BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ÁP DỤNG CÁC KỸ THUẬT TRONG BIG DATA VÀO LƯU TRỮ DỮ LIỆU

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công Nghệ Thông Tin

HẢI PHÒNG - 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ÁP DỤNG CÁC KỸ THUẬT TRONG BIG DATA VÀO LƯU TRỮ DỮ LIỆU

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công Nghệ Thông Tin

Sinh viên thực hiện: Nguyễn Chí Thanh
Giáo viên hướng dẫn: Nguyễn Trịnh Đông
Mã số sinh viên: 1212101002

HẢI PHÒNG - 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
-------o0o-------

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Nguyễn Chí Thanh Mã sinh viên: 1212101002

Lớp: CT1601 Ngành: Công Nghệ Thông Tin

Tên đề tài: Áp dụng các kỹ thuật trong Big data vào lưu trữ dữ liệu

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung:

- Tìm hiểu về các thành phần công nghệ và quản lý Big data.
- Tìm hiểu mô hình xử lý dữ liệu phân tán MapReduce.
- Tìm hiểu hệ thống Hadoop.
- Đề ra phương pháp xây dựng hệ thống.
- Thử nghiệm với các công cụ để giải quyết bài toán.

b. Các yêu cầu cần giải quyết

- Nắm được các thành phần công nghệ Big data.
- Nắm được nguyên lý hoạt động mô hình Map Reduece.
- Nắm được quy trình hoạt động cơ bản của hệ thống Hadoop.
- Áp dụng kiến thức trong xây dựng phần mềm thử nghiệm.

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Nguyễn Trịnh Đông

Học hàm, học vị: Thạc sĩ

Cơ quan công tác: Trường Đại học Dân lập Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu về các thành phần công nghệ và quản lý Big data.
- Tìm hiểu mô hình xử lý dữ liệu phân tán MapReduce.
- Tìm hiểu hệ thống Hadoop.
- Đề ra phương pháp xây dựng hệ thống.
- Thử nghiệm với các công cụ để giải quyết bài toán.

Đề tài tốt nghiệp được giao ngày 03 tháng 10 năm 2016

Yêu cầu phải hoàn thành trước ngày 30 tháng 12 năm 2016

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên	Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngày tháng năm 2016

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....

.....

.....

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....

.....

.....

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn
(Điểm ghi bằng số và chữ)

.....

.....

Ngày tháng năm 2016

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. **Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế,…)**

2. **Cho điểm của cán bộ phản biện**

(Điểm ghi bằng số và chữ)

.....

.....

Ngày tháng năm 2016

Cán bộ chấm phản biện

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Qua thời gian học tập và nghiên cứu tại trường Đại học Dân lập Hải Phòng, đầu tiên em xin chân thành cảm ơn sâu sắc tới thầy giáo GS.TS NGƯT Trần Hữu Nghị hiệu trưởng nhà trường là người đã tạo điều kiện về cơ sở vật chất trang thiết bị giúp chúng em học tập và nghiên cứu trong thời gian qua.

Em xin chân thành cảm ơn tới tất cả thầy giáo, cô giáo trong nhà trường.Em xin chân thành cảm ơn các thầy giáo cô giáo trong Bộ môn Tin học trực tiếp giảng dạy cho em những kiến thức bổ ích.

Đặc biệt em xin chân thành cảm ơn thầy giáo Nguyễn Trịnh Đông trong thời gian làm tốt nghiệp vừa qua, thầy đã giành nhiều thời gian và tâm huyết để hướng dẫn em thực hiện đề tài này.

Dưới đây là kết quả của quá trình tìm hiểu và nghiên cứu mà em đã đạt được trong thời gian vừa qua. Mặc dù rất cố gắng và được thầy cô giúp đỡ nhưng do hiểu biết và kinh nghiệm của mình còn hạn chế nên có thể đây chưa phải là kết quả mà thầy cô mong đợi từ em. Em rất mong nhận được những lời nhận xét và đóng góp quý báu của thầy cô để bài luận văn của em được hoàn thiện hơn cũng như cho em thêm nhiều kinh nghiệm cho công việc sau này.

Em xin chân thành cảm ơn!

Hải Phòng, tháng 12 năm 2016

Sinh viên thực hiện

Nguyễn Chí Thanh

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ	Diễn giải
API	Application Programming Interfaces	Giao diện lập trình ứng dụng
XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
HDFS	Hadoop Distributed File System	Hệ thống phân tán tập tin của Hadoop
SaaS	Software as a Service	Triển khai phần mềm như là một dịch vụ
IaaS	Infrastructure as a Service	Triển khai cơ sở hạ tầng như là một dịch vụ
PaaS	Platform as a Service	Triển khai nền tảng như là một dịch vụ
DaaS	Data as a Service	Triển khai dữ liệu như là một dịch vụ
RDBMS	Relational Database Management System	Hệ thống quản lý cơ sở dữ liệu quan hệ
KVP	Key-Value pair	Cặp khóa – giá trị

LỜI MỞ ĐẦU

Sự phát triển của xã hội dẫn đến bùng nổ dữ liệu trong những thập niên gần đây. Những sản phẩm công nghệ mới đem lại nhiều tiện ích trong cuộc sống, được ứng dụng ở nhiều lĩnh vực thông tin truyền thông. Hệ thống thông tin điện tử, trực tuyến, các website của những doanh nghiệp tổ chức được phát triển mạnh mẽ góp phần tăng cường mối quan hệ, hợp tác ở nhiều lĩnh vực như văn hóa xã hội, khoa học công nghệ, y tế, giáo dục, giải trí… Con người có trong tay nhiều công cụ để chia sẻ thông tin qua blog, website, diễn đàn, các mạng xã hội trực tuyến như Facebook, Twitter, Youtube…Cách đây không lâu, vào năm 2000, chỉ mới có một phần tư lượng thông tin lưu trữ ở dạng kỹ thuật số trên thế giới. Ba phần tư còn lại được lưu trữ trên giấy tờ, phim, và các phương tiện analog khác. Nhưng do lượng dữ liệu kỹ thuật số bùng nổ quá nhanh – cứ 3 năm lại tăng gấp đôi – cục diện trên nhanh chóng đảo ngược. Ngày nay, chỉ dưới 2% tổng lượng thông tin chưa được chuyển sang lưu trữ ở dạng kỹ thuật số.

Tuy nhiên những phương thức lưu trữ dữ liệu đã bộc lộ rất nhiều hạn chế. Ngày nay khối lượng dữ liệu vô cùng lớn, kích cỡ lên đến hàng trăm terabyte cho đến petabyte chỉ cho một tập hợp dữ liệu. Cùng với đó khi mà hơn 80% dữ liệu sinh ra là phi cấu trúc (tài liệu, blog, hình ảnh, video, bài hát, dữ liệu cảm biến, thiết bị chăm sóc sức khỏe…) thì những phương pháp lưu trữ dữ liệu truyền thống không thể đảm đương được. Những phương pháp đó không cho phép liên kết và phân tích nhiều dạng dữ liệu khác nhau. Khối lượng dữ liệu gia tăng nhanh nhưng tốc độ xử lý dữ liệu (thu nhận, xử lý, đáp trả) mất rất nhiều thời gian trong khi nhu cầu của con người muốn xử lý được ngay dữ liệu tức thời (tính đến bằng mili giây). Điều đó thúc đẩy con người tạo ra một phương pháp và quản lý dữ liệu khác phù hợp hơn.

Và Big data ra đời đã đánh dấu một trang mới trong lịch sử phát triển công nghệ. Big data là một hệ thống dữ liệu liệu vô cùng lớn, đến mức không thể lưu trữ trong các hệ thống cơ sở dữ liệu truyền thống. Sự phức tạp và không thể định hình thành một thể thống nhất của dữ liệu Big data cũng là một nhân tố làm nó trở nên khó đồng bộ để lưu trữ trong một hệ thống cơ sở dữ liệu truyền thống. Dữ liệu được thu thập từ nhiều nguồn khác nhau bao gồm: dữ liệu không giới hạn từ internet, web 2.0, từ các thiết bị nghiên cứu (dữ liệu thiên văn, dịch vụ y tế…), dữ liệu từ các thiết bị thông minh (hay còn gọi là smart device). Do đó nó mang cấu trúc không cố định.Big data đã thể hiện được sức mạnh và tầm ảnh hưởng đến mọi lĩnh vực trong xã hội.

Trong đề tài này em sẽ trình bày về công nghệ quản lý Big data, mô hình xử lý dữ liệu phân tán Mapreduce và hệ thống Hadoop.

CHƯƠNG 1: CÔNG NGHỆ QUẢN LÝ DỮ LIỆU LỚN

1.1 Công nghệ nền tảng áp dụng cho Big data

1.1.1 Tìm hiểu các thành phần công nghệ Big data

Big data có số lượng dữ liệu lưu trữ rất lớn và thường lưu trữ các dòng dữ liệu có kiểu khác nhau ở tốc độ cao. Nhiều kĩ sư phần mềm dầy dặn kinh nghiệm và các nhà phát triển biết cách để nhắm đến một thậm chí là hai tình huống này hoàn toàn dễ dàng. Ví dụ, nếu bạn phải đối mặt dữ liệu lớn cần giải quyết cùng với yêu cầu khả năng chịu lỗi, bạn có thể lựa chọn triển khai cụm cơ sở dữ liệu dư thừa trong trung tâm cơ sở dữ liệu với cơ sở hạ tầng mạng rất nhanh. Tương tự, nếu yêu cầu là kết hợp nhiều loại dữ liệu khác nhau từ sự hiểu biết và các nguồn thông tin ẩn danh, lựa chọn có thể là xây dựng một mô hình di chuyển kho dữ liệu theo yêu cầu của khách hàng.

Tuy nhiên bạn có thể không có không đủ điều kiện để triển khai cụ thể. Khi bạn rời khỏi nơi mình có quyền điều khiển và kiểm soát dữ liệu chặt chẽ, bạn cần tạo ra một mô hình kiến trúc để nhắm đến loại môi trường lai. Môi trường mới này đòi hỏi kiến trúc này phải hiểu về tính chất của Big data và yêu cầu để đưa kiến thức vào giải pháp kinh doanh. Trong chương này chúng ta sẽ tìm hiểu về kiến trúc liên quan đến dữ liệu lớn.

1.1.1.1 Sự dư thừa cơ sở hạ tầng vật lý

Ở cấp thấp nhất là cơ sở hạ tầng vật lý như phần cứng, mạng, ... Công ty của bạn có thể đã có trung tâm dữ liệu hoặc được đầu tư cơ sở vật chất nên bạn muốn tìm một cách để sử dụng dữ liệu hiện hữu. Thi hành dữ liệu lớn có yêu cầu rất cụ thể trên tất cả các phần tử trong kiến trúc tham khảo, vì thế bạn cần kiểm tra những yêu cầu này về nền tảng cơ sở lớp – lớp để đảm bảo sẽ thực hiện và nâng cấp theo đúng yêu cầu của công ty. Điều quan trọng là phải thực hiện theo đúng nguyên tắc. Mức độ ưu tiên theo danh sách nguyên tắc này bao gồm:

- Performance (Hiệu năng): Thực thi thường được tiến hành nối đuôi nhau dựa trên một giao dịch hay một câu hỏi có yêu cầu ở tốc độ rất nhanh (hiệu suất cao), do vậy chi phí cho cơ sở hạ tầng thường rất tốn kém.

- Availability (Tính sẵn có): Bạn có cần đảm bảo thời gian dịch vụ 100%? Công ty của bạn có thể chờ được bao lâu trong trường hợp dịch vụ gián đoạn hoặc không đạt yêu cầu? Cơ sở hạ tầng đảm bảo tính sẵn sàng cao cũng rất tốn kém.
- Scalability (Khả năng mở rộng): Cơ sở hạ tầng của bạn cần được mở rộng như thế nào? Dung lượng đĩa cần bao nhiêu để đảm bao yêu cầu trong thời điểm hiện tại và tương lai?
- Flexibility (Linh hoạt): Bạn thêm tài nguyên vào cơ sở hạ tầng sớm nhất là khi nào? Cơ sở hạ tầng khôi phục sau thất bại nhanh mức nào? Cơ sở hạ tầng đạt mức linh hoạt nhất rất tốn kém nhưng có thể kiểm soát bằng dịch vụ điện toán đám mây, nơi bạn chỉ trả tiền cho những gì bạn thực sự sử dụng.
- Cost (Chi phí): Bạn có thể đủ khả năng chi trả cho cái gì? Bởi cơ sở hạ tầng là tập hợp của rất nhiều thành phần, bạn có thể mua hệ thống mạng tốt nhất và tiết kiệm tiền cho việc lưu trữ hoặc ngược lại. Bạn cần lập yêu cầu đối với mỗi lĩnh vực trong hoàn cảnh ngân sách cụ thể và chi trả cho những nơi cần thiết.

Big data hoàn toàn tập trung vào tốc độ xử lý cao, khả năng lưu trữ dữ liệu lớn và đa dạng nên cơ sở vật chất theo nghĩa đen sẽ quyết định đến sự thành bại của việc thực hiện. Hầu hết việc thực thi Big data cần ở mức độ sẵn sàng cao nên mạng, server và bộ lưu trữ phải vừa có thể thay đổi (mở rộng, thu hẹp), vừa phải tránh dư thừa. Khả năng thay đổi và dư thừa có mối quan hệ với nhau. Về bản chất luôn có lý do khiến cho ngay cả hệ thống mạng tốt nhất cũng có thể bị lỗi như là một trục trặc phần cứng. Do đó công nghệ dự phòng đảm bảo rằng trục trặc này sẽ không gây ra ngưng trệ.

a. Sự dư thừa mạng

Mạng nên dư thừa và phải có đủ khả năng thích ứng trước số lượng và tốc độ của dữ liệu ra vào trong mạng lưới giao thông trên mạng. Khi bạn bắt đầu làm về Big data, mạng là phần thiết yếu trong chiến lược tin học của bạn. Nó là cơ sở để gia tăng số lượng và vận tốc.

Những người thiết kế cơ sở hạ tầng nên lập kế hoạch cho hệ thống mạng. Khi lưu lượng giao thông mạng thay đổi (tăng, giảm), có sự ảnh hưởng tới tài sản vật chất liên quan đến việc triển khai thực hiện. Cơ sở hạ tầng của bạn nên đưa ra khả năng giám sát giúp người điều hành có thể phản ứng khi lượng tài nguyên tăng lên dẫn đến thay đổi khối lượng công việc.

b. Quản lý phần cứng: Bộ lưu trữ và Server

Phần cứng (bộ lưu trữ và server) phải có đủ tốc độ và năng lực xử lý tất cả các khả năng của Big data. Đó là sử dụng một ít để có mạng tốc độ cao cùng với các server chậm bởi vì các máy chủ có thể trong tình trạng thắt nút cổ chai. Tuy nhiên một bộ lưu trữ dữ liệu nhanh và tính toán các máy chủ có thể vượt qua hiệu suất mạng. Tất nhiên, sẽ không có gì hoạt động tốt nếu hiệu suất mạng thấp và kém chất lượng.

c. Hoạt động cơ sở hạ tầng

Một thiết kế quan trọng cần quan tâm là quản lý hoạt động cơ sở hạ tầng. Mức cao nhất về hiệu suất và tính linh hoạt chỉ xuất hiện trong một môi trường được quản lý tốt. Các nhà quản lý dữ liệu có thể dự đoán và ngăn chặn các thất bại thảm hại, như vậy giữ được sự toàn vẹn của dữ liệu và duy trì quy trình nghiệp vụ.

1.1.1.2 Cơ sở hạ tầng an ninh

An ninh và bảo mật trong Big data tương tự như các yêu cầu về môi trường dữ liệu thông thường. Các yêu cầu về an ninh phải được liên kết chặt chẽ với nhu cầu nghiệp vụ cụ thể. Một số thách thức phát sinh khi Big data trở thành một phần của chiến lược bao gồm:

- Truy cập dữ liệu: Khả năng truy cập dữ liệu của người dùng để tính toán dữ liệu lớn có cùng mức độ yêu cầu kĩ thuật như triển khai dữ liệu không lớn. Dữ liệu cần có chỉ dành cho những người có nhu cầu nghiệp vụ để kiểm tra hoặc tương tác với nó. Hầu hết các nền tảng lưu trữ dữ liệu có hệ thống an ninh nghiêm ngặt và thường được tăng cường với mộ khả năng nhận dạng hợp nhất, cung cấp truy cập thích hợp trên nhiều lớp của kiến trúc.
- Truy cập ứng dụng: Truy cập dữ liệu ứng dụng cũng tương đối đơn giản từ góc độ kĩ thuật. Hầu hết các giao diện lập trình ứng dụng (API) cung cấp bảo vệ từ việc sử dụng trái phép hoặc truy cập. Mức độ bảo vệ thích hợp nhất cho triển khai thực hiện Big data.

