trúc của HDFS

Giống như các hệ thống file khác, HDFS duy trì một cấu trúc cây phân cấp các file, thư mục mà các file sẽ đóng vai trò là các node lá. Trong HDFS, mỗi file sẽ được chia ra làm một hay nhiều block và mỗi block này sẽ có một block ID để nhận diện. Các block của cùng một file (trừ block cuối cùng) sẽ có cùng kích thước và kích thước này được gọi là block size của file đó. Mỗi block của file sẽ được lưu trữ thành ra nhiều bản sao (replica) khác nhau vì mục đích an toàn dữ liệu.

1.2.2.4 NameNode và quá trình tương tác giữa client và HDFS

Việc tồn tại duy nhất một NameNode trên một hệ thống HDFS đã làm đơn giản hoá thiết kế của hệ thống và cho phép NameNode ra những quyết định thông minh trong việc sắp xếp các block dữ liệu lên trên các DataNode dựa vào các kiến thức về môi trường hệ thống như: cấu trúc mạng, băng thông mạng, khả năng của các DataNode... Tuy nhiên, cần phải tối thiểu hoá sự tham gia của NameNode vào các quá trình đọc/ghi dữ liệu lên hệ thống để tránh tình trạng nút thắt cổ chai (bottle neck).

Client sẽ không bao giờ đọc hay ghi dữ liệu lên hệ thống thông qua NameNode. Thay vào đó, client sẽ hỏi NameNode xem nên liên lạc với DataNode nào để truy xuất dữ liệu. Sau đó, client sẽ cache thông tin này lại và kết nối trực tiếp với các DataNode để thực hiện các thao tác truy xuất dữ liệu. Chúng ta sẽ bỏ xé quá trình đọc một file từ HDFS và ghi một file lên HDFS thông qua việc tương tác giữa các đối tượng từ phía client lên HDFS.

1.2.2.5 Quá trình đọc file

Trong quá trình một client đọc một file trên HDFS, ta thấy client sẽ trực tiếp kết nối với các DataNode để lấy dữ liệu chứ không cần thực hiện gián tiếp qua NameNode (master của hệ thống). Điều này sẽ làm giảm đi rất nhiều việc trao đổi dữ liệu giữa client NameNode, khối lượng luân chuyển dữ liệu sẽ được trải đều ra khắp cluster, tình trạng bottle neck sẽ không xảy ra. Do đó, cluster chạy HDFS có thể đáp ứng đồng thời nhiều client cùng thao tác tại một thời điểm.

1.2.2.6 Quá trình ghi file

Cũng giống như trong quá trình đọc, client sẽ trực tiếp ghi dữ liệu lên các DataNode mà không cần phải thông qua NameNode. Một đặc điểm nổi trội nữa là khi client ghi một block với chỉ số replication là n , tức nó cần ghi block lên n DataNode, nhờ cơ chế luân chuyển block dữ liệu qua ống dẫn (pipe) nên lưu lượng dữ liệu cần write từ client sẽ giảm đi n lần, phân đều ra các DataNode trên cluster.

1.2.2.7 Tổ chức dữ liệu

Hệ thống file phân tán Hadoop là hệ thống file dựa trên cây phân cấp truyền thống giống như UNIX. Người dùng có thể thêm, đổi tên, xóa file hoặc thư mục trong hệ thống. Thư mục gốc của Hadoop được kí hiệu bởi “/”, các thư mục con và file có thể được tạo ra bên trong thư mục gốc[3].

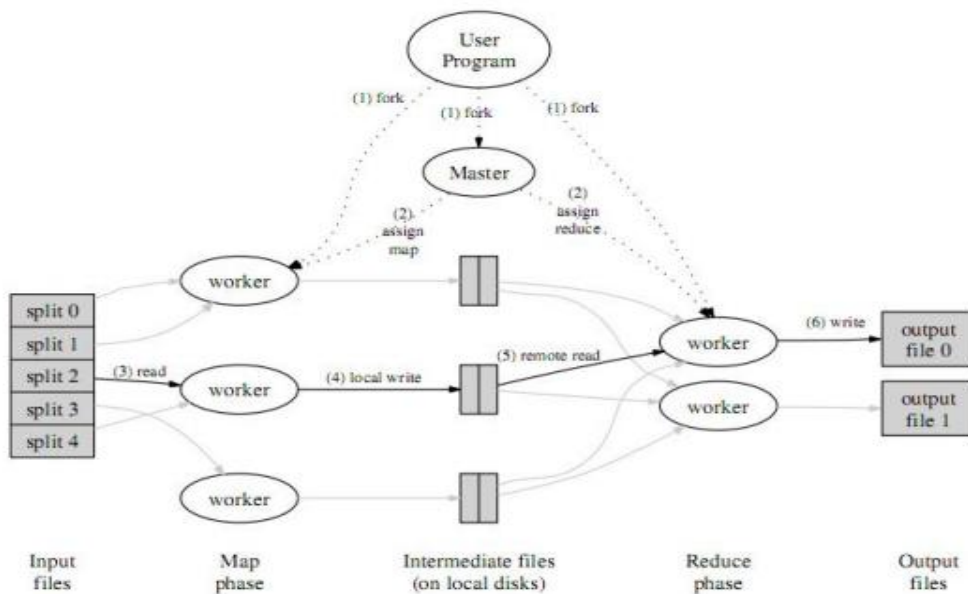
Dữ liệu được tổ chức dưới dạng nhiều khối dữ liệu. Kích thước mặc định của một khối dữ liệu là 64MB. Khi một file dữ liệu được tạo ra trên HDFS, nó sẽ được lưu tạm ở bộ nhớ địa phương cho đến khi nó đạt đủ kích thước của một khối dữ liệu, khi đó máy khách sẽ gửi yêu cầu đến NameNode, NameNode kiểm tra các DataNode đang sẵn sàng và gửi lại thông tin về địa chỉ khối đích, mã của node tới máy khách. Máy khách nhận được thông tin này sẽ đẩy dữ liệu từ bộ nhớ địa phương đến DataNode tương ứng.