- Mã hóa dữ liệu: Mã hóa dữ liệu là thách thức lớn nhất về bảo mật trong môi trường Big data. Trong môi trường truyền thống, mã hóa và giải mã dữ liệu thực sự cần nguồn lực lớn của hệ thống. Với khối lượng, tốc độ và sự đa dạng của Big data, vấn đề này càng khó khăn hơn. Mã hóa dữ liệu là cách tốt nhất để cung cấp khả năng tính toán nhiều hơn và nhanh hơn. Tuy nhiên điều này đi kèm với một bảng giá. Do vậy cần xác định phần dữ liệu nào cần bảo mật và để mã hóa các mục cần thiết.
- Phát hiện đe dọa: Bao gồm các thiết bị di động và các mạng xã hội theo cấp số nhân tăng cả số lượng dữ liệu và các mối đe dọa an ninh. Do đó điều quan trọng là các tổ chức có cách tiếp cận vòng ngoài an ninh.

1.1.1.3 Giao diện ứng dụng và Internet

Cơ sở hạ tầng vật lý cho phép tất cả mọi thứ và cơ sở hạ tầng an ninh bảo vệ tất cả các yếu tố trong môi trường Big data. Các cấp độ tiếp theo là các giao diện mà cung cấp truy cập hai chiều cho tất cả các thành phần của Stack – từ các ứng dụng doanh nghiệp đến dữ liệu từ Internet. Một phần quan trọng của việc thiết kế các giao diện này là tạo ra một cấu trúc phù hợp có thể chia sẻ cả bên trong lẫn bên ngoài công ty cũng như các đối tác trong kinh doanh.

Trong nhiều thập kỷ, các lập tình viên đã sử dụng API để cung cấp truy cập đến và đi từ việc triển khai phần mềm. Các nhà cung cấp công cụ và công nghệ sẽ đi đến độ dài lớn để đảm bảo rằng nó là một nhiệm vụ tương đối đơn giản để tạo ra các ứng dụng mới sử dụng sản phẩm của họ. Nó cần thiết cho các chuyên gia CNTT để tạo ra tùy chỉnh hoặc các API độc quyền cho công ty. Bạn cần làm điều này cho lợi thế cạnh tranh, một số nhu cầu nghiệp vụ và đó không phải là một nhiệm vụ đơn giản. API cần được lưu trữ và duy trì để bảo toàn giá trị cho doanh nghiệp. Vì lý do này, một số công ty lựa chọn để sử dụng bộ công cụ API để có được một bước nhảy về hoạt động quan trọng này.

Bộ công cụ API có một vài ưu điểm so với các API phát triển nội bộ. Đầu tiên là bộ công cụ API là sản phẩm được tạo ra, được quản lý và duy trì bởi một bên thứ ba dộc lập. Thứ hai, chúng được thiết kế để giải quyết một yêu cầu kỹ thuật cụ thể. Nếu bạn cần các API cho ứng dụng web hoặc ứng dụng di động, có nhiều lựa chọn cho bạn bắt đầu.

Bởi vì thu thập dữ liệu và chuyển động có đặc điểm rất giống nhau, có thể thiết kế một bộ dịch vụ để thu thập, làm sạch, biến đổi, chuẩn hóa và lưu trữ các dữ liệu lớn trong hệ thống lưu trữ. Để tạo ra sự linh hoạt khi cần thiết, các nhà máy được điều khiển cùng với mô tả giao diện được viết bằng Extensible Markup Language (XML). Mức độ này cho phép các giao diện cụ thể được tạo ra một cách dễ dàng và nhanh chóng mà không cần phải xây dựng các dịch vụ cụ thể cho từng nguồn dữ liệu.

1.1.1.4 Cơ sở dữ liệu hoạt động

Ở lõi của môi trường Big data là những cơ sở dữ liệu chứa các phần dữ liệu liên quan đến công ty của bạn. Không có sự lựa chọn duy nhất đúng liên quan đến ngôn ngữ cơ sở dữ liệu. Mặc dù SQL là ngôn ngữ thông dụng truy vấn cơ sở dữ liệu nhưng các ngôn ngữ khác cũng có thể cung cấp một cách hiệu quả hơn. Ví dụ nếu bạn sử dụng một mô hình quan hệ, bạn có thể sử dụng SQL để truy vấn nó. Tuy nhiên bạn có thể sử dụng ngôn ngữ khác như Python hay Java. Nó là rất quan trọng để hiểu các dạng dữ liệu có thể đang bị điều khiển bởi cơ sở dữ liệu và hỗ trợ các hành vi giao dịch đúng. Nhà thiết kế cơ sở dữ liệu mô tả hành vi này với ACID. Nó tượng trưng cho:

- Atomicity (Mức nguyên tử): Một giao dịch là “tất cả hoặc không có gì” khi nó ở mức nguyên tử. Nếu bất cứ phần nào của giao dịch hoặc những thất bại của hệ thống ở mức cơ bản thì toàn bộ giao dịch sẽ thất bại.
- Consistency (Tính nhất quán): Chỉ những giao dịch với dữ liệu hợp lệ sẽ được thực hiện trên cơ sở dữ liệu. Nếu dữ liệu bị hỏng hoặc không phù hợp thì các giao dịch sẽ không được hoàn thành và dữ liệu sẽ không được lưu vào cơ sở dữ liệu.
- Isolation (Độc lập): Nhiều giao dịch diễn ra đồng thời sẽ không tác động vào nhau. Tất cả các giao dịch hợp lệ sẽ thực hiện cho đến khi hoàn thành và theo thứ tự chúng được gửi đến để xử lý.
- Durability (Độ bền vững): Sau khi dữ liệu từ các giao dịch được ghi vào cơ sở dữ liệu, nó sẽ nằm ở đó mãi mãi.

1.1.1.5 Tổ chức dịch vụ và công cụ dữ liệu

Tổ chức dịch vụ dữ liệu và các công cụ xác thực, lắp ráp các phần khác nhau thành phần dữ liệu lớn đưa vào bộ sưu tập theo ngữ cảnh có liên quan. Bởi vì là dữ liệu lớn nên kỹ thuật đã tiến hóa để xử lý dữ liệu hiệu quả và liên tục.

Tổ chức dịch vụ dữ liệu, trong thực tế là một hệ sinh thái của các công cụ và công nghệ có thể được sử dụng để thu thập và tổng hợp số liệu. Như vậy các công cụ cần tích hợp, dịch thuật, chuẩn hóa, phạm vi. Công nghệ trong lớp này bao gồm:

- Một hệ thống tập tin phân phối: cần thiết để thích ứng với sự phân tách của các luồng dữ liệu và cung cấp khả năng lưu trữ.
- Dịch vụ chuyển đổi cấu trúc: cần thiết cho việc lưu trữ dữ liệu bền vững và các cuộc gọi thủ tục đa ngôn ngữ từ xa (RPC).
- Dịch vụ điều phối: cần thiết cho việc xây dựng ứng dụng phân tán.
- Trích đoạn, biến đổi, tải (ETL): cần thiết cho việc tải và chuyển đổi cấu trúc – phi cấu trúc vào Hadoop.
- Dịch vụ tiến độ công việc: cần thiết cho việc lập kế hoạch và cung cấp một cấu trúc để đồng bộ hóa yếu tố quá trình trên lớp.

1.1.1.6 Kho dữ liệu phân tích

Các kho dữ liệu từ lâu đã được coi là các kỹ thuật chính mà các tổ chức sử dụng để tối ưu hóa dữ liệu để giúp các nhà sản xuất quyết định. Thông thường, các kho dữ liệu bình thường thu thập từ nhiều nguồn khác nhau và lắp ráp để tạo điều kiện phân tích của doanh nghiệp. Kho dữ liệu đơn giản hóa việc tạo ra các báo cáo và trực quan của các mục dữ liệu khác nhau. Chúng thường được tạo ra từ cơ sở dữ liệu quan hệ, cơ sở dữ liệu đa chiều, các tập tin phẳng, và cơ sở dữ liệu đối tượng - về cơ bản của bất kỳ kiến trúc lưu trữ. Trong một môi trường truyền thống, nơi hiệu suất có thể không phải là ưu tiên cao nhất, sự lựa chọn của các công nghệ cơ bản được điều khiển bởi các yêu cầu cho việc phân tích, báo cáo và trực quan của dữ liệu công ty.

Là một tổ chức dữ liệu và nó luôn sẵn sàng để phân tích, triển khai hầu hết kho dữ liệu được lưu giữ qua hàng loạt quá trình. Vấn đề là kho dữ có thể không đủ cho nhiều ứng dụng dữ liệu lớn. Sự căng thẳng áp đặt bởi các dòng dữ liệu tốc độ cao có khả năng đòi hỏi một cách tiếp cận thời gian thực hơn với kho dữ liệu lớn. Điều này không có nghĩa là bạn sẽ không được tạo ra và cung cấp một kho dữ liệu và phân tích một siêu kho dữ liệu với các quá trình thực thi. Thay vào đó, bạn có thể sẽ có kho dữ liệu hoặc siêu kho dữ liệu, hiệu suất và quy mô sẽ phản ánh kịp thời yêu cầu của các nhà phân tích và ra quyết định.

Bởi vì kho nhiều dữ liệu và siêu kho dữ liệu được bao gồm các dữ liệu thu thập từ nhiều nguồn khác nhau trong công ty, các chi phí liên quan đến việc làm sạch và bình thường hóa của dữ liệu cũng phải được giải quyết. Với dữ liệu lớn, bạn tìm thấy một số khác biệt quan trọng:

- Dòng dữ liệu truyền thống (từ giao dịch, ứng dụng, vv) có thể sản xuất rất nhiều dữ liệu khác nhau.
- Hàng chục các nguồn dữ liệu mới cũng tồn tại, một trong số chúng cần một số thao tác xử lý trước khi nó có thể được dùng cho doanh nghiệp.
- Nguồn nội dung cũng sẽ cần phải được làm sạch, và có những yêu cầu kỹ thuật khác nhau để có thể sử dụng với dữ liệu có cấu trúc.

Trong lịch sử, các nội dung của kho dữ liệu và siêu kho dữ liệu được tổ chức và giao cho các nhà lãnh đạo kinh doanh phụ trách chiến lược và quy hoạch. Với dữ liệu lớn, chúng ta đang nhìn thấy các đội được tận dụng dữ liệu cho việc ra quyết định. Nhiều triển khai dữ liệu lớn cung cấp khả năng thời gian thực, vì vậy doanh nghiệp sẽ có thể cung cấp nội dung cho phép các cá nhân với vai trò hoạt động để giải quyết các vấn đề như hỗ trợ khách hàng, cơ hội kinh doanh, và thực thi dịch vụ trong thời gian thực gần. Bằng cách này, dữ liệu lớn giúp di chuyển hành động từ các văn phòng trở lại văn phòng.

1.1.1.7 Phân tích Big data

Hiện tại công cụ phân tích kỹ thuật và sẽ rất hữu ích trong việc đưa ra ý nghĩa của dữ liệu lớn. Tuy nhiên, có một nhược điểm. Các thuật toán là một phần của những công cụ này có để có thể làm việc với một lượng lớn có khả năng thời gian thực và dữ liệu khác nhau. Các cơ sở hạ tầng sẽ cần phải được thực hiện để hỗ trợ. Các nhà cung cấp các công cụ phân tích cũng cần phải đảm bảo rằng các thuật toán của họ làm việc qua việc triển khai phân phối.

- Báo cáo và biểu đồ: Những công cụ này cung cấp một đại diện "thân thiện" của thông tin từ các nguồn khác nhau. Mặc dù là một trụ cột trong thế giới dữ liệu truyền thống, chúng vẫn đang phát triển đối với dữ liệu lớn. Một số công cụ đang được sử dụng là loại mới của cơ sở dữ liệu gọi chung là NoSQL.

- Hình dung: Những công cụ này là bước tiếp theo trong quá trình báo cáo. Các đầu ra có xu hướng được tương tác cao và năng động trong tự nhiên. Một khác biệt quan trọng giữa các báo cáo đầu ra và hình dung là hình ảnh động. Người dùng doanh nghiệp có thể xem các thay đổi trong các dữ liệu sử dụng một loạt các kỹ thuật hiển thị khác nhau, bao gồm bản đồ tâm trí, bản đồ nhiệt, bản đồ thông tin, và sơ đồ kết nối. Thông thường, báo cáo và hình dung xảy ra ở phần cuối của các hoạt động kinh doanh. Mặc dù các dữ liệu có thể được nhập khẩu vào một công cụ khác để tính toán thêm, kiểm tra, đây là bước cuối cùng.
- Phân tích: Những công cụ tiếp cận vào kho dữ liệu và xử lý dữ liệu cho người dùng.

1.1.1.8 Những ứng dụng của Big data

Tuỳ chỉnh và bên thứ ba ứng dụng cung cấp một phương pháp khác để chia sẻ và kiểm tra các nguồn dữ liệu lớn. Mặc dù tất cả các lớp của kiến trúc tham khảo rất quan trọng trong quyền riêng của họ, lớp này là nơi gồm hầu hết đổi mới và sáng tạo.

Giống như bất kỳ sáng kiến phát triển ứng dụng nào, việc tạo ra các ứng dụng dữ liệu lớn sẽ yêu cầu cơ cấu, tiêu chuẩn, sự chặt chẽ, và API được xác định rõ. Hầu hết các ứng dụng kinh doanh muốn tận dụng dữ liệu lớn sẽ cần phải đăng ký để API trên toàn bộ stack. Nó có thể là cần thiết để xử lý dữ liệu thô từ các cửa hàng dữ liệu cấp thấp và kết hợp các dữ liệu thô với lượng dữ liệu được tổng hợp từ các kho hàng.

Big data di chuyển nhanh chóng và thay đổi trong chớp mắt, vì vậy nhóm phát triển phần mềm cần nhanh chóng tạo ra các ứng dụng phù hợp để giải quyết những thách thức kinh doanh của thời điểm này. Các công ty có thể cần phải suy nghĩ về việc tạo phát triển nhanh chóng đáp ứng với những thay đổi trong môi trường kinh doanh bằng cách tạo và triển khai các ứng dụng theo yêu cầu. Trong thực tế, nó có thể thích hợp hơn để nghĩ về những ứng dụng như "tùy chỉnh bán" bởi vì chúng liên quan đến lắp ráp hơn thực tế mã hóa ở mức độ thấp.

1.1.2 Ảo hóa và hỗ trợ tính toán phân tán

Ảo hóa là một công nghệ nền tảng áp dụng đối với việc thực hiện điện toán đám mây và dữ liệu lớn. Nó cung cấp cơ sở cho nhiều thuộc tính nền tảng cần thiết để truy cập, lưu trữ, phân tích và quản lý các thành phần tính toán phân tán trong môi trường dữ liệu lớn. Ảo hóa - quá trình sử dụng tài nguyên máy tính bắt chước các nguồn lực khác - được đánh giá cao vì khả năng của nó để tăng nguồn lực CNTT hiệu quả và khả năng mở rộng. Một ứng dụng chính của ảo

hóa là hợp nhất máy chủ, giúp các tổ chức nâng cao việc sử dụng các máy chủ vật lý và có khả năng tiết kiệm chi phí cơ sở hạ tầng. Tuy nhiên, có thể tìm thấy nhiều lợi ích của ảo hóa. Các công ty mà ban đầu chỉ tập trung vào công nghệ ảo hóa máy chủ hiện nay đang nhận ra rằng nó có thể được áp dụng trên cơ sở hạ tầng CNTT toàn bộ, bao gồm cả các phần mềm, lưu trữ và hệ thống mạng.

1.1.2.1 Tìm hiểu những vấn đề cơ bản của ảo hóa

Ảo hóa phân tách nguồn lực và dịch vụ từ các môi trường phân phối vật lý cơ bản, cho phép bạn tạo ra nhiều hệ thống ảo trong một hệ thống vật lý duy nhất. Một trong những lý do chính mà các công ty đã thực hiện ảo hóa là để cải thiện hiệu suất và hiệu quả xử lý kết hợp đa dạng của các khối lượng công việc. Thay vì chỉ định một nhóm dành riêng cho các nguồn lực vật chất để mỗi nhóm thực hiện nhiệm vụ, một nhóm gộp tài nguyên ảo để có thể nhanh chóng phân bổ trên tất cả các khối lượng công việc. Sự phụ thuộc vào biển tài nguyên ảo cho phép các công ty cải thiện độ trễ. tăng tốc độ cung cấp dịch vụ và hiệu quả. Đây là một chức năng của bản chất phân tán của các môi trường ảo hóa và giúp cải thiện tổng thể thời gian tới giá trị.

Sử dụng một bộ phân phối các nguồn lực vật chất, chẳng hạn như máy chủ, một cách linh hoạt và hiệu quả mang lại lợi ích đáng kể trong điều kiện cần tiết kiệm chi phí và cải thiện năng suất. Việc thực hành có nhiều lợi ích, bao gồm những điều sau đây:

- Ảo hóa các nguồn lực vật chất (như máy chủ, lưu trữ, và các mạng) cho phép cải thiện đáng kể trong việc sử dụng các nguồn lực này.
- Ảo hóa cho phép cải tiến kiểm soát việc sử dụng và hiệu suất của nguồn lực CNTT.
- Ảo hóa có thể cung cấp một mức độ tự động hóa và tiêu chuẩn hóa để tối ưu hóa môi trường máy tính.
- Ảo hóa cung cấp nền tảng cho điện toán đám mây.

Mặc dù có thể sử dụng ảo hóa để bổ sung nguồn lực song ảo hóa không phải là không tốn chi phí. Tài nguyên ảo phải được quản lý để đảm bảo an toàn. Một hình ảnh có thể là một kỹ thuật cho kẻ lạ xâm nhập truy cập trực tiếp vào trong hệ thống. Nếu công ty không có một quá trình xóa những hình ảnh không sử dụng, hệ thống sẽ không còn hoạt động hiệu quả.

a. Tầm quan trọng của ảo hóa dữ liệu lớn

Giải quyết thách thức Big data thường đòi hỏi việc quản lý khối lượng lớn các cửa hàng dữ liệu phân tán cao cùng với việc sử dụng các ứng dụng tính toán và dữ liệu chuyên sâu. Do đó, bạn cần một môi trường CNTT có hiệu quả cao để hỗ trợ dữ liệu lớn. Ảo hóa cung cấp mức độ gia tăng của hiệu quả để làm nền tảng dữ liệu lớn thành hiện thực. Mặc dù vậy, ảo hóa là kỹ thuật không phải là một yêu cầu để phân tích dữ liệu lớn, khung phần mềm như MapReduce, được sử dụng trong các môi trường dữ liệu lớn, có hiệu quả hơn trong một môi trường ảo hóa.

Ảo hóa có ba đặc điểm hỗ trợ khả năng mở rộng và hoạt động hiệu quả cần thiết cho môi trường dữ liệu lớn:

- Phân vùng: Trong ảo hóa, nhiều ứng dụng và hệ điều hành được hỗ trợ trong một hệ thống vật lý duy nhất bằng cách phân vùng (chia) các nguồn lực sẵn có.
- Cô lập: Mỗi máy ảo được phân tách từ hệ thống vật lý máy chủ và máy ảo khác. Nếu trong trường hợp máy ảo treo, các máy ảo khác và các hệ thống máy chủ không bị ảnh hưởng. Ngoài ra, dữ liệu không được chia sẻ giữa máy ảo và máy khác.
- Đóng gói: Một máy ảo có thể được cho là đại diện (và thậm chí lưu trữ). Đây là một tập tin duy nhất, vì vậy bạn có thể xác định nó một cách dễ dàng dựa vào các dịch vụ mà nó cung cấp. Ví dụ, tập tin có chứa các quá trình đóng gói có thể là một dịch vụ kinh doanh hoàn chỉnh. Máy ảo đóng gói này có thể được trình bày cho một ứng dụng như một thực thể hoàn chỉnh. Vì vậy, đóng gói có thể bảo vệ mỗi ứng dụng để nó không can thiệp vào một ứng dụng khác.

Một trong những yêu cầu quan trọng nhất để thành công với dữ liệu lớn là có phải đủ năng lực thực hiện để hỗ trợ việc phân tích khối lượng lớn và đa dạng các loại dữ liệu. Khi bạn bắt đầu với môi trường nền tảng như Hadoop MapReduce và, điều quan trọng là bạn có một cơ sở hạ tầng hỗ trợ, có thể mở rộng quy mô. Ảo hóa thêm hiệu quả ở mọi lớp của cơ sở hạ tầng CNTT. Áp dụng ảo hóa trên môi trường của bạn sẽ giúp đỡ để đạt được khả năng mở rộng cần thiết để phân tích dữ liệu lớn.

Toàn bộ môi trường CNTT cần phải được tối ưu hóa ở mỗi lớp, từ mạng vào cơ sở dữ liệu, lưu trữ và máy chủ. Nếu bạn chỉ ảo hóa máy chủ của bạn, bạn có thể gặp vướng mắc từ các yếu tố cơ sở hạ tầng khác như lưu trữ và mạng. Nếu bạn chỉ tập trung vào việc ảo hóa là một yếu tố của cơ sở hạ tầng của bạn, bạn ít có khả năng để đạt được độ trễ và hiệu và nhiều khả năng mất chi phí cao hơn và mắc phải những rủi ro an ninh.

Thực tế là hầu hết các tổ chức không cố gắng để ảo hóa tất cả các yếu tố của cơ sở hạ tầng của họ tại một thời gian. Nhiều tổ chức bắt đầu với ảo hóa máy chủ và đạt được một mức độ nhất định của những cải tiến hiệu quả. Các yếu tố khác có thể được ảo hóa khi cần thiết để tiếp tục cải thiện hiệu suất hệ thống tổng thể và hiệu quả. Sau đây mô tả cách ảo hóa của mỗi nguyên tố có trong môi trường CNTT - máy chủ, lưu trữ, các ứng dụng, dữ liệu, mạng, bộ vi xử lý, bộ nhớ, và các dịch vụ - có thể có một tác động tích cực trên phân tích dữ liệu lớn.

b. Ảo hóa máy chủ

Trong ảo hóa máy chủ, một máy chủ vật lý được phân chia thành nhiều máy chủ ảo. Các phần cứng và các tài nguyên của một máy - bao gồm bộ nhớ truy cập ngẫu nhiên (RAM), CPU, ổ cứng, và điều khiển mạng - có thể được ảo hóa (logic split) vào một loạt các máy ảo mà mỗi máy chạy các ứng dụng riêng của mình và hệ điều hành. Một máy ảo (VM) là một đại diện phần mềm của một máy vật lý có thể thực hiện các chức năng tương tự như các máy vật lý. Một lớp mỏng của phần mềm được đưa vào các phần cứng có chứa một màn hình máy ảo, hoặc hypervisor. Hypervisor có thể được coi như là công nghệ quản lý lưu lượng giữa các máy ảo và máy vật lý.

Ảo hóa máy chủ sử dụng hypervisor để cung cấp hiệu quả trong việc sử dụng các nguồn lực vật chất. Cài đặt, cấu hình và công việc hành chính có liên quan đến việc thiết lập các máy ảo. Điều này bao gồm quản lý giấy phép, quản lý mạng và quản lý khối lượng công việc, cũng như kế hoạch năng lực.

Máy chủ ảo hóa giúp đảm bảo rằng nền tảng có thể mở rộng khi cần thiết để xử lý khối lượng lớn và đa dạng các loại dữ liệu trong phân tích dữ liệu lớn. Bạn không thể biết được mức độ âm lượng hoặc nhiều loại dữ liệu có cấu trúc và không có cấu trúc cần thiết trước khi bạn bắt đầu phân tích của bạn. Chính điều này làm cho nhu cầu về máy chủ ảo hóa lớn hơn, cung cấp môi trường của bạn với khả năng để đáp ứng nhu cầu bất ngờ để xử lý tập dữ liệu rất lớn.

Ngoài ra, máy chủ ảo hóa cung cấp nền tảng cho phép rất nhiều các dịch vụ đám mây được sử dụng như nguồn dữ liệu trong phân tích Big data. Ảo hóa làm tăng hiệu quả của các đám mây mà làm cho nhiều hệ thống phức tạp dễ dàng để tối ưu hóa hơn. Các tổ chức có hiệu quả hoạt động và tối ưu hóa để có thể truy cập dữ liệu mà trước đây hoặc là không có hoặc rất khó để thu thập. Các công ty có thể tích hợp thông tin này với các dữ liệu sản phẩm bán hàng nội bộ và để đạt được cái nhìn sâu sắc vào sở thích của khách hàng.

c. Ứng dụng ảo hóa

Ảo hóa cơ sở hạ tầng ứng dụng cung cấp một cách hiệu quả để quản lý các ứng dụng trong hoàn cảnh với nhu cầu khách hàng khác nhau. Các ứng dụng được gói gọn mà loại bỏ sự phụ thuộc của nó từ hệ thống máy tính vật lý bên dưới. Điều này giúp nâng cao khả năng quản lý tổng thể và tính di động của các ứng dụng. Ngoài ra, các ứng dụng cơ sở hạ tầng phần mềm ảo hóa thường cho phép cho việc hệ thống hóa các chính sách sử dụng kinh doanh và kỹ thuật để đảm bảo rằng mỗi ứng dụng của bạn thúc đẩy nguồn tài nguyên ảo và vật lý trong một cách dự đoán được. Hiệu quả có được là bởi vì bạn có thể dễ dàng phân phối các nguồn lực CNTT theo các giá trị kinh doanh tương đối từ các ứng dụng của bạn. Nói cách khác, các ứng dụng quan trọng nhất của bạn có thể nhận được ưu tiên hàng đầu để rút ra từ máy tính sẵn có và khả năng lưu trữ khi cần thiết.

Ảo hóa cơ sở hạ tầng ứng dụng sử dụng kết hợp với ảo hóa máy chủ có thể giúp đảm bảo rằng các thỏa thuận kinh doanh dịch vụ cấp (SLAs) được đáp ứng. Ảo hóa máy chủ theo dõi CPU và bộ nhớ sử dụng, nhưng không tính đến sự khác biệt về ưu tiên kinh doanh khi phân bổ nguồn lực. Ví dụ, bạn có thể yêu cầu tất cả các ứng dụng đang được điều trị với cùng một ưu tiên kinh doanh. Bằng cách thực hiện ảo hóa cơ sở hạ tầng ứng dụng ngoài việc ảo hóa máy chủ, bạn có thể đảm bảo rằng các ứng dụng ưu tiên cao nhất có quyền truy cập ưu tiên hàng đầu đến các nguồn dữ liệu.

Nền tảng dữ liệu lớn được thiết kế để hỗ trợ phân phối, các ứng dụng dữ liệu chuyên sâu sẽ chạy tốt hơn và nhanh hơn trong một môi trường ảo. Điều này không có nghĩa rằng bạn sẽ muốn ảo hóa tất cả các ứng dụng liên quan đến dữ liệu lớn. Ví dụ, một ứng dụng phân tích văn bản có thể chạy tốt nhất trong một môi trường khép kín và ảo hóa sẽ không thêm bất kỳ lợi ích gì.

d. Ảo hóa mạng

Mạng ảo hóa cung cấp một cách hiệu quả để sử dụng mạng như một biển tài nguyên kết nối. Mạng được ảo hóa cũng tương tự như trong công nghệ vật lý khác. Thay vì dựa vào các mạng vật lý cho việc quản lý giao thông giữa các kết nối, bạn có thể tạo ra nhiều mạng ảo mà tất cả sử dụng thực hiện vật lý như nhau. Điều này có thể hữu ích nếu bạn cần phải xác định một mạng cho các dữ liệu thu thập với một tập hợp các đặc tính hiệu suất, năng lực và một mạng cho các ứng dụng với hiệu suất và công suất khác nhau. Hạn chế trong các lớp mạng có thể dẫn đến tắc nghẽn dẫn đến độ trễ không thể chấp nhận trong môi trường dữ liệu lớn. Ảo hóa mạng giúp giảm những tắc nghẽn và cải thiện khả năng quản lý dữ liệu lớn phân phối cần thiết để phân tích dữ liệu lớn.

e. Bộ vi xử lý và bộ nhớ ảo

Bộ vi xử lý ảo hóa giúp tối ưu hóa bộ xử lý và tối đa hóa hiệu suất. Bộ nhớ ảo tách riêng bộ nhớ từ các máy chủ. Trong phân tích Big data, bạn có thể lặp đi lặp lại các truy vấn của tập dữ liệu lớn và tạo ra các thuật toán phân tích tiên tiến, tất cả được thiết kế để tìm kiếm các mẫu và xu hướng chưa được hiểu rõ. Những phân tích tiên tiến có thể đòi hỏi nhiều sức mạnh xử lý (CPU) và bộ nhớ (RAM). Đối với một số tính toán, nó có thể mất một thời gian dài mà không có đủ CPU và tài nguyên bộ nhớ. Bộ vi xử lý và bộ nhớ ảo có thể giúp tăng tốc độ xử lý và nhận được kết quả phân tích của bạn sớm hơn.

f. Dữ liệu và lưu trữ ảo hóa

Ảo hóa dữ liệu có thể được sử dụng để tạo ra một nền tảng cho các dịch vụ dữ liệu liên kết động. Điều này cho phép dữ liệu được dễ dàng tìm kiếm và kết nối thông qua một nguồn tham khảo thống nhất. Kết quả là dữ liệu ảo hóa cung cấp một dịch vụ trừu tượng mà không phụ thuộc vào cơ sở dữ liệu vật lý bên dưới. Ngoài ra, dữ liệu ảo hóa cho thấy nhiều dữ liệu được lưu trữ cho tất cả các ứng dụng để cải thiện hiệu suất.

Ảo hóa lưu trữ kết hợp tài nguyên lưu trữ vật lý để chia sẻ hiệu quả hơn. Điều này làm giảm chi phí lưu trữ và làm cho nó dễ dàng hơn để quản lý các cửa hàng dữ liệu cần thiết phân tích dữ liệu lớn.

Ảo hóa dữ liệu và ảo hóa lưu trữ đóng một vai trò quan trọng trong làm cho dễ dàng hơn và ít tốn kém hơn để lưu trữ, tải về, và phân tích khối lượng lớn các loại dữ liệu. Hãy nhớ rằng một số dữ liệu lớn có thể không có cấu trúc và không dễ dàng được lưu trữ bằng phương pháp truyền thống. Ảo hóa lưu trữ làm cho nó dễ dàng hơn để lưu trữ các loại dữ liệu lớn và không có cấu trúc. Trong một môi trường dữ liệu lớn, đó là lợi thế để có quyền truy cập vào một loạt các cửa hàng dữ liệu hoạt động theo yêu cầu. Ví dụ, bạn có thể chỉ cần truy cập vào một cơ sở dữ liệu dạng cột thường xuyên. Với ảo hóa, các cơ sở dữ liệu có thể được lưu trữ như là một hình ảnh ảo và gọi bất cứ khi nào nó là cần thiết mà không cần tiêu tốn tài nguyên trung tâm dữ liệu có giá trị hoặc công suất.

1.1.2.2 Quản lý ảo hóa với Hypervisor

Trong một thế giới lý tưởng, bạn không muốn lo lắng về các hệ thống điều hành cơ bản và các phần cứng vật lý. Hypervisor là công nghệ có trách nhiệm đảm bảo rằng chia sẻ tài nguyên diễn ra một cách trật tự và lặp lại, cho phép nhiều hệ điều hành để chia sẻ một máy chủ duy nhất. Nó tạo ra và chạy các máy ảo. Hypervisor nằm ở mức thấp nhất của môi trường phần cứng và sử dụng một lớp mỏng của mã lệnh để cho phép chia sẻ tài nguyên động.

Trong thế giới của Big data, bạn có thể cần phải hỗ trợ nhiều môi trường hoạt động khác nhau. Hypervisor trở thành một cơ chế cung cấp lý tưởng cho các thành phần công nghệ của các dữ liệu ngăn xếp lớn. Hypervisor cho phép bạn hiển thị các ứng dụng tương tự trên rất nhiều hệ thống mà không cần phải thể chất sao chép ứng dụng vào từng hệ thống. Là một lợi ích bổ sung, vì kiến trúc hypervisor, nó có thể tải bất kỳ (hoặc nhiều) hệ điều hành khác nhau như thể họ chỉ là một ứng dụng khác.

1.1.2.3 Trừu tượng hóa và ảo hóa

Đối với các nguồn tài nguyên và các dịch vụ được ảo hóa, chúng được tách ra khỏi môi trường phân phối vật lý cơ bản. Thuật ngữ kỹ thuật đối với hành vi tách biệt này được gọi là trừu tượng hóa. Trừu tượng hóa là một khái niệm quan trọng trong dữ liệu lớn. MapReduce và Hadoop được phân phối môi trường điện toán mà tất cả mọi thứ là trừu tượng. Với các chi tiết là trừu tượng hóa thì các nhà phát triển hoặc nhà phân tích không cần phải quan tâm đến nơi mà các yếu tố dữ liệu đó thực sự nằm vị trí nào.

Trừu tượng hóa giảm thiểu sự phức tạp của một dữ liệu nào đó bằng cách ẩn các chi tiết và chỉ cung cấp các thông tin có liên quan. Ví dụ, nếu bạn đã đi để lấy một người mà bạn chưa bao giờ gặp nhau trước đây, họ có thể cho bạn biết vị trí để gặp người đó, chiều cao, màu tóc, và sẽ mặc những gì. Họ không cần phải cho bạn biết nơi họ được sinh ra, có bao nhiêu tiền trong ngân hàng, ngày tháng năm sinh của mình. Đó là ý tưởng với trừu tượng - đó là cung cấp một đặc điểm kỹ thuật cao cấp hơn là đi sâu vào nhiều chi tiết về một cái gì đó làm việc như thế nào. Trong đám mây, ví dụ, trong một cơ sở hạ tầng như là một mô hình cung cấp dịch vụ (IaaS), các chi tiết của cơ sở hạ tầng vật lý và ảo được trừu tượng hóa từ người dùng.

1.1.2.4 Triển khai ảo hóa để làm việc với Big data

Ảo hóa giúp làm cho môi trường CNTT của bạn đủ thông minh để xử lý phân tích dữ liệu lớn. Bằng cách tối ưu hóa tất cả các yếu tố của cơ sở hạ tầng, bao gồm cả phần cứng, phần mềm và lưu trữ, bạn đạt được hiệu quả cần thiết để xử lý và quản lý khối lượng lớn dữ liệu có cấu trúc và không có cấu trúc. Với dữ liệu lớn, bạn cần truy cập, quản lý và phân tích dữ liệu có cấu trúc và phi cấu trúc trong một môi trường phân phối.