HDFS có thể được truy cập sử dụng các cách sau[3] :

- Java APIs.
- Hadoop command line APIs.
- C/C++ language wrapper APIs.
- WebDAV.
- DFSAdmin.
- RESTFul APIs for HDFS.

1.3 Giới thiệu MapReduce

1.3.1 Giới thiệu mô hình tính toán MapReduce

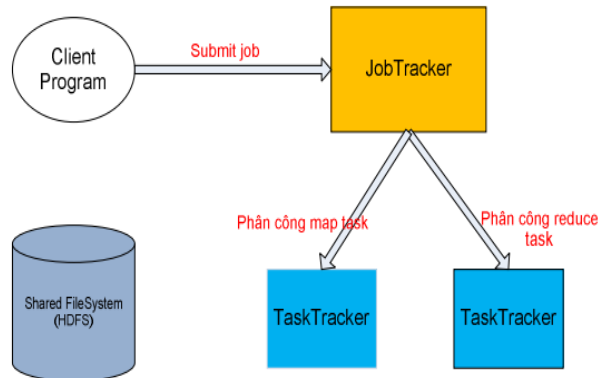


Hình 1-3: Mô hình MapReduce của Google

MapReduce là một framework cho phép lập trình viên viết các chương trình xử lý lượng lớn dữ liệu một cách song song thông qua môi trường phân tán. MapReduce là mô hình lập trình dựa trên key – value, chương trình sẽ đọc vào tập key – value và trả về tập key – value mới sau khi xử lý[3].

1.3.2 Kiến trúc của MapReduce

1.3.2.1 Kiến trúc các thành phần (JobTracker, TaskTracker)



Hình 1-4: Kiến trúc các thành phần

Xét một cách trừu tượng, Hadoop MapReduce gồm 4 thành phần chính riêng biệt:

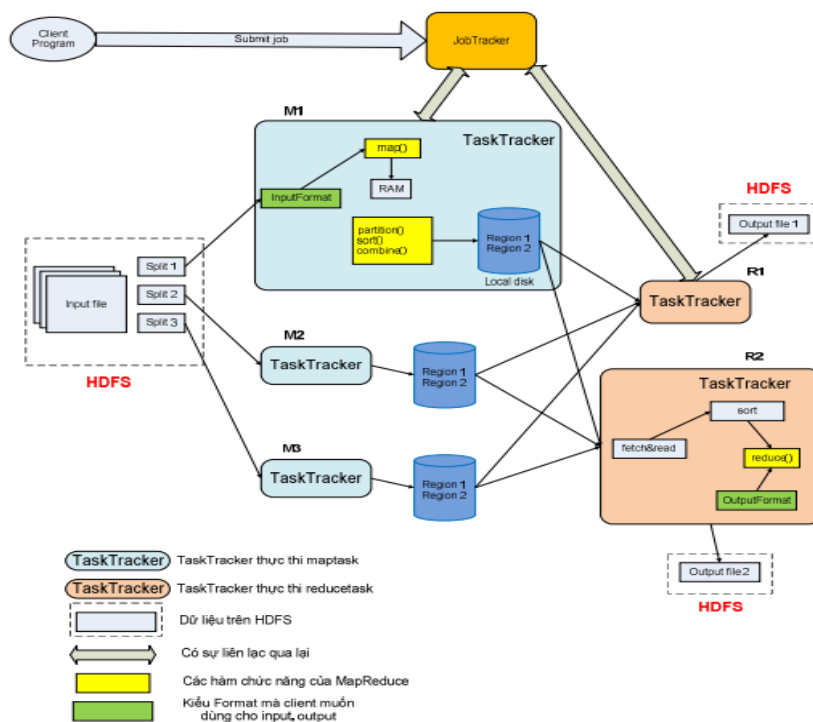
Client Program.

JobTracker

TaskTracker

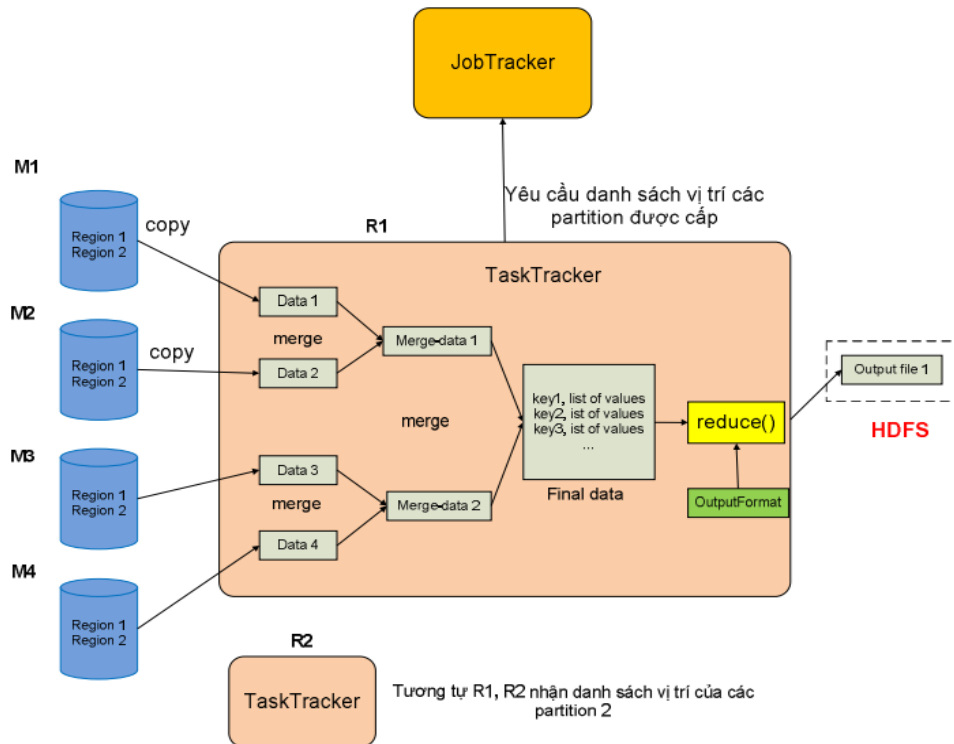
1.3.2.2 Cơ chế hoạt động

Hình I-8 mô tả cơ chế hoạt động tổng quát của Hadoop MapReduce, mô tả quá trình từ lúc ClientProgram yêu cầu thực hiện job đến lúc các TaskTracker thực hiện reduce task trả về các kết quả output cuối cùng