Big data giả định phân phối. Trong thực tế, bất kỳ loại MapReduce nào cũng sẽ làm việc tốt hơn trong một môi trường ảo hóa. Bạn cần có khả năng di chuyển khối lượng công việc xung quanh dựa trên yêu cầu cho sức mạnh tính toán và lưu trữ. Ảo hóa sẽ cho phép bạn để giải quyết những vấn đề lớn mà chưa được giới hạn phạm vi. Ảo hóa sẽ cho phép hỗ trợ một loạt các cửa hàng dữ liệu lớn hoạt động. Ví dụ, một cơ sở dữ liệu đồ thị có thể trở thành như một hình ảnh.

Lợi ích trực tiếp nhất từ ảo hóa là để đảm bảo rằng công cụ MapReduce làm việc tốt hơn. Ảo hóa sẽ dẫn đến phạm vi tốt hơn và hiệu suất cao hơn cho MapReduce. Mỗi một Map và Reduce cần được thực hiện một cách độc lập. Nếu động cơ MapReduce là song song và được cấu hình để chạy trong một môi trường ảo, bạn có thể giảm chi phí quản lý và cho phép mở rộng và co thắt trong các khối lượng công việc nhiệm vụ. MapReduce chính nó là vốn song song và phân tán. Bằng cách bắt các MapReduce trong một container ảo, bạn có thể chạy những gì bạn muốn bất cứ khi nào bạn cần nó. Với ảo hóa, bạn tăng cường sử dụng tài sản mà bạn đã trả tiền bằng cách chuyển chúng các nguồn tài nguyên chung.

1.1.3 Kiểm tra đám mây và Big data

Sức mạnh của đám mây là người dùng có thể truy cập vào tài nguyên máy tính và lưu trữ cần thiết với rất ít hoặc không có hỗ trợ IT hay phải mua thêm phần cứng hoặc phần mềm. Một trong những đặc điểm quan trọng của đám mây là khả năng mở rộng đàn hồi: Người dùng có thể thêm hoặc bớt đi các nguồn lực trong gần như thời gian thực dựa trên yêu cầu thay đổi. Các đám mây đóng một vai trò quan trọng trong thế giới dữ liệu lớn. Những thay đổi lớn xảy ra khi các thành phần cơ sở hạ tầng được kết hợp với những tiến bộ trong quản lý dữ liệu. Mở rộng chiều ngang và tối ưu hóa cơ sở hạ tầng hỗ trợ việc thực hiện thực tế của dữ liệu lớn.

1.1.3.1 Xác định các đám mây trong Big data

Điện toán đám mây là một phương pháp cung cấp một tập hợp các tài nguyên máy tính chia sẻ bao gồm các ứng dụng, tính toán, lưu trữ, mạng, phát triển và nền tảng triển khai, cũng như các quá trình kinh doanh. Điện toán đám mây biến tài sản máy tính bị bưng bít truyền thống vào biển chia sẻ các nguồn tài nguyên dựa trên một nền tảng Internet cơ bản. Trong điện toán đám mây, tất cả mọi thứ, từ sức mạnh tính toán cơ sở hạ tầng điện toán và từ các ứng dụng và các quá trình kinh doanh để dữ liệu và phân tích, có thể được gửi đến bạn như một dịch vụ. Để được hoạt động trong thế giới thực, các đám mây phải được thực hiện với quy trình chuẩn hóa thông thường và tự động hóa.

Nhiều dịch vụ điện toán đám mây doanh nghiệp tận dụng cho tất cả mọi thứ từ sao lưu vào phần mềm như một dịch vụ (SaaS) tùy chọn như quản lý quan hệ khách hàng dịch vụ (CRM). Với sự phát triển của điện toán di động, nhiều người tiêu dùng, các chuyên gia và các công ty đang tạo và truy cập dữ liệu với các dịch vụ dựa trên đám mây. Người tiêu dùng trung bình có thể được gửi một phiếu giảm giá trực tuyến với một cửa hàng yêu thích; một người quản lý kiểm soát chất lượng trong một nhà máy sản xuất có thể thu thập dữ liệu cảm biến từ một loạt các máy móc để xác định liệu một vấn đề chất lượng tồn tại. Các kịch bản này được xác định trên cơ sở hạ tầng dịch vụ dữ liệu dựa trên đám mây.

Một ví dụ phổ biến về lợi ích của điện toán đám mây hỗ trợ dữ liệu lớn có thể được ghi nhận ở cả Google và Amazon.com. Cả hai công ty phụ thuộc vào khả năng quản lý một lượng lớn dữ liệu để di chuyển các doanh nghiệp của họ về phía trước. Các nhà cung cấp cần thiết để đến với cơ sở hạ tầng và các công nghệ có thể hỗ trợ các ứng dụng ở quy mô lớn. Hãy xem xét Gmail và hàng triệu triệu thông điệp rằng Google sẽ xử lý mỗi ngày như là một phần của dịch vụ này. Google đã có thể tối ưu hóa hệ điều hành Linux và môi trường phần mềm của mình để hỗ trợ e-mail theo cách hiệu quả nhất. Do đó, nó có thể dễ dàng hỗ trợ hàng trăm triệu người sử dụng. Quan trọng hơn nữa, Google có thể nắm bắt và tận dụng số lượng lớn các dữ liệu về cả hai người sử dụng mail của mình và sử dụng công cụ tìm kiếm của mình.

Tương tự như vậy, Amazon.com, với các trung tâm dữ liệu IaaS của nó, được tối ưu hóa hỗ trợ cho những khối lượng công việc để Amazon có thể tiếp tục cung cấp các dịch vụ mới và hỗ trợ một số lượng ngày càng tăng của khách hàng mà không vi phạm các ngân hàng. Để phát triển kinh doanh bán lẻ của mình, Amazon phải có khả năng quản lý dữ liệu về hàng hóa của mình, người mua nó, và kênh của các thương gia của đối tác. Nhắm mục tiêu quảng cáo dựa trên mô hình mua của khách hàng là rất quan trọng cho sự thành công của công ty. Các công ty này hiện cung cấp một loạt các dịch vụ dựa trên đám mây cho dữ liệu.

1.1.3.2 Tìm hiểu về triển khai đám mây và mô hình phân phát

Hai mô hình điện toán đám mây chính trong các cuộc thảo luận về dữ liệu lớn là những đám mây công cộng và đám mây riêng. Đối với những tổ chức thông qua việc triển khai điện toán đám mây và cung cấp các mô hình, hầu hết sẽ sử dụng một sự kết hợp của nguồn tin máy tính (trung tâm dữ liệu và những đám mây tư nhân) và các dịch vụ công cộng (điều hành bởi một công ty bên ngoài để sử dụng chia sẻ của một loạt các khách hàng trả một trọng phí sử dụng). Làm thế nào các công ty cân bằng cung cấp công cộng và tư nhân phụ thuộc vào một số vấn đề, trong đó có sự riêng tư, độ trễ, và mục đích. Điều quan trọng là phải hiểu những môi trường và ý nghĩa của chúng đối với việc triển khai dữ liệu lớn tiềm năng. Bằng cách đó, bạn có thể xác định xem bạn có thể muốn sử dụng một IaaS đám mây công cộng (mô tả sau) - ví dụ, đối với các dự án lớn dữ liệu của bạn - hoặc nếu bạn muốn tiếp tục để giữ tất cả các dữ liệu của bạn trên cơ sở. Hoặc, bạn có thể muốn sử dụng một sự kết hợp của cả hai. Vì vậy, họ phác thảo các mô hình triển khai và phân phối đầu tiên và sau đó nói chuyện nhiều hơn về những gì họ có nghĩa là dữ liệu lớn.

a. Mô hình triển khai điện toán đám mây

Các đám mây công cộng

Các đám mây công cộng là một tập hợp các phần cứng, mạng, lưu trữ, dịch vụ, ứng dụng và giao diện thuộc sở hữu và điều hành bởi một bên thứ ba để sử dụng bởi các công ty và cá nhân khác. Các nhà cung cấp thương mại tạo ra một trung tâm dữ liệu cao khả năng mở rộng mà giấu đi các chi tiết của cơ sở hạ tầng cơ bản từ người tiêu dùng. đám mây công cộng là khả thi bởi vì họ thường quản lý khối lượng công việc tương đối lặp đi lặp lại hoặc đơn giản. Ví dụ, thư điện tử là một ứng dụng rất đơn giản. Do đó, một nhà cung cấp điện toán đám mây có thể tối ưu hóa môi trường để nó là thích hợp nhất để hỗ trợ một số lượng lớn các khách hàng, thậm chí nếu nó giúp tiết kiệm rất nhiều tin nhắn.

Tương tự như vậy, các nhà cung cấp điện toán đám mây công cộng cung cấp dịch vụ lưu trữ hoặc máy tính tối ưu hóa phần cứng và phần mềm máy tính của họ để hỗ trợ các loại hình cụ thể của khối lượng công việc.

Ngược lại, các trung tâm dữ liệu điển hình hỗ trợ rất nhiều các ứng dụng khác nhau và khối lượng công việc mà nó không thể dễ dàng tối ưu hóa. Một đám mây công cộng có thể rất hiệu quả khi một tổ chức đang thực hiện một dự án phân tích dữ liệu phức tạp và cần chu kỳ tính toán thêm để xử lý các nhiệm vụ. Ngoài ra, các công ty có thể chọn để lưu trữ dữ liệu trong một đám mây công cộng, nơi chi phí cho mỗi gigabyte là tương đối rẻ tiền so với dung lượng đã mua. Những vấn đề quan trọng hơn với những đám mây công cộng cho dữ liệu lớn là các yêu cầu an ninh và số lượng của độ trễ đó là chấp nhận được.

Tất cả các đám mây công cộng là không giống nhau. Một số đám mây công cộng là dịch vụ quản lý khả năng mở rộng với một mức độ bảo mật cao và một mức độ cao về quản lý dịch vụ. Những đám mây công cộng khác ít mạnh mẽ và kém an toàn, nhưng họ ít chi phí để sử dụng. Sự lựa chọn của bạn sẽ phụ thuộc vào tính chất của dự án dữ liệu lớn của bạn và mức độ rủi ro mà bạn có thể lường trước.

Các đám mây riêng

Một đám mây riêng là một tập hợp các phần cứng, mạng, lưu trữ, dịch vụ, ứng dụng và giao diện thuộc sở hữu và điều hành bởi một tổ chức đối với việc sử dụng các nhân viên của mình, đối tác và khách hàng. Một đám mây riêng có thể được tạo ra và bởi một bên thứ ba quản lý cho việc sử dụng độc quyền của một doanh nghiệp. Các đám mây riêng là một môi trường kiểm soát chặt chẽ không mở cửa cho công chúng. Do đó, các đám mây riêng nằm sau tường lửa. Các đám mây riêng được tự động hóa cao, tập trung vào quản trị, an ninh, và tuân thủ. Tự động thay thế các quy trình thủ công hơn trong việc quản lý dịch vụ CNTT để hỗ trợ khách hàng. Bằng cách này, các quy định và quy trình kinh doanh có thể được thực hiện bên trong phần mềm để các môi trường trở nên dễ dự đoán hơn và dễ quản lý. Nếu các tổ chức đang quản lý một dự án dữ liệu lớn mà yêu cầu xử lý một lượng lớn dữ liệu, các đám mây riêng có thể là sự lựa chọn tốt nhất về độ trễ và an ninh.

Một đám mây lai là một sự kết hợp của một đám mây riêng kết hợp với việc sử dụng các dịch vụ đám mây công cộng với một hoặc một số điểm tiếp xúc giữa các môi trường. Mục đích là để tạo ra một môi trường điện toán đám mây được quản lý tốt, có thể kết hợp các dịch vụ và dữ liệu từ một loạt các mô hình điện toán đám mây để tạo ra một môi trường tính toán thống nhất, tự động và được quản lý tốt.

b. Mô hình cung cấp điện toán đám mây

Ngoài các mô hình triển khai điện toán đám mây đã thảo luận trước đây, một số mô hình cung cấp điện toán đám mây cũng tồn tại. Bốn trong những phổ biến nhất được mô tả trong các phần sau.

- Cơ sở hạ tầng như một dịch vụ (IaaS): là một trong những mô hình đơn giản nhất của các dịch vụ điện toán đám mây. IaaS là việc cung cấp các dịch vụ điện toán bao gồm phần cứng, mạng, lưu trữ, và không gian trung tâm dữ liệu dựa trên một mô hình cho thuê. Người tiêu dùng của dịch vụ mua lại một nguồn tài nguyên và được tính cho rằng nguồn tài nguyên dựa trên số tiền sử dụng và thời gian sử dụng mà. Bạn tìm thấy phiên bản cả công cộng và cá nhân của IaaS. Trong IaaS công cộng, người dùng sử dụng một thẻ tín dụng để có được các nguồn lực này. Khi người dùng ngừng trả tiền, tài nguyên biến mất. Trong một dịch vụ IaaS cá nhân, nó thường là các tổ chức CNTT hoặc tích hợp một người tạo ra cơ sở hạ tầng được thiết kế để cung cấp các nguồn tài nguyên theo yêu cầu cho người sử dụng nội bộ và đôi khi các đối tác kinh doanh.

- Nền tảng như một dịch vụ (PaaS): là một cơ chế cho việc kết hợp IaaS với một bộ trừu tượng của các dịch vụ trung gian, phát triển phần mềm, và các công cụ triển khai cho phép tổ chức để có một cách phù hợp để tạo ra và triển khai các ứng dụng trên một đám mây hoặc trên cơ sở. Một PaaS cung cấp một tập hợp các chương trình hoặc dịch vụ trung gian để đảm bảo rằng các nhà phát triển có một cách cũng được thử nghiệm và tích hợp tốt để tạo ra các ứng dụng trong môi trường điện toán đám mây. Một môi trường PaaS mang lại sự phát triển và triển khai với nhau để tạo ra một cách dễ quản lý hơn để xây dựng, triển khai và quy mô ứng dụng. Một PaaS yêu cầu một IaaS.
- Phần mềm như là một dịch vụ (SaaS): là một ứng dụng kinh doanh tạo ra và lưu trữ bởi một nhà cung cấp trong một mô hình multitenant (cho thuê). Multitenancy đề cập đến tình huống mà một trường hợp duy nhất của một ứng dụng chạy trong một môi trường điện toán đám mây, nhưng phục vụ nhiều tổ chức khách hàng (người thuê), giữ tất cả dữ liệu của họ riêng biệt. Khách hàng trả tiền cho các dịch vụ cho mỗi người dùng hoặc trên một mô hình hợp đồng hàng tháng hoặc hàng năm. Mô hình SaaS ngồi trên đầu trang của cả PaaS và IaaS nền tảng.
- Dữ liệu như một dịch vụ (DaaS): là một mô hình phân phối. DaaS liên quan chặt chẽ đến SaaS. DaaS là một dịch vụ độc lập nền tảng đó sẽ cho phép bạn kết nối với các đám mây để lưu trữ và lấy dữ liệu của bạn. Ngoài ra, bạn tìm thấy một số các dịch vụ dữ liệu chuyên ngành là lợi ích lớn trong một môi trường dữ liệu lớn. Ví dụ, Google cung cấp một dịch vụ mà có thể xử lý một truy vấn với 5 terabyte dữ liệu chỉ trong 15 giây. Đây là loại truy vấn thường sẽ mất mười lần như lâu dài với một trung tâm dữ liệu điển hình. Hàng trăm dịch vụ phân tích chuyên ngành đã được phát triển bởi các công ty như IBM và những người khác.

1.1.3.3 Điện toán đám mây như là bắt buộc cho Big data

Rõ ràng, rất nhiều sự kết hợp của việc triển khai và chuyển giao mô hình tồn tại đối với dữ liệu lớn trong các đám mây.Trong thực tế, một số đặc điểm đám mây làm cho nó trở thành một phần quan trọng của hệ sinh thái dữ liệu lớn:

- **Khả năng mở rộng**: Khả năng mở rộng liên quan đến phần cứng với đề cập đến khả năng đi từ nhỏ đến một lượng lớn sức mạnh xử lý với kiến trúc giống nhau. Đối với các phần mềm, nó đề cập đến sự thống nhất về hiệu suất trên một đơn vị điện là tài nguyên phần cứng tăng. Những đám mây có thể mở rộng lên tới khối lượng dữ liệu lớn. phân phối máy tính, một phần không thể thiếu của các mô hình điện toán đám mây, thực sự làm việc trên một kế hoạch "chia để trị". Vì vậy, nếu bạn có khối lượng lớn dữ liệu, chúng có thể được phân chia trên các máy chủ đám mây. Một đặc tính quan trọng của IaaS là nó có thể tự động mở rộng quy mô. Điều này có nghĩa rằng nếu bạn gió lên cần nhiều nguồn lực hơn mong đợi, bạn có thể nhận được chúng. Điều này gắn vào các khái niệm về khả năng mở rộng.
- **Độ co dãn**: Độ đàn hồi đề cập đến khả năng mở rộng hoặc thu nhỏ tính toán nhu cầu tài nguyên trong thời gian thực, dựa trên nhu cầu. Một trong những lợi ích của điện toán đám mây là khách hàng có khả năng truy cập càng nhiều các dịch vụ khi họ cần khi họ cần nó. Điều này có thể hữu ích cho các dự án dữ liệu lớn, nơi bạn có thể cần phải mở rộng số lượng tài nguyên máy tính bạn cần phải đối phó với khối lượng và vận tốc của dữ liệu. Tất nhiên, tính năng này của các đám mây làm cho nó rất hấp dẫn cho người dùng và các nhà cung cấp dịch vụ cần phải thiết kế một kiến trúc nền tảng được tối ưu hóa cho các loại hình dịch vụ.
- **Tổng hợp tài nguyên**: Kiến trúc điện toán đám mây cho phép việc tạo ra hiệu quả của các nhóm tài nguyên chia sẻ rằng làm cho các đám mây hiệu quả kinh tế.
- **Tự phục vụ**: Với tự phục vụ, người dùng của một tài nguyên điện toán đám mây có thể sử dụng một trình duyệt hoặc một giao diện cổng thông tin để có được các nguồn lực cần thiết. Ví dụ, để chạy một mô hình dự đoán rất lớn. Đây là sự khác lạ so với cách bạn có thể đạt được các nguồn lực từ một trung tâm dữ liệu, các bạn sẽ phải yêu cầu các nguồn lực từ các hoạt động CNTT.

- **Chi phí thường thấp**: Nếu bạn sử dụng một nhà cung cấp điện toán đám mây, chi phí thường có thể được giảm bớt bởi vì bạn không mua một lượng lớn phần cứng, cho thuê không gian mới để đối phó với dữ liệu lớn của bạn. Bằng cách lợi dụng các nền kinh tế của quy mô kết hợp với các môi trường điện toán đám mây, điện toán đám mây có thể trông hấp dẫn. Tất nhiên, bạn sẽ cần phải làm tính toán riêng của mình để đánh giá xem bạn quan tâm đến một đám mây công cộng, đám mây riêng, đám mây lai, hoặc không có mây.
- **Trả tiền khi bạn đi**: Một lựa chọn thanh toán điển hình cho một nhà cung cấp điện toán đám mây là phải trả tiền như You Go (PAYG), có nghĩa là bạn sẽ được thanh toán cho các nguồn lực được sử dụng dựa trên giá. Điều này có thể hữu ích nếu bạn không chắc chắn những gì các nguồn lực cần thiết cho dự án dữ liệu lớn của bạn.
- **Khả năng chịu lỗi**: Cung cấp dịch vụ đám mây nên có khả năng chịu lỗi được xây dựng trong kiến trúc của họ, cung cấp dịch vụ không bị gián đoạn bất chấp sự thất bại của một hoặc nhiều thành phần của hệ thống.

Trong một số tình huống, một nhà cung cấp dịch vụ không thể dự đoán được nhu cầu của khách hàng. Do đó, nó được phổ biến cho một nhà cung cấp dịch vụ để thêm năng lực bổ sung từ một nhà cung cấp dịch vụ của bên thứ ba. Thông thường, người tiêu dùng không hề biết rằng mình đang đối phó với một nhà cung cấp dịch vụ đám mây khác.

1.1.3.4 Sử dụng điện toán đám mây cho Big data

Rõ ràng, chính bản chất của các đám mây làm nên một môi trường máy tính lý tưởng cho các dữ liệu lớn. Vậy làm thế nào bạn có thể sử dụng dữ liệu lớn cùng với những đám mây? Dưới đây là một số ví dụ:

- IaaS trong một đám mây công cộng: Trong hoàn cảnh này, bạn sẽ được sử dụng cơ sở hạ tầng một nhà cung cấp điện toán đám mây công cộng cho các dịch vụ dữ liệu lớn của bạn, bởi vì bạn không muốn sử dụng cơ sở hạ tầng vật lý của riêng bạn. IaaS có thể cung cấp cho việc tạo ra các máy ảo với dung lượng gần như vô hạn và sức mạnh tính toán. Bạn có thể chọn hệ điều hành mà bạn muốn, và bạn có sự linh hoạt để tự động mở rộng môi trường để đáp ứng nhu cầu của bạn. Bạn có thể xử lý hàng tỷ dòng dữ liệu để nhắm mục tiêu với các quảng cáo của khách hàng ngay trong thời gian thực.

- PaaS trong một đám mây riêng: PaaS là toàn bộ cơ sở hạ tầng đóng gói để có thể được sử dụng để thiết kế, thực hiện và triển khai các ứng dụng và dịch vụ trong một môi trường đám mây công cộng hay riêng. PaaS cho phép một tổ chức tận dụng các dịch vụ trung gian quan trọng mà không cần phải đối phó với sự phức tạp của việc quản lý phần cứng và phần mềm. Nhà cung cấp PaaS đang bắt đầu để kết hợp các công nghệ dữ liệu lớn như Hadoop MapReduce và thành PaaS dịch vụ của họ. Ví dụ, bạn có thể muốn xây dựng một ứng dụng chuyên ngành để phân tích một lượng lớn các dữ liệu y tế. Các ứng dụng sẽ sử dụng thời gian thực cũng như dữ liệu phi thời gian thực. Nó sẽ đòi hỏi Hadoop MapReduce lưu trữ và xử lý. Có gì tuyệt vời về PaaS trong kịch bản này là cách nhanh chóng các ứng dụng có thể được triển khai. Bạn sẽ không phải chờ đợi cho các đội IT nội bộ để có được tốc độ trên các công nghệ mới và bạn có thể thử nghiệm tự do hơn. Một khi bạn đã xác định được một giải pháp vững chắc, bạn có thể sử dụng nó khi mà CNTT luôn sẵn sàng để hỗ trợ nó.
- SaaS trong một đám mây lai: Tại đây bạn có thể muốn phân tích "tiếng nói của khách hàng" dữ liệu từ nhiều kênh. Nhiều công ty đã nhận ra rằng một trong những nguồn dữ liệu quan trọng nhất là những gì khách hàng nghĩ và nói về công ty của họ, sản phẩm của họ, và các dịch vụ của họ. Tiếp cận được tiếng nói của các dữ liệu khách hàng có thể cung cấp những hiểu biết vô giá vào hành vi và hành động. Ngày càng có nhiều khách hàng đang đưa ra ý kiến của mình trên các trang web công cộng trên Internet. Các giá trị đầu vào của khách hàng có thể được tăng cường rất nhiều bằng cách kết hợp dữ liệu công cộng này vào phân tích của bạn. Nhà cung cấp SaaS của bạn cung cấp nền tảng cho việc phân tích cũng như các dữ liệu truyền thông xã hội. Ngoài ra, bạn có thể sử dụng dữ liệu CRM doanh nghiệp của bạn trong môi trường đám mây riêng của mình để đưa vào phân tích.

Một số người trong ngành công nghiệp đang sử dụng các ứng dụng dữ liệu lớn khi mô tả các ứng dụng chạy trong đám mây sử dụng Big data. Ví dụ này bao gồm Amazon.com và LinkedIn. Bây giờ một số người có thể tranh luận rằng đây thực sự là những ứng dụng SaaS giải quyết những vấn đề kinh doanh cụ thể. Nó thường là một vấn đề của ngữ nghĩa trong một không gian mới nổi.

1.1.3.5 Nhà cung cấp điện toán đám mây trên Big data

Điện toán đám mây có đủ hình dạng và kích cỡ và cung cấp nhiều sản phẩm khác nhau. Một số các nhà cung cấp điện toán đám mây cung cấp dịch vụ IaaS có thể được sử dụng cho dữ liệu lớn bao gồm Amazon.com, AT & T, GoGrid, Joyent, Rackspace, IBM, và Verizon / Terremark. Tuy nhiên, các công ty điện toán đám mây và các nhà cung cấp dịch vụ điện toán đám mây cũng được cung cấp phần mềm có mục tiêu đặc biệt cho dữ liệu lớn.

a. Điện toán đám mây công cộng của Amazon

Hiện nay, một trong những nhà cung cấp dịch vụ IaaS cao nhất là Amazon Web Services với Elastic Compute Cloud (Amazon EC2). Amazon đã không bắt đầu với một tầm nhìn để xây dựng một doanh nghiệp dịch vụ cơ sở hạ tầng lớn. Thay vào đó, công ty đã xây dựng một cơ sở hạ tầng lớn để hỗ trợ kinh doanh bán lẻ của riêng mình và phát hiện ra rằng các nguồn lực của nó đã không được tận dụng. Thay vì cho phép tài sản này để ngồi nhàn rỗi, họ quyết định để tận dụng nguồn tài nguyên này trong khi thêm vào dòng dưới cùng. Dịch vụ EC2 của Amazon đã được đưa ra vào năm 2006 và tiếp tục phát triển.

Amazon EC2 cung cấp khả năng mở rộng dưới sự kiểm soát của người dùng, với người sử dụng phải trả cho nguồn tài nguyên theo giờ. Việc sử dụng các giới hạn linh hoạt trong việc đặt tên của EC2 của Amazon là đáng kể. Ở đây, độ linh hoạt đề cập đến khả năng mà người sử dụng EC2 phải tăng hoặc giảm các nguồn lực cơ sở hạ tầng giao để đáp ứng nhu cầu của họ.

Amazon cũng cung cấp các dịch vụ dữ liệu lớn khác cho khách hàng với danh mục đầu tư của Amazon Web Services của nó. Chúng bao gồm những điều sau đây:

- Amazon Elastic MapReduce: Mục tiêu cho xử lý khối lượng lớn dữ liệu. Elastic MapReduce sử dụng một khuôn khổ lưu trữ Hadoop đang chạy trên EC2 và Amazon Simple Storage Service (Amazon S3). Người dùng có thể chạy HBase (một phân phối, lưu trữ dữ liệu cột định hướng).
- Amazon DynamoDB: Một dịch vụ cơ sở dữ liệu quản lý hoàn toàn không chỉ SQL (NoSQL). DynamoDB là khả năng chịu lỗi, tính sẵn sàng cao dịch vụ lưu trữ dữ liệu cung cấp tự cung, khả năng mở rộng trong suốt, đơn giản và quản. Nó được thực hiện trên các ổ SSD (ổ đĩa trạng thái rắn) cho độ tin cậy và hiệu suất cao.

- Amazon Simple Storage Service (S3): Một dịch vụ web quy mô được thiết kế để lưu trữ bất kỳ số lượng dữ liệu. Sức mạnh của trung tâm thiết kế của nó là hiệu suất và khả năng mở rộng, vì vậy nó không phải là tính năng đầy như các cửa hàng dữ liệu khác. Dữ liệu được lưu trữ trong "thùng" và bạn có thể chọn một hoặc nhiều khu vực trên toàn cầu cho lưu trữ vật lý để giải quyết nhu cầu độ trễ.
- Amazon High Performance Computing: Điều chỉnh cho các nhiệm vụ chuyên môn, dịch vụ này cung cấp độ trễ thấp, điều chỉnh các cụm tính toán hiệu năng cao. Thường xuyên được sử dụng nhất bởi các nhà khoa học và các viện nghiên cứu, HPC đang bước vào dòng chính. Các cụm Amazon HPC mục đích là xây dựng cho khối lượng công việc cụ thể và có thể được cấu hình lại dễ dàng cho nhiệm vụ mới.
- Amazon RedShift: Có sẵn trong bản xem trước hạn, RedShift là một dịch vụ lưu trữ dữ liệu quy mô petabyte được xây dựng trên một kiến trúc MPP khả năng mở rộng. Được quản lý bởi Amazon, nó cung cấp một thay thế đáng tin cậy an toàn cho kho dữ liệu trong nhà và tương thích với một số công cụ kinh doanh thông minh phổ biến.

b. Dịch vụ dữ liệu lớn Google

Google, người khổng lồ tìm kiếm Internet, cũng cung cấp một số dịch vụ đám mây nhắm mục tiêu cho dữ liệu lớn. Chúng bao gồm những điều sau đây:

- Google Compute Engine: Một khả năng dựa trên đám mây điện toán máy ảo, Google Compute Engine cung cấp một môi trường máy tính an toàn, linh hoạt từ các trung tâm dữ liệu hiệu quả năng lượng. Google cũng cung cấp các giải pháp quản lý khối lượng công việc từ một số đối tác công nghệ đã được tối ưu hóa sản phẩm của mình cho Google Compute Engine.
- Google Big Query: Cho phép bạn chạy các SQL giống như truy vấn ở một tốc độ cao với hàng tỷ bộ dữ liệu lớn. Mặc dù nó là tốt để truy vấn dữ liệu, dữ liệu không thể được sửa đổi sau khi nó đang ở trong đó. Hãy xem xét Google Big Query một loại hệ thống trực tuyến Analytical Processing (OLAP) cho dữ liệu lớn. Nó rât tốt để báo cáo đột xuất hoặc phân tích thăm dò.

- Google Prediction API: Hỗ trợ đám mây, công cụ dự báo có khả năng xác định các mẫu trong dữ liệu và sau đó ghi nhớ chúng. Nó có thể tìm hiểu thêm về một mô hình mỗi khi nó được sử dụng. Các mô hình có thể được phân tích cho nhiều mục đích, bao gồm phát hiện gian lận, phân tích thùng đựng, và ý kiến của khách hàng.

c. Microsoft Azure

Dựa trên khái niệm trừu tượng Windows và SQL, Microsoft đã sản xuất một bộ công cụ phát triển, hỗ trợ máy ảo, quản lý và dịch vụ truyền thông, và các dịch vụ điện thoại di động trong một cung cấp PaaS. Đối với khách hàng có chuyên môn sâu trong Net, SQLServer, và Windows, việc áp dụng các PaaS dựa trên Azure rất đơn giản.

Để giải quyết những yêu cầu mới để tích hợp dữ liệu lớn vào các giải pháp Windows Azure, Microsoft cũng đã bổ sung thêm Windows Azure HDInsight. Được xây dựng trên nền tảng dữ liệu Hortonworks (HDP), mà theo Microsoft, cung cấp khả năng tương thích 100% với Apache Hadoop, HDInsight hỗ trợ kết nối với Microsoft Excel và các công cụ khác kinh doanh thông minh (BI). Ngoài Azure HDInsight cũng có thể được triển khai trên Windows Server.

d. OpenStack

Được khởi xướng bởi Rackspace và NASA, OpenStack đang thực hiện một nền tảng đám mây mở nhắm đến hai đám mây công cộng hay riêng. Trong khi tổ chức được quản lý chặt chẽ bởi Rackspace, nó chuyển đến một nền tảng OpenStack riêng biệt. Mặc dù các công ty có thể tận dụng OpenStack tạo triển khai độc quyền, việc chỉ định OpenStack đòi hỏi sự phù hợp với việc thực hiện tiêu chuẩn của dịch vụ.

Mục tiêu OpenStack là cung cấp một cách ồ, cho thuê đám mây có thể chạy trên bất kỳ phần cứng. OpenStack đang xây dựng một hệ sinh thái rộng lớn của các đối tác quan tâm trong việc áp dụng nền tảng đám mây của mình, bao gồm Dell, HP, Intel, Cisco, Red Hat, và IBM, cùng với ít nhất 100 người khác đang sử dụng OpenStack là nền tảng cho các dịch vụ đám mây của mình. Về bản chất, OpenStack là một nguồn mở IaaS sáng kiến xây dựng trên Ubuntu, một hệ điều hành dựa trên các phân phối Linux Debian. Nó cũng có thể chạy trên các phiên bản của Linux Red Hat.

OpenStack cung cấp một loạt các dịch vụ, bao gồm cả tính toán, lưu trữ đối tượng, danh mục và kho lưu trữ, đo tốc độ, danh tính, và kết nối mạng. Trong điều kiện của dữ liệu lớn, Rackspace và Hortonworks (một nhà cung cấp một nền tảng quản lý dữ liệu mã nguồn mở dựa trên Apache Hadoop) thông báo rằng Rackspace sẽ phát hành dựa trên đám mây công cộng dịch vụ Hadoop OpenStack, mà sẽ được xác nhận và hỗ trợ bởi Hortonworks và sẽ cho phép khách hàng nhanh chóng tạo ra một môi trường dữ liệu lớn.

e. Trường hợp phải cẩn thận khi sử dụng các dịch vụ điện toán đám mây

Dịch vụ dựa trên đám mây có thể cung cấp một giải pháp kinh tế cho nhu cầu dữ liệu lớn của bạn, nhưng những đám mây có vấn đề của nó. Dưới đây là một số vấn đề cần xem xét:

- Tính toàn vẹn dữ liệu: Bạn cần phải chắc chắn rằng nhà cung cấp của bạn có các điều khiển ngay tại chỗ để đảm bảo, duy trì tính toàn vẹn của dữ liệu của bạn.
- Tuân thủ: Hãy chắc chắn rằng nhà cung cấp của bạn có thể thực hiện với bất kỳ vấn đề tuân thủ đặc biệt cho công ty hay ngành công nghiệp của bạn.
- Chi phí: Chi phí nhỏ có thể tăng lên. Hãy cẩn thận để đọc phần nhỏ của bất kỳ hợp đồng nào, và chắc chắn rằng bạn biết những gì bạn muốn làm trong các đám mây.
- Vận chuyển dữ liệu: Hãy chắc chắn để tìm hiểu làm thế nào bạn nhận được dữ liệu của bạn vào các đám mây ở nơi đầu tiên. Ví dụ, một số nhà cung cấp sẽ cho phép bạn gửi cho họ trên phương tiện truyền thông. Những người khác nhấn mạnh vào tải nó qua mạng. Điều này có thể tốn kém, vì vậy hãy cẩn thận.
- Hiệu suất: Bởi vì bạn đang quan tâm đến hiệu suất từ nhà cung cấp dịch vụ của bạn, hãy chắc chắn rằng các định nghĩa rõ ràng các thỏa thuận cấp độ dịch vụ tồn tại sẵn có, hỗ trợ, và hiệu suất. Ví dụ, nhà cung cấp của bạn có thể nói với bạn rằng bạn sẽ có thể truy cập dữ liệu của bạn 99,999% thời gian. Tuy nhiên, hãy chú ý đọc các hợp đồng: thời gian hoạt động này có bao gồm việc bảo trì theo lịch trình hay không?