Hình 1-5: Cơ chế hoạt động của Hadoop MapReduce

Trong cấu trúc dữ liệu của mình, JobTracker có một job scheduler với nhiệm vụ lấy vị trí các split (từ HDFS do chương trình client tạo), sau đó nó sẽ tạo một danh sách các task để thực thi. Với từng split thì nó sẽ tạo một maptask để thực thi, mặc nhiên số lượng maptask bằng với số lượng split. Còn đối với reduce task, số lượng reduce task được xác định bởi chương trình client. Bên cạnh đó, JobTracker còn lưu trữ thông tin trạng thái và tiến độ của tất cả các task.



Hình 1-6: Cơ chế hoạt động của Reduce Task

Khác với TaskTracker thực hiện maptask, TaskTracker thực hiện reduce task theo một cách khác. TaskTracker thực hiện reduce task với dữ liệu input là danh sách các vị trí của một region cụ thể trên các output được ghi trên localdisk của các maptask. Điều này có nghĩa là với region cụ thể, JobTracker sẽ thu thập các region này trên các output của các maptask thành một danh sách các vị trí của các region

Khi TaskTracker thực hiện thành công reduce task, thì nó sẽ gửi thông báo trạng thái “completed” của reduce task được phân công đến JobTracker. Nếu reduce task này là task cuối cùng của job thì JobTracker sẽ trả về cho chương trình người dùng biết job này đã hoàn thành. Ngay lúc đó JobTracker sẽ làm sạch cấu trúc dữ liệu của mình mà dùng cho job này, và thông báo cho các TaskTracker xóa tất cả các dữ liệu output của các map task (Do dữ liệu maptask chỉ là dữ liệu trung gian làm input cho reduce task, nên không cần thiết để lưu lại trong hệ thống).

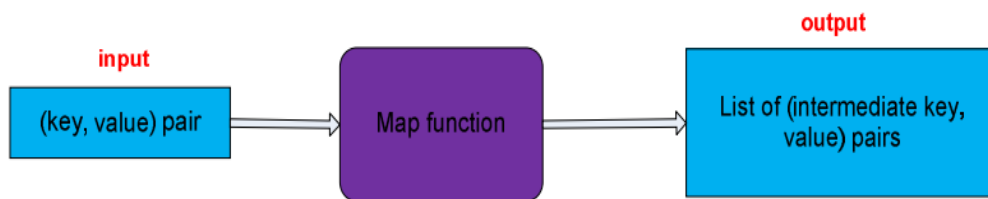
1.3.3 Mô hình làm việc và luồng dữ liệu của MapReduce

Phần này em sẽ trình bày về cách mà một chương trình MapReduce xử lý dữ liệu và luồng di chuyển của dữ liệu trong quá trình thực thi một chương trình MapReduce[7, 6].

MapReduce Job là một đơn vị thực thi hoàn chỉnh, nó thực hiện một công việc nhất định mà client muốn thể hiện. Một MapReduce Job bao gồm 3 thành phần là: dữ liệu đầu vào, chương trình MapReduce, thông tin cấu hình.

Hadoop chia mỗi MapReduce Job thành các tasks, có 2 loại tasks là: map task và reduce task.

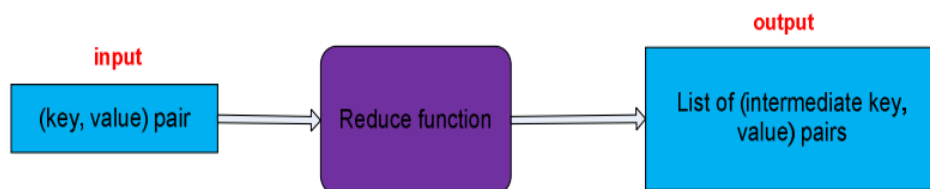
1.3.3.1 Hàm Map



Hình 1-7: Hàm Map

Người dùng đưa một cặp dữ liệu (key,value) làm input cho hàm map, và tùy vào mục đích của người dùng mà hàm map sẽ trả ra danh sách các cặp dữ liệu (intermediate key,value).

1.3.3.2 Hàm Reduce



Hình 1-8: Hàm Reduce

Hệ thống sẽ gom nhóm tất cả value theo intermediate key từ các output của hàm map, để tạo thành tập các cặp dữ liệu với cấu trúc là (key, tập các value cùng key). Dữ liệu input của hàm reduce là từng cặp dữ liệu được gom nhóm ở trên và sau khi thực hiện xử lý nó sẽ trả ra cặp dữ liệu (key, value) output cuối cùng cho người dùng.

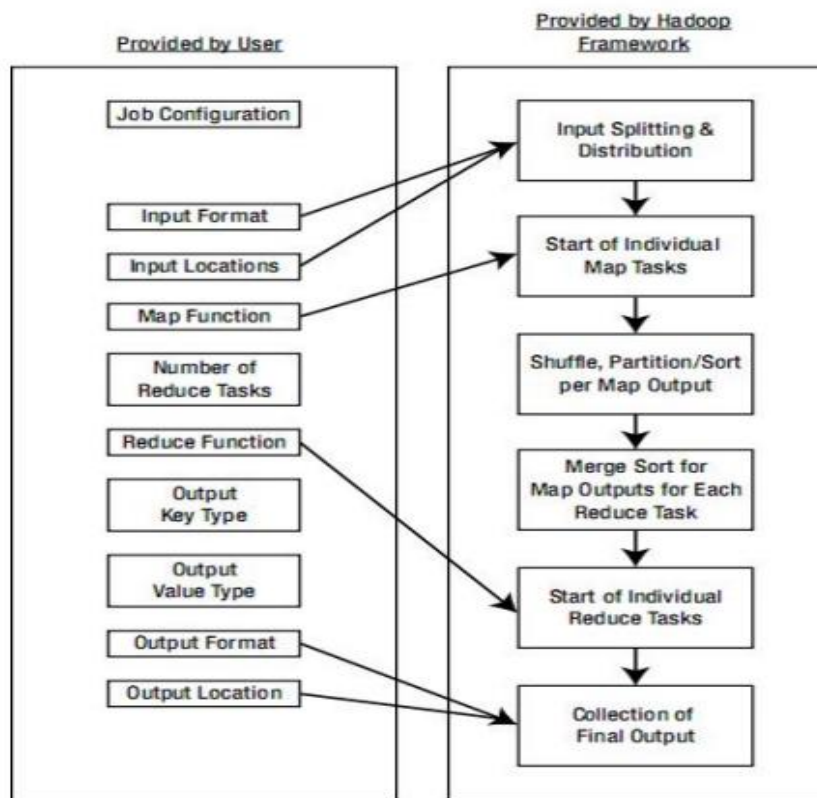
1.3.4 MapReduce và HDFS (Các đặc điểm tối ưu của MapReduce khi kết hợp với HDFS)