- Truy cập dữ liệu: Điều gì điều khiển được thực hiện để đảm bảo rằng bạn và chỉ có bạn có thể truy cập dữ liệu của bạn? Nói cách khác, những gì hình thức kiểm soát truy cập an toàn được đưa ra? Điều này có thể bao gồm quản lý danh tính, nơi mà mục tiêu chính là bảo vệ thông tin nhận dạng cá nhân để truy cập vào tài nguyên máy tính, các ứng dụng, dữ liệu và các dịch vụ được kiểm soát đúng.
- Vị trí: Dữ liệu của bạn sẽ được đặt ở đâu? Trong một số công ty và các quốc gia, các vấn đề pháp ngăn chặn dữ liệu được lưu trữ hoặc xử lý trên máy trong một quốc gia khác nhau.

1.2 Quản lý dữ liệu lớn

1.2.1 Cơ sở dữ liệu hoạt động

Dữ liệu lớn đang trở thành một yếu tố quan trọng trong cách tổ chức tận dụng dữ liệu có dung lượng lớn với tốc độ cao để giải quyết vấn đề dữ liệu cụ thể. Tuy nhiên, dữ liệu lớn không tồn tại độc lập. Để có hiệu quả, các công ty thường cần kết hợp các kết quả phân tích dữ liệu lớn với các dữ liệu hiện có trong kinh doanh. Nói cách khác, bạn không thể nghĩ về dữ liệu lớn trong sự độc lập từ các nguồn dữ liệu hoạt động. Có một loạt các dịch vụ dữ liệu hoạt động quan trọng.

Một trong những dịch vụ quan trọng nhất được cung cấp bởi cơ sở dữ liệu hoạt động (các cửa hàng cũng được gọi là dữ liệu) là kiên trì. Sự kiên trì đảm bảo rằng các dữ liệu được lưu trữ trong cơ sở dữ liệu sẽ không được thay đổi mà không cần sự cho phép và nó sẽ có sẵn miễn là nó quan trọng đối với các doanh nghiệp. Những gì tốt là một cơ sở dữ liệu, nếu nó không thể được tin cậy để bảo vệ dữ liệu mà bạn đặt vào nó? Với yêu cầu quan trọng này, bạn phải suy nghĩ về những loại dữ liệu bạn muốn lưu giữ, làm thế nào bạn có thể truy cập và cập nhật nó, và làm thế nào bạn có thể sử dụng nó để đưa ra quyết định nghiệp vụ. Ở cấp độ cơ bản này, sự lựa chọn của các công cụ cơ sở dữ liệu là rất quan trọng để thành công trong việc thực hiện dữ liệu lớn của bạn.

Cơ sở dữ liệu quan hệ được xây dựng trên một hoặc nhiều mối quan hệ và được đại diện bởi các bảng. Các bảng này được định nghĩa bởi các cột, và các dữ liệu được lưu trữ trong các hàng. Các khóa chính thường là cột đầu tiên trong bảng. Sự nhất quán của cơ sở dữ liệu và phần lớn giá trị của nó được thực hiện bằng cách "bình thường hóa" các dữ liệu. Như tên của nó, dữ liệu được chuẩn hóa đã được chuyển đổi từ định dạng gốc vào một chia sẻ, được thoả thuận định dạng. Ví dụ trong một cơ sở dữ liệu bạn có thể có "điện thoại" như XXX-XXX-XXXX trong khi ở khác nó có thể là XXXXXXXXX. Để đạt được một cái nhìn

nhất quán của thông tin, lĩnh vực này sẽ cần phải được bình thường đến một hình thức này hay cách khác. Năm mức độ tiêu chuẩn tồn tại bình thường. Các bộ sưu tập của các bảng, chìa khóa, các yếu tố, và như vậy được gọi là giản đồ cơ sở dữ liệu.

Qua nhiều năm, các ngôn ngữ truy vấn có cấu trúc (SQL) đã tiến hóa với công nghệ RDBMS và là cơ chế sử dụng rộng rãi nhất cho việc tạo ra, truy vấn, bảo trì và vận hành cơ sở dữ liệu quan hệ. Những nhiệm vụ này được gọi là CRUD: Tạo, truy xuất, cập nhật và xóa là phổ biến, hoạt động liên quan bạn có thể sử dụng trực tiếp trên một cơ sở dữ liệu hoặc thông qua một giao diện lập trình ứng dụng (API).

1.2.1.1 RDBMS rất quan trọng trong môi trường Big data

Trong các công ty lớn và nhỏ, hầu hết các thông tin hoạt động quan trọng của họ có thể lưu trữ trong RDBMS. Nhiều công ty có RDBMS khác nhau cho các khu vực khác nhau của kinh doanh. Dữ liệu giao dịch có thể được lưu trữ trong cơ sở dữ liệu một nhà cung cấp, trong khi thông tin khách hàng có thể được lưu trữ trong một. Hiểu biết những gì dữ liệu được lưu trữ và nơi nó được lưu trữ được khối xây dựng quan trọng trong việc thực hiện dữ liệu lớn của bạn. Nó không phải là khả năng bạn sẽ sử dụng RDBMS cho phần lõi của việc thực hiện, nhưng bạn sẽ cần phải dựa trên các dữ liệu được lưu trữ trong RDBMS để tạo ra mức cao nhất của giá trị cho doanh nghiệp với dữ liệu lớn. Mặc dù nhiều cơ sở dữ liệu quan hệ thương mại khác nhau có sẵn từ các công ty như Oracle, IBM và Microsoft, bạn cần phải hiểu một cơ sở dữ liệu quan hệ mã nguồn mở được gọi là PostgreSQL.

PostgreSQL cơ sở dữ liệu quan hệ

PostgreSQL (www.postgresql.org) được sử dụng rộng rãi nhất mã nguồn mở cơ sở dữ liệu quan hệ. Ban đầu nó được phát triển tại Đại học California ở Berkeley và đã được phát triển hoạt động như là một dự án mã nguồn mở trong hơn 15 năm. Một số yếu tố góp phần vào sự phổ biến của PostgreSQL. Là một RDBMS với sự hỗ trợ cho các tiêu chuẩn SQL, nó làm tất cả những điều mong đợi ở một sản phẩm cơ sở dữ liệu, cộng với tuổi thọ của nó và sử dụng rộng rãi đã làm cho nó trở thành "trận chiến thử nghiệm". PostgreSQL cũng hỗ trợ nhiều tính năng chỉ tìm thấy trong RDBMS độc quyền đắt tiền, bao gồm những điều sau đây:

- Khả năng xử lý trực tiếp "đối tượng" trong giản đồ quan hệ.
- Khóa ngoại (tham khảo các khóa từ một bảng trong một bảng khác).

- Khởi sự quá trình (sự kiện sử dụng để tự động bắt đầu một thủ tục lưu trữ).
- Truy vấn phức tạp (các truy vấn con và tham gia nhiều bảng rời rạc).
- Toàn vẹn giao dịch.
- Kiểm soát đồng thời đa phiên bản.

Sức mạnh thực sự của PostgreSQL là khả năng mở rộng của nó. Người sử dụng và lập trình cơ sở dữ liệu có thể thêm các khả năng mới mà không ảnh hưởng đến các hoạt động cơ bản hoặc độ tin cậy của các cơ sở dữ liệu. Phần mở rộng có thể bao gồm:

- Loại dữ liệu
- Các nhà khai thác
- Chức năng
- Phương pháp lập chỉ mục
- Ngôn ngữ thủ tục

Mức độ tùy chỉnh làm cho PostgreSQL mong muốn khi không linh hoạt, sản phẩm độc quyền sẽ không đạt được. Nó mở rộng vô hạn. Cuối cùng, các giấy phép PostgreSQL cho phép sửa đổi, phân phối dưới mọi hình thức, mã nguồn mở hoặc đóng. Bất kỳ thay đổi có thể được giữ riêng hoặc chia sẻ với cộng đồng như bạn muốn. Mặc dù cơ sở dữ liệu quan hệ (bao gồm cả PostgreSQL) đóng một vai trò quan trọng trong dữ liệu lớn "doanh nghiệp", bạn cũng có một số cách tiếp cận khác.

1.2.1.2 Cơ sở dữ liệu không quan hệ

Cơ sở dữ liệu không quan hệ không dựa trên các bảng khóa đặc hữu của RDBMS. Một số công nghệ cơ sở dữ liệu không quan hệ đều có riêng về khả năng tập trung vào các vấn đề cụ thể ngoài phạm vi của RDBMS truyền thống. Tóm lại, dữ liệu đặc biệt trong thế giới dữ liệu lớn đòi hỏi sự kiên trì và các kỹ thuật thao tác dữ liệu. Mặc dù những phong cách mới của cơ sở dữ liệu cung cấp một số câu trả lời cho những thách thức lớn dữ liệu của bạn nhưng nó không phải là một vé nhanh để về đích.

Một trường mới nổi, lớp phổ biến của cơ sở dữ liệu không quan hệ được gọi là không chỉ SQL (NoSQL). Ban đầu khởi tạo hình dung cơ sở dữ liệu mà không đòi hỏi các mô hình quan hệ và SQL. Các lớp khác là cơ sở dữ liệu mà không hỗ trợ mô hình quan hệ, nhưng dựa vào SQL như một phương tiện chính để thao tác dữ liệu bên trong. Mặc dù cơ sở dữ liệu quan hệ quan hệ và không có yếu tố cơ bản tương tự, làm thế nào các nguyên tắc cơ bản được thực hiện tạo sự khác biệt? Công nghệ cơ sở dữ liệu không quan hệ có các đặc điểm sau:

- Khả năng mở rộng: Trong trường hợp này, chúng ta đang đề cập đến khả năng ghi dữ liệu trên nhiều cửa hàng dữ liệu cùng một lúc mà không liên quan đến giới hạn vật lý của các cơ sở hạ tầng cơ bản. Một khía cạnh quan trọng là liền mạch. Các cơ sở dữ liệu phải có khả năng mở rộng và hợp đồng để đáp ứng với các luồng dữ liệu.
- Dữ liệu và mô hình truy vấn: Thay vì các hàng, cột, kết cấu chính, dữ liệu không quan hệ sử dụng các khuôn khổ đặc biệt để lưu trữ dữ liệu với một bộ điều kiện tiên quyết của các API truy vấn đặc biệt để truy cập dữ liệu thông minh.
- Thiết kế kiên trì: kiên trì vẫn là một yếu tố quan trọng trong cơ sở dữ liệu không quan hệ. Do vận tốc cao, chủng loại và khối lượng dữ liệu lớn, các cơ sở dữ liệu sử dụng các cơ chế khác nhau cho sự bền bỉ dữ liệu. Các tùy chọn hiệu suất cao nhất là "trong bộ nhớ", nơi mà toàn bộ cơ sở dữ liệu được lưu giữ trong bộ nhớ hệ thống rất nhanh chóng của máy chủ của bạn.
- Giao diện đa dạng: Mặc dù hầu hết các công nghệ hỗ trợ API RESTful đi đến giao diện, họ cũng cung cấp một loạt các cơ chế kết nối cho các lập trình viên và các nhà quản lý cơ sở dữ liệu, bao gồm các công cụ phân tích và báo cáo.
- Thống nhất cuối cùng: Trong khi sử dụng RDBMS ACID như một cơ chế để đảm bảo tính thống nhất của dữ liệu. Nó có trách nhiệm giải quyết xung đột khi dữ liệu đang chuyển động giữa các nút trong việc thực hiện phân phối. Các trạng thái dữ liệu được duy trì bởi các phần mềm và các mô hình truy cập dựa trên khả cơ bản.

1.2.1.3 Cơ sở dữ liệu cặp Khóa – Giá trị

Đến nay, các cơ sở dữ liệu NoSQL sử dụng các mô hình cặp Key - Value (KVP). Cơ sở dữ liệu KVP không đòi hỏi một sơ đồ (như RDBMS) và cung cấp sự linh hoạt tuyệt vời và khả năng mở rộng. Cơ sở dữ liệu KVP không cung cấp khả năng ACID (hóa trị, nhất quán, cách ly, độ bền), và đòi hỏi người thực hiện phải suy nghĩ về vị trí dữ liệu, sao chép, và khả năng chịu lỗi khi họ không kiểm soát được công nghệ. Cơ sở dữ liệu KVP không có kiểu. Kết quả là, hầu hết các dữ liệu được lưu trữ như chuỗi.

Khi số lượng người dùng tăng lên, việc lưu giữ dấu của các khóa chính xác và giá trị liên quan có thể được thử thách. Nếu bạn cần phải theo dõi các ý kiến của hàng triệu người dùng, số lượng các cặp khóa-giá trị liên kết với chúng có thể tăng theo cấp số nhân. Nếu bạn không muốn để hạn chế sự lựa chọn cho các giá trị, chuỗi đại diện chung của KVP cung cấp sự linh hoạt và khả năng đọc.

Bạn có thể cần bổ sung một số dữ liệu tổ chức trong một cơ sở dữ liệu khóa – giá trị. Hầu hết các cung cấp khóa tổng hợp (và các giá trị liên quan của chúng) vào một bộ sưu tập. Bộ sưu tập có thể bao gồm bất kỳ số lượng các cặp khóa - giá trị và không yêu cầu kiểm soát độc quyền của các yếu tố KVP cá nhân.

1.2.1.4 Cơ sở dữ liệu tài liệu

Bạn tìm thấy hai loại cơ sở dữ liệu tài liệu. Một thường được mô tả như một kho lưu trữ cho toàn bộ nội dung tài liệu kiểu (file Word, trang web hoàn chỉnh,…). Loại kia là một cơ sở dữ liệu để lưu trữ các thành phần tài liệu cho lưu trữ vĩnh viễn như một thực thể tĩnh hoặc để lắp ráp động các bộ phận của một tài liệu. Cấu trúc của các tài liệu và các bộ phận được cung cấp bởi JavaScript Object Notation (JSON) và / hoặc JSON Binary (BSON). Cơ sở dữ liệu tài liệu rất hữu ích khi bạn có để sản xuất rất nhiều báo cáo và họ cần phải được lắp ráp tự động từ các yếu tố làm thay đổi thường xuyên. Một ví dụ là hoàn thành tài liệu y tế, nơi mà phần nội dung sẽ khác nhau dựa trên hồ sơ thành viên (tuổi, cư trú, mức thu nhập), kế hoạch chăm sóc sức khỏe, và hưởng chương trình của chính phủ. Đối với việc triển khai dữ liệu lớn, cả hai phong cách này rất quan trọng, vì vậy bạn nên hiểu các chi tiết của mỗi loại.

Tại cốt lõi của nó, JSON là một định dạng dữ liệu trao đổi, dựa trên một tập hợp con của các ngôn ngữ lập trình JavaScript. Mặc dù là một phần của một ngôn ngữ lập trình, nó là văn bản trong tự nhiên và rất dễ đọc và viết. Nó cũng có lợi thế là dễ dàng cho các máy tính để xử lý. Hai cấu trúc cơ bản tồn tại trong JSON, và họ được hỗ trợ bởi nhiều người, nếu không phải tất cả, các ngôn ngữ lập trình hiện đại. Cấu trúc cơ bản đầu tiên là một bộ sưu tập của các cặp tên / giá trị, và chúng được đại diện lập trình như các đối tượng, hồ sơ, danh sách được khóa,... Cấu trúc cơ bản thứ hai là một danh sách có thứ tự các giá trị, và chúng được đại diện lập trình như mảng, danh sách, hoặc các trình tự.

Cơ sở dữ liệu tài liệu đang trở thành một tiêu chuẩn vàng để áp dụng dữ liệu lớn, vì vậy chúng ta xem xét hai trong những triển khai phổ biến nhất.

MongoDB

MongoDB (www.mongodb.com) là tên dự án cho hệ thống. Nó được duy trì bởi một công ty gọi là 10gen là mã nguồn mở và được tự do hoạt động theo giấy phép GNU v3.0 AGPL. Hoạt động thương mại với đầy đủ hỗ trợ có sẵn từ 10gen.

MongoDB đang ngày càng phổ biến và có thể là một lựa chọn tốt cho các cửa hàng dữ liệu hỗ trợ thực hiện dữ liệu lớn của bạn. MongoDB gồm cơ sở dữ liệu chứa "bộ sưu tập". Một bộ sưu tập gồm "tài liệu", và mỗi tài liệu bao gồm các lĩnh vực. Cũng như trong cơ sở dữ liệu quan hệ, bạn có thể chỉ là một bộ sưu tập. Làm như vậy làm tăng hiệu suất của tra cứu dữ liệu. Không giống như các cơ sở dữ liệu khác, tuy nhiên, MongoDB trả về một cái gì đó gọi là "con trỏ", phục vụ như là một con trỏ đến dữ liệu. Đây là một khả năng rất hữu ích vì nó cung cấp các tùy chọn đếm, phân loại dữ liệu mà không cần trích xuất. Nguyên bản, MongoDB hỗ trợ BSON, việc thực hiện các văn bản nhị phân của JSON.

MongoDB cũng là một hệ sinh thái bao gồm các yếu tố sau:

- Tính sẵn sàng cao và dịch vụ sao chép để nhân rộng trên toàn mạng lưới địa phương và khu vực rộng.
- Một hệ thống lưới điện dựa trên tập tin (GridFS), cho phép lưu trữ các đối tượng lớn bằng cách chia chúng trong nhiều tài liệu.
- MapReduce để hỗ trợ phân tích và tổng hợp các bộ sưu tập / tài liệu khác nhau.

- Dịch vụ phân phối một cơ sở dữ liệu duy nhất trên một cụm máy chủ trong một hay nhiều trung tâm dữ liệu. Dịch vụ này được điều khiển bởi một mảnh chìa khóa. Mảnh chìa khóa được sử dụng để phân phối các tài liệu thông minh trên nhiều trường hợp.
- Một dịch vụ truy vấn hỗ trợ quảng cáo hoc truy vấn, truy vấn phân tán, và tìm kiếm toàn văn bản.

Hiệu quả MongoDB triển khai bao gồm:

- Quản lý dung lượng nội dung lớn.
- Mạng xã hội
- Lưu trữ
- Phân tích thời gian thực

CouchDB

Một cơ sở dữ liệu không quan hệ rất phổ biến là CouchDB (http: // couchdb.apache.org). Giống như MongoDB, CouchDB là mã nguồn mở. Nó được duy trì bởi Quỹ Phần mềm Apache (www.apache.org) và được thực hiện theo Giấy phép Apache v2.0 có sẵn. Không giống như MongoDB, CouchDB được thiết kế để bắt chước các web trong tất cả các khía cạnh. Nó có trên một điện thoại thông minh hoặc trong một trung tâm dữ liệu. Tất cả điều này đi kèm với một vài cuộc đua thương mại. Bởi vì sự bắt chước web cơ bản, CouchDB có độ trễ cao, nó thích hợp cho việc lưu trữ dữ liệu khu vực. Mặc dù có khả năng làm việc một cách không phân phối, CouchDB cũng không thích hợp để triển khai thực hiện mô hình nhỏ hơn. Bạn phải xác định xem những đánh đổi này có thể được bỏ qua khi bạn bắt đầu thực hiện dữ liệu lớn của bạn.

Cơ sở dữ liệu CouchDB bao gồm các tài liệu bao gồm các lĩnh vực và các file đính kèm cũng như một "mô tả" của các tài liệu dưới dạng siêu dữ liệu sẽ tự động được duy trì bởi hệ thống. Các tính năng công nghệ cơ bản có tất cả các khả năng ACID. Ưu điểm trong CouchDB trên quan hệ là các dữ liệu được đóng gói và sẵn sàng cho các thao tác hoặc lưu trữ thay vì rải rác trên các hàng và bảng.

CouchDB cũng là một hệ sinh thái với các khả năng sau:

- Nén: Các cơ sở dữ liệu được nén để loại bỏ không gian lãng phí. Điều này giúp hiệu suất và hiệu quả cho sự bền bỉ.

- Xem mô hình: Một cơ chế để lọc, tổ chức, và báo cáo số liệu sử dụng một tập hợp các định nghĩa được lưu trữ như tài liệu trong cơ sở dữ liệu. Bạn tìm thấy một mối quan hệ một-nhiều cơ sở dữ liệu để xem, vì vậy bạn có thể tạo ra nhiều cách khác nhau để đại diện cho dữ liệu bạn đã "cắt lát và cắt nhỏ."
- Nhân rộng và phân phối các dịch vụ: lưu trữ tài liệu được thiết kế để cung cấp cho nhân rộng hai chiều. bản sao một phần có thể được duy trì để hỗ trợ phân phối theo tiêu chí hoặc di chuyển đến các thiết bị có kết nối hạn chế.

Hiệu quả của triển khai CouchDB bao gồm:

- Lượng lớn nội dung quản lý
- Mở rộng quy mô từ điện thoại thông minh thành trung tâm dữ liệu
- Các ứng dụng với kết nối mạng hạn chế hoặc chậm

1.2.1.5 Cơ sở dữ liệu cột

Cơ sở dữ liệu quan hệ là định hướng hàng, như các dữ liệu trong mỗi hàng của một bảng được lưu trữ với nhau. Trong một cột, hoặc cơ sở dữ liệu cột theo định hướng, các dữ liệu được lưu trữ trên các hàng. Mặc dù điều này có vẻ như một sự phân biệt tầm thường, nó là đặc tính cơ bản quan trọng nhất của cơ sở dữ liệu cột. Nó rất dễ dàng để thêm vào các cột, và họ có thể được thêm vào từng hàng, cung cấp sự linh hoạt tuyệt vời, hiệu suất và khả năng mở rộng. Khi bạn có khối lượng lớn và nhiều loại dữ liệu, bạn có thể muốn sử dụng một cơ sở dữ liệu cột. Nó rất dễ thích nghi, bạn chỉ cần tiếp tục bổ sung thêm cột.

Cơ sở dữ liệu HBase

Một trong những cơ sở dữ liệu cột phổ biến nhất là HBase (http://hbase.pache.org). Nó cũng là một dự án trong Apache Software Foundation phân phối theo Apache Software License v2.0.

Các thiết kế của HBase được mô phỏng trên BigTable của Google (một hình thức lưu trữ hiệu quả dữ liệu không quan hệ). Do đó, việc triển khai các HBase được đánh giá cao về khả năng mở rộng, rải rác, phân phối, bản đồ được sắp xếp đa chiều. Bản đồ được lập chỉ mục của một khóa hàng, trọng cột, và một dấu thời gian; mỗi giá trị trong bản đồ là một mảng byte. Khi thực hiện dữ liệu lớn của bạn đòi hỏi phải ngẫu nhiên, truy cập thời gian thực đọc / ghi dữ liệu. HBase là một giải pháp rất tốt. Nó thường được sử dụng để lưu trữ kết quả xử lý phân tích sau đó.

Đặc điểm quan trọng của HBase bao gồm những điều sau đây:

- Tính nhất quán: Mặc dù không phải là một "ACID" thực hiện, HBase cung cấp mạnh mẽ phù hợp đọc và viết và không dựa trên một mô hình phù hợp cuối cùng. Điều này có nghĩa là bạn có thể sử dụng nó cho các yêu cầu tốc độ cao miễn là bạn không cần "thêm tính năng" cung cấp bởi RDBMS như hỗ trợ giao dịch đầy đủ hoặc phân loại các cột.
- Chia nhỏ: Bởi vì dữ liệu được phân phối bởi hệ thống tập tin hỗ trợ, HBase tự động chia nhỏ và phân bố lại nội dung của nó.
- Tính sẵn sàng cao: Thông qua việc thực hiện của các máy chủ khu vực, HBase hỗ trợ chuyển đổi dự phòng và phục hồi mạng LAN và WAN. Tại cốt lõi, đó là một máy chủ tổng thể trách nhiệm giám sát các máy chủ khu vực và tất cả các siêu dữ liệu cho các cụm.
- Máy khách API: HBase cung cấp chương trình truy cập thông qua một API Java.
- Hỗ trợ cho các hoạt động CNTT: Người thực hiện có thể phơi bày hiệu suất và số liệu khác thông qua một tập hợp xây dựng trong các trang web.

Triển khai HBase phù hợp nhất cho:

- Khối lượng dữ liệu lớn, thu thập dữ liệu gia tăng và xử lý.
- Thời gian thực trao đổi thông tin.
- Thường xuyên thay đổi nội dung phục vụ.

1.2.1.6 Cơ sở dữ liệu đồ thị

Cấu trúc cơ bản cho cơ sở dữ liệu đồ thị được gọi là "nút - mối quan hệ." Cấu trúc này là hữu ích nhất khi bạn phải đối phó với các dữ liệu liên quan lẫn nhau. Các nút và các mối quan hệ hỗ trợ thuộc tính, một cặp khóa-giá trị nơi mà dữ liệu được lưu trữ. Các cơ sở dữ liệu được di chuyển theo các mối quan hệ. Kiểu lưu trữ và chuyển hướng là không thể trong RDBMS do các cấu trúc bảng cứng nhắc và không có khả năng theo dõi các kết nối giữa các dữ liệu bất cứ nơi nào. Một cơ sở dữ liệu đồ thị có thể được sử dụng để quản lý dữ liệu địa lý cho thăm dò dầu hoặc mô hình hóa và tối ưu hóa mạng lưới nhà cung cấp viễn thông.

Cơ sở dữ liệu Neo4J

Một trong những cơ sở dữ liệu đồ thị sử dụng rộng rãi nhất là Neo4J (www.neo4j.org). Đây là một dự án mã nguồn mở theo giấy phép GNU v3.0. Một hỗ trợ phiên bản thương mại được cung cấp bởi công nghệ Neo dưới v3.0 GNU AGPL và cấp giấy phép thương mại. Neo4J là một cơ sở dữ liệu giao dịch ACID cung cấp tính sẵn sàng cao thông qua cluster. Đó là một cơ sở dữ liệu đáng tin cậy và khả năng mở rộng đó dễ dàng để làm mô hình bởi vì cấu trúc cơ bản các tính nut-mối quan hệ. Nó không đòi hỏi một lược đồ, cũng không đòi hỏi dữ liệu đánh máy, do đó, nó rất linh hoạt.

Đặc điểm quan trọng của Neo4J bao gồm những điều sau đây:

- Tích hợp với các cơ sở dữ liệu khác: Neo4J hỗ trợ quản lý giao dịch để cho phép khả năng tương tác liền mạch với các cửa hàng dữ liệu phi đồ họa.
- Dịch vụ đồng bộ hóa: Neo4J hỗ trợ hành vi hướng sự kiện thông qua một xe buýt sự kiện, đồng bộ hóa định kỳ sử dụng chính nó, hoặc một RDBMS như là bậc thầy, và đồng bộ hàng loạt truyền thống.
- Khả năng phục hồi: Neo4J hỗ trợ lạnh (có nghĩa là, khi cơ sở dữ liệu không phải là chạy) và nóng (khi nó đang chạy) sao lưu, cũng như một chế độ phân nhóm sẵn sàng cao. Cảnh báo chuẩn có sẵn cho tích hợp với các hệ thống quản lý hoạt động hiện tại.
- Ngôn ngữ truy vấn: Neo4J hỗ trợ một ngôn ngữ khai báo tên là Cypher, thiết kế đặc biệt để truy vấn đồ thị và các thành phần của họ.Lệnh Cypher đang lỏng lẻo dựa trên cú pháp SQL và được nhắm mục tiêu vào quảng cáo truy vấn đặc biệt của dữ liệu đồ thị.

Triển khai Neo4J phù hợp nhất cho:

- Mạng xã hội
- Phân loại các lĩnh vực sinh học hoặc y tế
- Tạo cộng đồng năng động

1.2.1.7 Cơ sở dữ liệu không gian

Cho dù bạn có biết hay không, bạn có thể tương tác với dữ liệu không gian mỗi ngày. Nếu bạn sử dụng một điện thoại thông minh hoặc hệ thống định vị toàn cầu (GPS) để tìm đường đến một địa điểm cụ thể, hoặc nếu bạn hỏi một công cụ tìm kiếm cho các vị trí của nhà hàng, một địa chỉ vật lý hay địa danh, bạn đang sử dụng các ứng dụng dựa trên dữ liệu không gian. Dữ liệu không gian riêng được chuẩn hóa thông qua những nỗ lực của các tổ chức không gian địa lý (OGC; www.opengeospatial.org), trong đó thiết lập OpenGIS (Hệ thống thông tin địa lý) và một số tiêu chuẩn khác cho dữ liệu không gian.

Điều này quan trọng bởi vì cơ sở dữ liệu không gian là hiện thực của các tiêu chuẩn OGC, và công ty của bạn có thể có những nhu cầu cụ thể đáp ứng (hoặc không đáp ứng) theo các tiêu chuẩn. Một cơ sở dữ liệu không gian trở nên quan trọng khi tổ chức bắt đầu tận dụng kích thước khác nhau của dữ liệu để giúp đưa ra quyết định. Ví dụ, một nhà khí tượng nghiên cứu có thể muốn lưu trữ và đánh giá dữ liệu liên quan đến một cơn bão, bao gồm nhiệt độ, tốc độ gió, độ ẩm, và mô hình những kết quả trong ba chiều.

Ở dạng đơn giản nhất, cơ sở dữ liệu không gian lưu trữ dữ liệu về 2 chiều, 2,5 chiều và 3 chiều đối tượng. Bạn có thể quen với đối tượng 2D và 3D như chúng ta tương tác với tất cả các thời gian. Một đối tượng 2D có chiều dài và chiều rộng. Một đối tượng 3D thêm chiều sâu cho chiều dài và chiều rộng. 2.5D là gì? Đối tượng 2.5D là một loại đặc biệt của dữ liệu không gian. Chúng là đối tượng 2D với độ cao như thêm "một nửa" chiều. Hầu hết các cơ sở dữ liệu không gian 2.5D chứa thông tin bản đồ và thường được gọi là Hệ thống thông tin địa lý (GIS).

Các yếu tố của cơ sở dữ liệu không gian là những đường, điểm, và đa giác. Do tính chất đặc biệt của các đối tượng dữ liệu không gian, các nhà thiết kế tạo ra các cơ chế lập chỉ mục (chỉ số không gian) được thiết kế để hỗ trợ truy vấn và đại diện trực quan của các nội dung của cơ sở dữ liệu. Ví dụ, một chỉ số không gian sẽ trả lời các truy vấn hoặc “Có một dòng cụ thể giao nhau với một tập hợp các đa giác?” Nếu điều này có vẻ như một vấn đề lớn “khoảng cách giữa một điểm và một điểm là gì?”. Dữ liệu không gian cũng có thể đại diện cho các thách thức dữ liệu lớn nhất.

PostGIS / OpenGEO

Mặc dù các cơ sở dữ liệu là rất quan trọng, bạn cũng sẽ yêu cầu phần khác của công nghệ để giải quyết các yêu cầu ứng dụng không gian. May mắn thay, PostGIS là một phần của một hệ sinh thái của các thành phần được thiết kế để làm việc với nhau và giải quyết những nhu cầu này. Ngoài PostGIS, các OpenGEO Suite bao gồm những điều sau đây:

- Geoserver: Thực hiện trong Java, Geoserver có thể xuất bản thông tin không gian từ một số trong những nguồn chính của dữ liệu không gian trên web.
- OpenLayers: Một thư viện JavaScript hữu ích cho việc hiển thị bản đồ và các đại diện khác của dữ liệu không gian trong một trình duyệt web. Nó có thể thao tác hình ảnh từ hầu hết các nguồn bản đồ trên web, bao gồm Bing Maps, Google Maps, Yahoo Maps, OpenStreetMap…
- GeoExt: Được thiết kế để làm cho thông tin bản đồ từ OpenLayers có sẵn cho các nhà phát triển ứng dụng web. GeoExt có thể được sử dụng để tạo ra chỉnh sửa, xem, phong cách và kinh nghiệm web tương tác khác.
- GeoWebCache: Sau khi bạn có dữ liệu trong một máy chủ và có thể hiển thị nó trong một trình duyệt, bạn cần phải tìm một cách để làm cho nó nhanh. GeoWebCache là máy gia tốc. Nó lưu trữ khối dữ liệu hình ảnh và làm cho chúng sẵn sàng cho giao hàng nhanh chóng cho các thiết bị hiển thị.

Trong khi rất nhiều những ứng dụng của dữ liệu không gian liên quan đến bản đồ và địa điểm, dữ liệu không gian có nhiều ứng dụng hiện đại và tương lai khác, bao gồm:

- Mô hình 3D chính xác của cơ thể con người, các tòa nhà, không khí…
- Thu thập và phân tích dữ liệu từ các mạng cảm biến.
- Tích hợp với các dữ liệu lịch sử để kiểm tra không gian 3D / đối tượng theo thời gian.