MapReduce đơn thuần làm nhiệm vụ xử lý tính toán song song, vậy trong một hệ thống phân tán thì dữ liệu sẽ được kiểm soát như thế nào để người dùng có thể dễ dàng truy

xuất, do đó việc sử dụng HDFS cho việc bổ sung các input split của MapReduce xuống và có kích thước gần bằng với kích thước block, điều này làm tăng hiệu suất cho việc xử lý song song và đồng bộ của các TaskTracker với từng split mà có thể xử lý riêng biệt này. Thêm vào đó, các dữ liệu output cuối cùng của một MapReduce Job cũng được lưu trữ xuống HDFS, điều này giúp cho người dùng tại một máy tính nào đó trong hệ thống đều có thể lấy được toàn bộ kết quả output này thông qua các phương thức thuộc cơ chế quản lý của HDFS (Tính trong suốt). Bên cạnh đó, khi các block không đạt tình trạng cân bằng (load-balancer) thì HDFS có cơ chế thực hiện việc cân bằng các block trở lại một cách hiệu quả, điều này sẽ làm gia tăng hiệu suất của data locality.

1.3.5 Phát triển ứng dụng theo mô hình MapReduce với Hadoop MapReduce

Sau đây là toàn bộ quá trình phát triển một ứng dụng theo mô hình MapReduce với HadoopMapReduce.



Hình 1-9: Phát triển ứng dụng MapReduce trên Hadoop

Quá trình phát triển được phân rõ ra theo công việc nào do người dùng thực hiện can thiệp và công việc nào bên trong framework tự làm.

1.4 Kết luận chương 1

Chương 1 đã giới thiệu và đưa ra tầm quan trọng của Big Data trong việc phân tích dữ liệu lớn. Khi Big Data được lưu trữ, xử lý và phân tích một cách chuẩn xác, chúng ta có thể có được nhiều thông tin hữu ích để hiểu hơn về công việc của mình qua đó giúp cho công việc đạt hiệu quả cao hơn. Có rất nhiều công nghệ được triển khai phục vụ cho việc khai thác dữ liệu lớn trong đó tiêu biểu là việc áp dụng framework Hadoop – MapReduce một phần mềm mã nguồn mở cho phép tính toán phân tán hứa hẹn sẽ đem lại hiệu quả cao về năng suất, tốc độ xử lý với yêu cầu đáp ứng thời gian thực.

Trong chương tiếp theo em sẽ trình bày một số phương pháp cụ thể ở đây là lọc cộng tác dựa trên mô hình để áp dụng cho mô hình MapReduce trong bài toán về hệ tư vấn.

CHƯƠNG II. ỨNG DỤNG HADOOP-MAPREDUCE CHO HỆ TƯ VẤN

2.1 Giới thiệu vấn đề

2.1.1 Phát biểu bài toán tư vấn

Tùy vào phương pháp lọc tin, các hệ tư vấn được phân làm 3 loại : Tư vấn dựa vào phương pháp lọc theo nội dung(Content-Based Filtering Recommendation), Tư vấn dựa vào lọc cộng tác (Collaborative Filtering Recommendation), Tư vấn dựa vào phương pháp lọc kết hợp(Hybrid Filtering Recommendation)

Mỗi phương pháp lọc áp dụng cho các hệ tư vấn được phân thành 2 hướng tiếp cận: lọc dựa vào bộ nhớ (memory-based filtering) và lọc dựa vào mô hình(Model-Based Filtering).

Trong đồ án này em tập trung nghiên cứu vào phương pháp tư vấn cộng tác.

Để xây dựng hệ tư vấn bằng lọc cộng tác, cần phải xác định được lịch sử quan điểm của các người dùng với các sản phẩm khác nhau trong hệ thống. Quan điểm ở đây được chia làm 2 loại là quan điểm tường minh (explicit) và quan điểm không tường minh (implicit)[5].

Bảng 2-1: Ma trận đánh giá ví dụ 2.1

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	5	2	4	3	5	0
u_2	5	4	1	0	2	3
u_3	0	3	2	2	4	5
u_4	5	3	?	3	?	?

2.1.2 Các phương pháp xây dựng hệ tư vấn

Lọc cộng tác là kỹ thuật tư vấn được sử dụng rộng rãi trong các hệ tư vấn. Lọc cộng tác là phương pháp tự động đưa ra các dự đoán (lọc –filtering) về sở thích của một người dùng bằng cách thu thập và phân tích thông tin về sở thích của một lượng lớn người dùng khác. Phương pháp lọc cộng tác dựa trên giả thiết rằng nếu người dùng A có sở thích tương đồng với người dùng B ở một vấn đề nào đó thì có nhiều khả năng người dùng A có cùng quan điểm với người dùng B về vấn đề khác.

Đồ án sẽ trình bày phương pháp lọc cộng tác dựa vào mô hình gồm 2 kỹ thuật: Lọc cộng tác dựa vào mô hình mạng Bayes(Bayesian networks), Lọc cộng tác dựa vào mô hình phân cụm SVD.

2.2 Phương pháp mạng Bayes cho lọc cộng tác

2.2.1 Định lý mạng Bayes

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được kí hiệu là $P(A|B)$, và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi là xác có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó

Theo định lý Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

Xác suất xảy ra A của riêng nó, không quan tâm tới B. Kí hiệu là $P(A)$ và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là “tiên nghiệm” theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B.

Xác suất xảy ra B của riêng nó, không quan tâm tới A. Kí hiệu là $P(B)$ và đọc là “xác suất của B”. Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết.

Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là $P(B|A)$ và đọc là “xác suất của B nếu có A”. Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra.

Khi biết 3 đại lượng này, xác suất của A khi biết B cho bởi công thức:

$$P_{(A|B)} = \frac{P(B|A)P(A)}{P(B)} = \frac{\text{likelihood} * \text{prior}}{\text{normalizing_constant}}$$

Từ đó dẫn tới

$$P(A|B)P(B) = P(A \cap B) = P(B|A) P(A)$$

2.2.2 Phân loại dựa vào mạng Bayes

Khi không gian mẫu được đặc trưng bởi n thuộc tính, với mỗi giá trị đích v, việc ước lượng xác suất đồng thời $P((a_1, a_2, \dots, a_n)/v)$ cho mỗi (a_1, a_2, \dots, a_n) sẽ rất khó khăn. Một phương pháp phân lớp mạng Bayes thường được áp dụng.

Phân lớp Naive Bayes áp dụng cho các bài toán mà mẫu x được mô tả bởi liên kết các giá trị-thuộc tính và hàm đích $f(x)$ có thể lấy giá trị trong tập hữu hạn V. Với tập mẫu huấn luyện có giá trị hàm mục tiêu cho trước và có một mẫu mới được biểu diễn bởi các giá trị thuộc tính (a_1, a_2, \dots, a_n) . Ta phải dự đoán giá trị mục tiêu hay phân lớp mẫu thử này.

Tóm lại, phương pháp học mạng Bayes Naive bao gồm các bước tính toán $P(v_j)$, $P(a_i|v_j)$ dựa trên tần suất dữ liệu, sau đó các kết quả này được dùng để phân lớp mẫu mới theo biểu thức (2.2c) với giả thiết các giá trị thuộc tính là độc lập có điều kiện.

2.2.3 Áp dụng mạng Bayes cho hệ tư vấn

Ý tưởng : tính xác suất một mẫu X với tập thuộc tính đã biết phụ thuộc vào một phân lớp C_i :

$$P_{(C_i/X)} = \frac{P(X/C_i)P(C_i)}{P(X)}$$

Công thức xác định xác suất Naive Bayes:

$$\text{Prediction} = \operatorname{argmax}_{j \in \text{classSet}} P(\text{class}_j | \prod_o X_o = x_o | \text{class}_j)$$

Theo Laplace Estimator:

$$P(X_i=x_i|Y=y) = \frac{(X_i=x_i, Y=y)+1}{(Y=y)+X_i}$$

Trong đó:

- $|X_i|$: kích thước của tập $\{x_i\}$
- $X=(x_1, x_2, \dots, x_n)$ là mẫu cần xét , x_i là thuộc tính thứ i của c
- Prediction : kết quả dự đoán của cặp user-item
- class_j : phân lớp j ($j=1, 2, 3, 4, 5$)

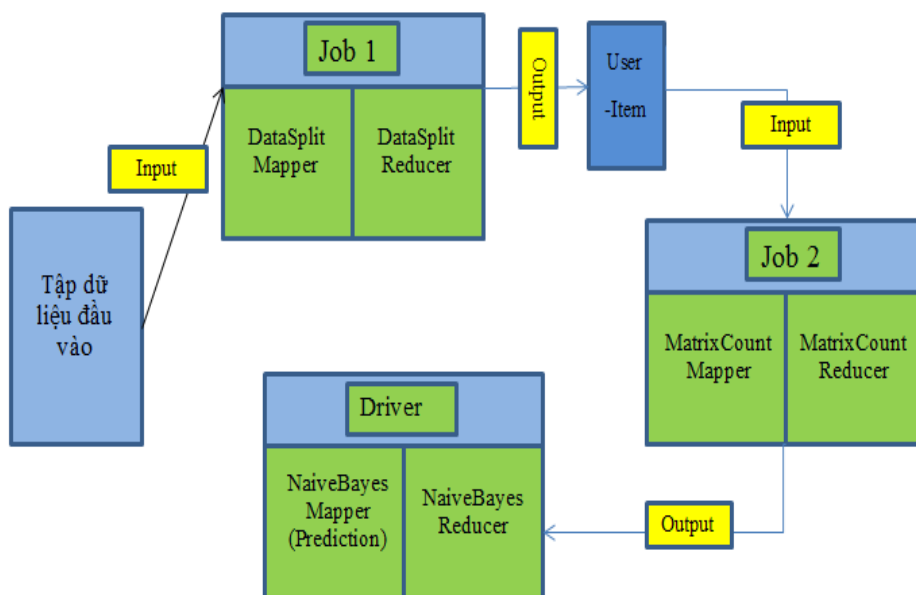
Kết quả dự đoán là giá trị class_j có giá trị xác suất lớn nhất để phân lớp cho đối tượng cần phân loại.

Thuật toán chia ra làm 2 giai đoạn:

Giai đoạn 1: Huấn luyện

Giai đoạn 2: Phân lớp

2.2.4 Hadoop-Map Reduce dựa vào mạng Bayes



Hình 2-1: Áp dụng MapReduce cho kỹ thuật lọc cộng tác sử dụng định lí Bayes

Bước 1: Xử lý tập dữ liệu đầu vào (Job 1)

Bước 2: Đếm số lượng đánh giá (Job 2)

Bước 3: Tính toán và đưa ra dự đoán.

2.3 Phương pháp SVD cho lọc cộng tác

2.3.1 Giới thiệu phương pháp SVD (Singular Value Decomposition)

Bài toán phân rã giá trị riêng SVD được phát biểu như sau:

Với bất kỳ ma trận A kích thước $M \times N$ nào có số $M \geq N$, có thể được viết dưới dạng tích của một ma trận U trực giao theo cột có kích thước $M \times N$, một ma trận chéo W có kích thước $N \times N$ với các số trên đường chéo là không âm, và ma trận chuyển vị của một ma trận trực giao V có kích thước $N \times N$:

$$[A] = [U] \times [S] \times [V^T] = [U] \times \begin{bmatrix} w_1 & & \\ & \dots & \\ & & w_N \end{bmatrix} \times [V^T]$$

Đường chéo khởi tạo r của $S(s_1, s_2, \dots, s_r)$ có các đặc trưng $s_i > 0$ và $s_1 \geq s_2 \geq \dots \geq s_r$. Trong đó, r cột đầu tiên của U là vector riêng của AA^T và đại diện cho các vector riêng trái của A trong không gian mở rộng cột. r cột đầu tiên của V là vector riêng của $A^T A$ và đại diện cho các vector riêng phải của A trong không gian mở rộng hàng. Nếu chúng ta chỉ tập trung vào các r giá trị riêng khác 0, kích thước hiệu quả SVD của ma trận U , S và V sẽ trở thành $M \times r$, $r \times r$ và $r \times N$ tương ứng.

2.3.2 Áp dụng phương pháp SVD cho hệ tư vấn

Các bước mà thuật toán SVD sẽ tiến hành như sau:

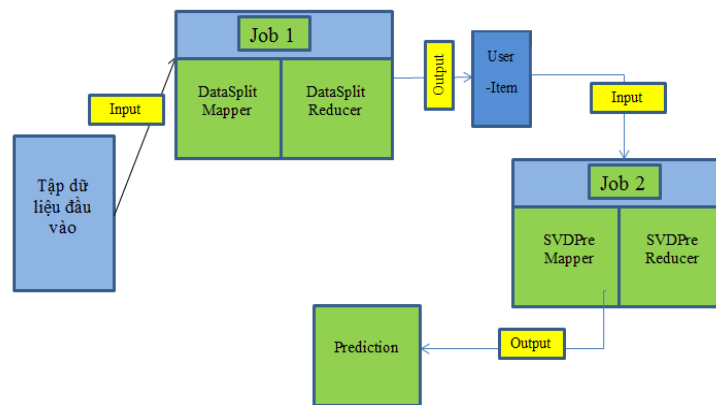
Bước 1: Xác định các ma trận đánh giá R ban đầu, có kích thước $M \times N$, trong đó bao gồm các xếp hạng của M người dùng trên N sản phẩm trong đó r_{ij} là đánh giá của người dùng u_i trên sản phẩm p_j .

Bước 2: Xử lý ma trận đánh giá R để loại bỏ tất cả các giá trị dữ liệu chưa đánh giá.

Bước 3: Tính toán SVD của R'' và có được ma trận U , S và V , có kích thước $M \times M$, $M \times N$, và $N \times N$ tương ứng.

Bước 4: Thực hiện các bước giảm chiều bằng cách giữ chỉ k đường chéo từ ma trận S để có được một ma trận $k \times k$ ký hiệu S_k ..

2.3.3 Phương pháp SVD dựa vào Hadoop Map Reduce



Hình 2-2: Áp dụng MapReduce cho kỹ thuật lọc cộng tác bằng phương pháp SVD

Bước 1: Xử lý tập dữ liệu đầu vào (Job 1)

Bước 2: Tính toán và đưa ra dự đoán đánh giá của người dùng u với sản phẩm i (Job2)

Lớp SVDPreMapper thực hiện công việc:

Bước 1: Đọc tập dữ liệu user-item đưa ra ma trận đánh giá R ban đầu có kích thước $M \times N$, trong đó M là số lượng item được đánh giá, N số lượng người dùng, r_{ij} tương ứng là đánh giá của người dùng j cho sản phẩm i .

Bước 2: Chuẩn hóa ma trận đầu vào:

Bước 3: Tính toán SVD

Bước 4: Đưa ra kết quả dự đoán.

2.3.4 Vấn đề chọn chiều trong phương pháp SVD.

Trong phương pháp SVD độ chính xác hay tốc độ tính toán của các đánh giá phụ thuộc lớn vào việc chọn K là hạng hay số chiều của ma trận S .

K càng lớn tốc độ tính toán càng lâu hơn do phải tính ma trận có chiều lớn hơn. Tuy nhiên độ chính xác có thể cao hơn.

2.4 Kết luận chương 2

Như vậy, chương 2 đã trình bày về định nghĩa và cách áp dụng framework MapReduce thành công cho 2 phương pháp mạng Bayes và SVD trong lọc cộng tác dựa trên mô hình cho bài toán về hệ tư vấn. Việc áp dụng Hadoop và MapReduce cho các phương pháp mang đến nhiều lợi ích như: tăng hiệu năng của việc tính toán khi tận dụng được tài nguyên phần cứng để thực hiện tính toán song song, tăng khả năng chịu lỗi khi việc tính toán không phụ thuộc vào việc xảy ra lỗi của một máy.

Cụ thể trong chương tiếp theo em sẽ đưa ra thực nghiệm và so sánh kết quả, độ đo kiểm nghiệm, độ chính xác cũng như hiệu quả thời gian cho từng phương pháp.

CHƯƠNG III. THỰC NGHIỆM VÀ KẾT QUẢ

3.1 Dữ liệu thực nghiệm

Đồ án sử dụng tập dữ liệu MovieLens[13]. Đây là tập dữ liệu được thu thập bởi Dự án nghiên cứu GroupLens của Đại học Minnesota. Tập dữ liệu MovieLens có ba lựa chọn với kích thước khác nhau lần lượt là: MovieLens 100k, MovieLens 1M và MovieLens 10M. Đồ án sử dụng 2 tập dữ liệu

- Tập MovieLens 100k chứa 100K đánh giá của 943 người dùng cho khoảng 1682 bộ phim.
- Tập MovieLens 1M chứa 1M đánh giá của 6040 người dùng cho khoảng 3952 bộ phim
- Tất cả đánh giá được lưu trong file “ratings.dat” theo định dạng:
 - UserID::MovieID::Rating::Timestamp
- UserID là số nguyên trong khoảng 1 đến 6040.
- MovieID là số nguyên trong khoảng 1 đến 3952.
- Rating là số nguyên trong khoảng 1 đến 5.

Ví dụ:

```
1::1193::5::978300760
1::661::3::978302109
1::914::3::978301968
```

3.2 Độ đo kiểm nghiệm

Đề án sử dụng hai độ đo sau đây để kiểm nghiệm độ chính xác của phương pháp thực nghiệm:

Trung bình sai số tuyệt đối (MAE)[18] : đây là phương pháp phổ biến để đánh giá độ chính xác của dự đoán. Trung bình sai số tuyệt đối được tính theo công thức:

$$MAE = \frac{\sum_{i=1}^n |f_i - y_i|}{n} \quad (3.1)$$

Trung bình bình phương sai số (RMSE)[18] : đây là phương pháp tính trung bình của bình phương sai số theo công thức:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (f_i - y_i)^2}{n}} \quad (3.2)$$

3.3 Phương pháp thử nghiệm

Cách phân chia tập dữ liệu huấn luyện và kiểm nghiệm:

Tập dữ liệu được chia thành tập dữ liệu huấn luyện và tập dữ liệu kiểm nghiệm với kích thước lần lượt là 80% và 20%. Việc phân chia tập dữ liệu được thực hiện bởi lớp ProcessInput.

Tập dữ liệu huấn luyện và tập kiểm nghiệm được chia một cách ngẫu nhiên, dữ liệu mỗi lần chia là khác nhau, do đó có thể thực hiện chia nhiều lần và thực hiện chạy trên các tập khác nhau đó để kết quả được chính xác nhất.

3.4 Kết quả thử nghiệm

Thử nghiệm 1: Thử nghiệm phương pháp lọc cộng tác bằng phương pháp sử dụng SVD với độ giảm ma trận khác nhau:

Nhận xét: Đối với phương pháp lọc cộng tác sử dụng SVD độ chính xác của thuật toán phụ thuộc vào giá trị chọn k (kích thước ma trận đường chéo S).

3.4.1 Thử nghiệm về thời gian chạy

Thử nghiệm 1: Trong thử nghiệm này nhằm so sánh về thời gian chạy đồng thời đưa ra đánh giá về độ chính xác của 4 phương pháp.

- Mạng Bayes-MapReduce
- SVD MapReduce
- Mạng Bayes thường
- SVD thường

Thử nghiệm 2:

Thử nghiệm thời gian chạy trên tập MoviesLen 1M:

3.5 Đánh giá và so sánh

3.5.1 Đánh giá giữa các phương pháp lọc cộng tác

Về thời gian chạy: phương pháp lọc cộng tác dựa vào mô hình sử dụng mạng Bayes-MapReduce cho tốc độ tính toán nhanh hơn (27s) so phương pháp sử dụng SVD(109s).

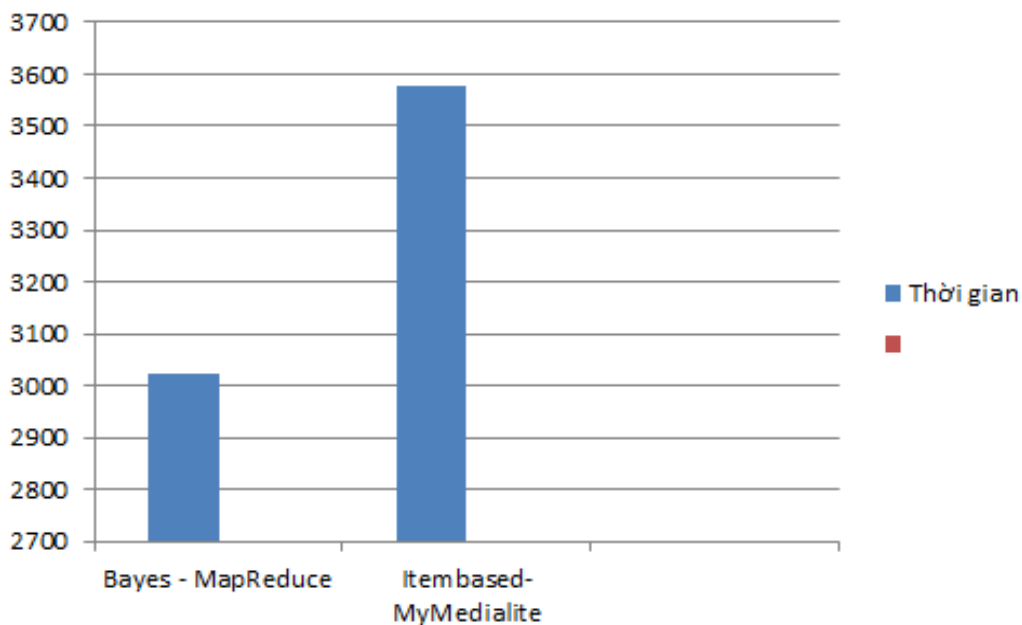
Về độ chính xác: phương pháp áp dụng SVD (MAE: 0.882287) cho kết quả chính xác cao hơn so với việc áp dụng Mạng Bayes (MAE: 1.007064)

3.5.2 Đánh giá thuật toán trước và sau khi dùng Hadoop-MapReduce

Về thời gian: theo bảng so sánh, rõ ràng thời gian chạy khi sử dụng MapReduce ngắn hơn nhiều so với không sử dụng MapReduce.

Về độ chính xác: do cùng bộ dữ liệu, cùng thuật toán, nên độ chính xác được đánh giá là tương đối giống nhau đối với cả 2 hệ thống có sử dụng và không sử dụng Hadoop-Mapreduce.

3.5.3 Đánh giá phương pháp tư vấn sử dụng MapReduce và MyMedialite



Hình 3-1: Đánh giá phương pháp tư vấn sử dụng MapReduce và MyMedialite

Theo số liệu trong mục trên thời gian chạy khi sử dụng thư viện MyMedialite lớn hơn khá nhiều khi sử dụng MapReduce. Trong điều kiện chỉ thực hiện trên một máy, không tận dụng được hết khả năng tính toán song song của Hadoop – MapReduce, kết quả này là

khá tốt. Như vậy, áp dụng Hadoop – MapReduce cho hệ tư vấn góp phần giải quyết được vấn đề mở rộng khi xây dựng hệ tư vấn cho dữ liệu lớn.

3.6 Kết luận chương 3

Chương 3 của đồ án đã trình bày quá trình thực nghiệm và đánh giá kết quả khi xây dựng hệ tư vấn dựa trên Hadoop – MapReduce. Quá trình thực nghiệm cho các kết quả khá tốt. Các kết quả thực nghiệm cho thấy tính khả thi của việc mở rộng hệ tư vấn sử dụng Hadoop – MapReduce và khẳng định tính đúng đắn của những vấn đề lý thuyết được nêu trong chương 2. Dựa trên phương pháp xây dựng hệ tư vấn đã được trình bày ở chương 2 và thực nghiệm ở chương 3, hoàn toàn có thể áp dụng phương pháp này vào thực tế để giải quyết vấn đề dữ liệu lớn hiện nay. Các kết quả này chưa phải là tối ưu nhất do điều kiện cơ sở vật chất để thực nghiệm còn hạn chế.

KẾT LUẬN

1. Kết quả đạt được

Luận văn đã trình bày phương pháp áp dụng MapReduce để mở rộng hệ tư vấn với phương pháp lọc cộng tác (CF) dựa vào mô hình, tập trung nghiên cứu 2 kỹ thuật cơ bản:

- Lọc cộng tác sử dụng định lý mạng Bayes
- Lọc cộng tác sử dụng công thức SVD

Kết quả thực nghiệm và đánh giá cho thấy:

- Trên cả 2 tập dữ liệu MovieLens 1M và 100K, kỹ thuật lọc cộng tác dựa vào mô hình Mạng Bayes và SVD khi được áp dụng MapReduce cho kết quả thời gian chạy tốt hơn rất nhiều độ chính xác tương tự so với phương pháp truyền thống.
- Trong 2 phương pháp đã sử dụng, phương pháp Mạng Bayes cho ta kết quả thời gian chạy tốt hơn so với phương pháp sử dụng SVD tuy nhiên về sai số tuyệt đối của phương pháp này là cao hơn.
- Phương pháp lọc cộng tác dựa vào mô hình sử dụng SVD tốc độ và độ chính xác của thuật toán còn phụ thuộc vào việc chọn K (kích thước của ma trận đường chéo S).

2. Hạn chế

Đồ án còn một số hạn chế là chưa thực nghiệm được phương pháp SVD trên tập dữ liệu lớn hơn do cơ sở vật chất không cho phép.

3. Hướng phát triển tiếp theo:

Nghiên cứu phương pháp phân tích ma trận (Matrix Factorization - MF) và áp dụng việc tính độ tương tự sản phẩm thông qua ma trận đại diện người dùng và sản phẩm từ đó đưa ra hệ tư vấn sẽ cho độ chính xác của thuật toán cao hơn.

Hadoop- MapReduce ngày càng trở nên phổ biến và được ứng dụng nhiều trong các lĩnh vực công nghệ cao hứa hẹn sẽ giải quyết vấn đề thời gian, đáp ứng thời gian thực của các hệ thống tư vấn thông minh đem lại tiện lợi cho con người.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng anh

- [1] Jean-Perrie Dijick (June 2013), “Oracle: Big Data For Enterprise”, *Oracle White Paper*;
- [2] Perera and Thilina Gunarathne (2013), “*Hadoop MapReduce Cookbook*”, Packt Publishing;
- [3] Hrishikesh Karambelkar (2013), “*Scaling Big Data With Hadoop and Solr*”, Pakt Publishing Ltd;
- [4] Ahmed Metwally, Christos Faloutsos (2012) “*V-SMART-Join: A Scalable MapReduce Framework for All-Pair Similarity Joins of Multisets and Vectors*” in Proceedings of the VLDB Endowment VLDB Endowment Homepage archive Volume 5 Issue 8, Pages 704-715.
- [5] Sebastian Schelter, Christoph Boden and Volker Markl (2012), “*Scalable Similarity-Based Neighborhood Methods with MapReduce*”, in RecSys '12 Proceedings of the sixth ACM conference on Recommender systems Pages 163-170, Technische Universität Berlin, Germany
- [6] Garry Turkington (2013), “*Hadoop Beginner’s Guide*”, Packt Publishing.
- [7] Tom White (May 2012), “*Hadoop: The definitive guide, Third edition*”, O'Reilly Media / Yahoo Press;
- [8] Byoungju Yang, Jaeseok Myung, Sang-goo Lee and Dongjoo Lee (2013), “*A MapReduce-based Filtering Algorithm for Vector Similarity Join*” in ICUIMC '13 Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication Article No. 71.

Website

- [9] http://en.wikipedia.org/wiki/Big_data. Truy cập ngày 2/3/2016
- [10] http://en.wikipedia.org/wiki/Cold_start. Truy cập ngày 5/3/2016
- [11] Hadoop: <http://hadoop.apache.org/>. Truy cập ngày 2/3/2016
- [12] <http://blogs.msdn.com/b/avkashchauhan/archive/2012/03/29/how-to-chain-multiple-mapreduce-jobs-in-hadoop.aspx>. Truy cập ngày 2/3/2016
- [13] Data set: <http://grouplens.org/datasets/movielens/>. Truy cập ngày 1/4/2016
- [14] <http://ankitasblogger.blogspot.com/2011/01/hadoop-cluster-setup.html>.
- [15] <http://mymedialite.net/>. Truy cập ngày 2/4/2016
- [16] http://en.wikipedia.org/wiki/Recommender_system. Truy cập ngày 5/5/2016
- [17] http://en.wikipedia.org/wiki/Mean_absolute_error. Truy cập ngày 12/5/2016
- [18] http://en.wikipedia.org/wiki/Root-mean-square_deviation. Truy cập ngày 15/5/2016