1.2.1.8 Tồn tại với nhiều thứ tiếng

Định nghĩa chính thức của nhiều thứ tiếng là "một người nói hoặc viết một số ngôn ngữ". Thuật ngữ được vay mượn trong bối cảnh này và định nghĩa lại như một tập hợp các ứng dụng sử dụng một số công nghệ cơ sở dữ liệu cốt lõi, và đây là kết quả rất có thể lập kế hoạch thực hiện dữ liệu lớn của bạn . Nó sẽ khó khăn để lựa chọn một vấn đề như thế nào thu hẹp cách tiếp cận của bạn để dữ liệu lớn có thể được. Một cơ sở dữ liệu duy trì với nhiều thứ tiếng được sử dụng khi nó cần thiết để giải quyết một vấn đề phức tạp bằng cách phá vỡ vấn đề đó vào phân đoạn và áp dụng mô hình cơ sở dữ liệu khác nhau. Sau đó tổng hợp các kết quả vào một lưu trữ dữ liệu lai và giải pháp phân tích. Một số yếu tố ảnh hưởng đến quyết định này:

- Nếu doanh nghiệp hoặc tổ chức của bạn lớn, bạn có thể sử dụng nhiều RDBMS, kho dữ liệu, siêu thị dữ liệu, các tập tin phẳng, máy chủ quản lý nội dung... Môi trường lai này là phổ biến, và bạn có thể quyết định hội nhập, phân tích dữ liệu, tầm nhìn dữ liệu... Bạn cần phải hiểu tất cả điều đó vì bạn cần phải tìm hiểu làm thế nào nó sẽ phù hợp với việc thực hiện dữ liệu lớn của bạn.
- Các lý tưởng nhất của môi trường, nơi mà bạn chỉ có một công nghệ duy trì, có lẽ không phù hợp để giải quyết vấn đề lớn dữ liệu. Ít nhất, bạn sẽ cần phải giới thiệu một kiểu cơ sở dữ liệu và các công nghệ hỗ trợ khác để thực hiện.
- Tùy thuộc vào sự đa dạng và tốc độ thu thập dữ liệu lớn, bạn có thể cần phải xem xét cơ sở dữ liệu khác nhau để hỗ trợ một thực hiện. Bạn cũng nên xem xét yêu cầu của bạn cho toàn vẹn giao dịch. Bạn cần hỗ trợ tuân thủ ACID hay tuân thủ BASE là đủ?

Đây là một thách thức dữ liệu lớn. Nhiều nguồn dữ liệu có cấu trúc rất khác nhau cần phải được thu thập và phân tích để các bạn có thể nhận được câu trả lời cho những câu hỏi này. Sau đó, bạn cần xác định xem liệu các khách hàng đủ điều kiện để thúc đẩy trong thời gian thực, cung cấp cho họ một cái gì đó mới và thú vị.

Đây là loại vấn đề không thể được giải quyết một cách dễ dàng hoặc chi phí hiệu quả với một loại công nghệ cơ sở dữ liệu. Mặc dù một số thông tin cơ bản là giao dịch và có lẽ trong một RDBMS, các thông tin khác là không quan hệ và sẽ cần ít nhất hai loại phương tiện duy trì (không gian và đồ thị).

1.2.2 Thiết bị và kho dữ liệu lớn

1.2.2.1 Tích hợp dữ liệu lớn với các kho dữ liệu truyền thống

Không giống như các hệ thống cơ sở dữ liệu hoạt động truyền thống và các ứng dụng, các kho dữ liệu đã được sử dụng bởi các ngành nghề kinh doanh và các nhà phân tích tài chính giúp đưa ra quyết định về hướng đi của một chiến lược kinh doanh. Dữ liệu đã được thu thập từ nhiều nguồn cơ sở dữ liệu quan hệ khác nhau, sau đó đảm bảo rằng các siêu dữ liệu là phù hợp, và các dữ liệu là không có lỗi và sau đó tích hợp tốt. Bill Inmon, được coi là cha đẻ của các kho dữ liệu hiện đại, thành lập một tập hợp các nguyên tắc của các kho dữ liệu, trong đó bao gồm các đặc điểm sau:

- Nó nên là đối tượng theo định hướng.
- Nó cần được tổ chức để các sự kiện liên quan được liên kết với nhau.
- Các thông tin không thể vô tình thay đổi.
- Thông tin trong kho nên bao gồm tất cả các nguồn hoạt động áp dụng. Các thông tin cần được lưu trữ trong một cách có định nghĩa thống nhất.

1.2.2.2 Phân tích dữ liệu lớn và các kho dữ liệu

Bạn cần phải tạo ra một môi trường lai, nơi dữ liệu lớn có thể làm việc với kho dữ liệu. Đầu tiên, điều quan trọng là nhận ra các kho dữ liệu vì ngày nay nó được thiết kế sẽ không thay đổi trong ngắn hạn. Vì vậy, nó thực dụng hơn để sử dụng kho dữ liệu cho những gì nó đã được thiết kế để làm - cung cấp một phiên bản tốt. Các kho có thể bao gồm thông tin về các dòng của một công ty cụ thể sản phẩm, khách hàng, nhà cung cấp của nó, và các chi tiết của giá trị của giao dịch trong một năm. Các thông tin quản lý trong kho dữ liệu của bộ đã được xây dựng một cách cẩn thận để siêu dữ liệu là chính xác. Với sự phát triển của thông tin trên web mới, nó thực tế và thường cần thiết để phân tích số lượng lớn các dữ liệu này trong bối cảnh với các dữ liệu lịch sử. Đây là nơi mà các mô hình lai đến.

Rất nhiều các nguồn dữ liệu lớn đến từ các nguồn bao gồm siêu dữ liệu được thiết kế riêng của họ. Các trang web thương mại điện tử phức tạp bao gồm các yếu tố được xác định rõ dữ liệu (khách hàng, giá cả, và do đó trên). Do đó, khi tiến hành phân tích giữa các kho hàng và các nguồn dữ liệu lớn, các tổ chức quản lý thông tin đang làm việc với hai bộ dữ liệu và mô hình siêu dữ liệu thiết kế cẩn thận mà phải được hợp lý hóa.

Trước khi một nhà phân tích có thể kết hợp lịch suer giao dịch các dữ liệu với các dữ liệu lớn ít có cấu trúc, công việc đã được thực hiện. Thông thường, phân tích ban đầu của dữ liệu petabytes sẽ tiết lộ những điều thú vị mà có thể giúp dự đoán những thay đổi tinh tế trong giải pháp kinh doanh hoặc tiềm năng để chẩn đoán một bệnh nhân. Các phân tích ban đầu có thể được hoàn thành các công cụ như MapReduce tận dụng với khuôn khổ hệ thống tập tin Hadoop phân phối. Tại thời điểm này, bạn có thể bắt đầu hiểu được cho dù nó có thể giúp đánh giá các vấn đề được giải quyết. Trong quá trình phân tích, nó cũng quan trọng để loại bỏ dữ liệu không cần thiết vì nó là để xác định dữ liệu có liên quan đến bối cảnh kinh doanh. Khi giai đoạn này hoàn tất, các dữ liệu còn lại cần được chuyển hóa để định nghĩa siêu dữ liệu là chính xác. Bằng cách này, khi các dữ liệu lớn được kết hợp với truyền thống, dữ liệu lịch sử từ các kho hàng, các kết quả sẽ chính xác và có ý nghĩa.

1.2.2.3 Thay đổi vai trò của kho dữ liệu

Nó rất hữu ích để suy nghĩ về những điểm tương đồng và khác biệt giữa cách thức dữ liệu được quản lý trong kho dữ liệu truyền thống và khi kho được kết hợp với dữ liệu lớn. Những điểm tương đồng giữa hai phương pháp quản lý dữ liệu bao gồm:

- Yêu cầu đối với các định nghĩa dữ liệu chung
- Yêu cầu để trích xuất và chuyển đổi các nguồn dữ liệu quan trọng
- Sự cần thiết phải phù hợp với các quy trình kinh doanh cần thiết và quy tắc

Sự khác biệt giữa các kho dữ liệu truyền thống và dữ liệu lớn bao gồm:

- Mô hình tính toán phân tán của dữ liệu lớn sẽ là cần thiết để cho phép các mô hình lai hoạt động.
- Phân tích dữ liệu lớn sẽ là trọng tâm chính của những nỗ lực, trong khi các kho dữ liệu truyền thống sẽ được sử dụng để thêm bối cảnh kinh doanh lịch sử và giao dịch.

CHƯƠNG 2: XÂY DỰNG KHO DỮ LIỆU VĂN BẢN

Hầu hết các dữ liệu là phi cấu trúc. Dữ liệu phi cấu trúc bao gồm các thông tin được lưu trữ nội bộ, chẳng hạn như tài liệu, e-mail, và thư từ của khách hàng, cũng như các nguồn thông tin bên ngoài rất quan trọng cho tổ chức của bạn (tweet, blog, video YouTube, và hình ảnh vệ tinh). Số lượng và sự đa dạng loại dữ liệu này được phát triển nhanh chóng. Ngày càng có nhiều công ty muốn tận dụng lợi thế dữ liệu để phát triển doanh nghiệp của họ ngày hôm nay và trong tương lai.

Trong khi phân tích hình ảnh và âm thanh vẫn còn đang trong giai đoạn đầu, phân tích văn bản được phát triển thành một công nghệ chủ đạo. Dưới đây là một ví dụ về làm thế nào một công ty có thể tận dụng dữ liệu văn bản của mình để hỗ trợ việc ra quyết định kinh doanh. Một nhà sản xuất ô tô lớn cần thiết để cải thiện các vấn đề chất lượng với chiếc xe của mình. Họ phát hiện ra rằng bằng cách phân tích các văn bản từ các đối tác sửa chữa, họ có thể xác định các vấn đề chất lượng với chiếc xe của mình khi tham gia vào thị trường. Công ty phân tích này xem như một hệ thống cảnh báo sớm. Trước đó họ có thể xác định những vấn đề, những thay đổi họ có thể thực hiện trên sàn nhà máy. Trước khi sử dụng phân tích văn bản, các công ty khai thác thông tin từ dòng các hệ thống kinh doanh. Các hệ thống truyền thống không thể tiết lộ những vấn đề ẩn.

Trong thực tế, phân tích văn bản đang được sử dụng trong một loạt các trường hợp sử dụng dữ liệu lớn từ phân tích phương tiện truyền thông xã hội đếm phân tích bảo hành, phân tích lừa đảo. Ngoài ra, các doanh nghiệp đang bắt đầu phân tích một cái nhìn hợp nhất dữ liệu có cấu trúc và phi cấu trúc với nhau để có được một bức tranh đầy đủ. Trong chương này, sẽ đi sâu vào công nghệ này và cung cấp một chiều sâu ví dụ về cách thức hoạt động. Đồng thời cũng cung cấp một số trường hợp sử dụng khác của phân tích văn bản trong hành động, bao gồm cả khả năng để kết hợp dữ liệu phi cấu trúc với các dữ liệu có cấu trúc. Kết thúc chương với tên của một số nhà cung cấp đang cung cấp các công cụ phân tích văn bản cho dữ liệu lớn.

2.1 Khám phá dữ liệu phi cấu trúc

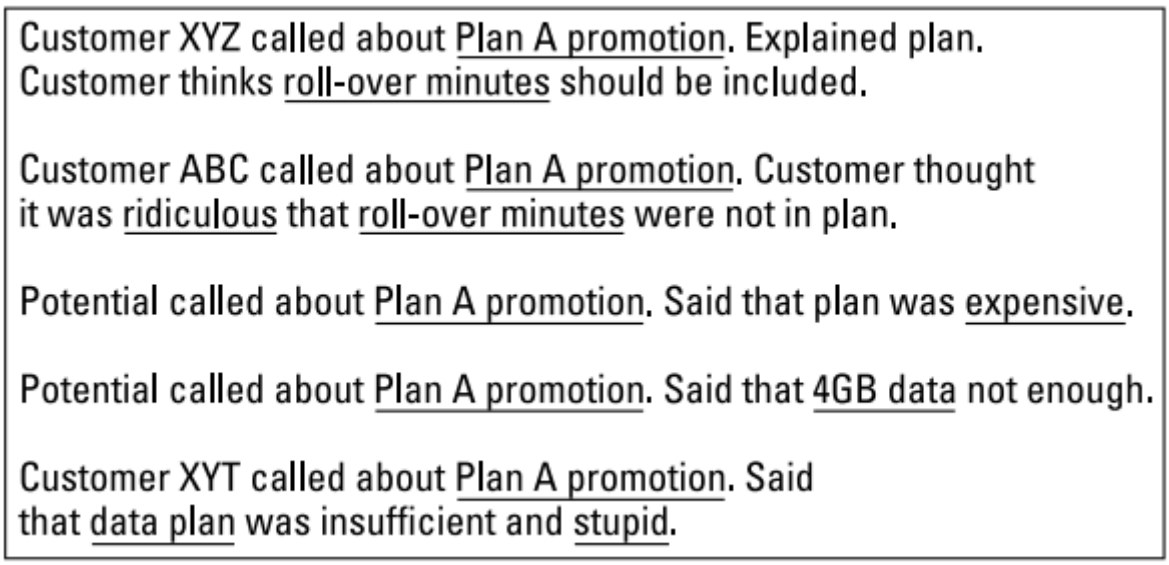
Dữ liệu phi cấu trúc hiểu đơn giản là cấu trúc của dữ liệu đó không thể đoán biết được. Một số người cho rằng dữ liệu phi cấu trúc giới hạn là sai lầm bởi vì mỗi nguồn văn bản có thể chứa các cấu trúc cụ thể cho riêng mình hoặc định dạng được dựa trên phần mềm tạo ra nó. Trong thực tế nội dung của các tài liệu thực sự không có cấu trúc.

Ví dụ, một note cho vay ngân hàng có một số cấu trúc về câu. Một e-mail có thể có cấu trúc nhỏ. Một tweet hoặc tin nhắn Facebook có thể có chữ viết tắt lạ hoặc ký tự. Một tập tin log có thể có cấu trúc riêng của mình. Vì vậy, câu hỏi là, làm thế nào để bạn phân tích loại khác nhau của dữ liệu phi cấu trúc văn bản?

2.2 Tìm hiểu về phân tích văn bản

Nhiều phương pháp tồn tại cho việc phân tích dữ liệu phi cấu trúc. Trong lịch sử, những kỹ thuật này ra khỏi khu vực kỹ thuật như xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), phát hiện kiến thức, khai thác dữ liệu, tìm kiếm thông tin và thống kê. Phân tích văn bản là quá trình xử lý văn bản phi cấu trúc, giải nén thông tin có liên quan, và biến nó thành thông tin có cấu trúc mà sau đó có thể được tận dụng theo những cách khác nhau. Các quá trình phân tích và khai thác tận dụng lợi thế của kỹ thuật có nguồn gốc từ ngôn ngữ toán học, thống kê và các ngành khoa học máy tính khác.

Giả sử rằng bạn làm việc cho bộ phận tiếp thị của một công ty điện thoại không dây. Bạn vừa tung ra hai kế hoạch kêu gọi mới - Kế hoạch A và kế hoạch B - và bạn không nhận được sự tiếp nhận mà bạn muốn vào Kế hoạch A. Các văn bản không có cấu trúc từ các ghi chú trung tâm cuộc gọi có thể cung cấp cho bạn một số thông tin như tại sao điều này xảy ra.


Customer XYZ called about Plan A promotion. Explained plan.
Customer thinks roll-over minutes should be included.

Customer ABC called about Plan A promotion. Customer thought
it was ridiculous that roll-over minutes were not in plan.

Potential called about Plan A promotion. Said that plan was expensive.

Potential called about Plan A promotion. Said that 4GB data not enough.

Customer XYT called about Plan A promotion. Said
that data plan was insufficient and stupid.

Hình 2-1: Mẫu hồ sơ cuộc gọi

Các từ được gạch chân cung cấp các thông tin mà bạn có thể cần phải hiểu lý do tại sao kế hoạch A không áp dụng nhanh chóng. Ví dụ, các thông tin kế hoạch A xuất hiện trong suốt những cuộc gọi chỉ ra rằng các báo cáo đề cập đến kế hoạch. Roll-over minutes, 4GB data, data plan, và expensive là bằng chứng cho thấy một vấn đề tồn tại với kế hoạch dữ liệu và giá cả. Những từ như vô lý và ngu ngốc cung cấp mức độ đánh giá của khách hàng.

Quá trình phân tích văn bản sử dụng thuật toán khác nhau, chẳng hạn như hiểu cấu trúc câu, để phân tích văn bản phi cấu trúc và sau đó trích xuất thông tin, và chuyển thông tin vào dữ liệu có cấu trúc.

Making Structured Data from Unstructured Text			
Identifier	**Entity**	**Issue**	**Sentiment**
Cust XYZ	Plan A	Roll-over minutes	Neutral
Cust ABC	Plan A	Roll-over minutes	Negative
XXXX	Plan A	Expensive	Neutral
Cust XYT	Plan A	Data plan	Negative

Bảng 2-1: Chuyển văn bản phi cấu trúc thành dữ liệu có cấu trúc

Bạn có thể xem xét điều này và nói: “Nhưng tôi có thể đã đoán ra bằng cách nhìn vào hồ sơ trung tâm cuộc gọi”. Tuy nhiên, đây chỉ là một phần nhỏ của các thông tin được ghi lại bởi hàng ngàn đại lý trung tâm cuộc gọi. Mỗi đại lý cá nhân không thể đảm đương được một lượng thông tin quá rộng liên quan đến các vấn đề được cung cấp bởi các công ty. Đại lý không có thời gian và yêu cầu chia sẻ thông tin này trên tất cả các đại lý trung tâm cuộc gọi khác, những người có thể nhận được con số tương tự của các cuộc gọi về Kế hoạch A. Tuy nhiên, sau khi thông tin này được tổng hợp và xử lý bằng các thuật toán phân tích văn bản, một xu hướng có thể xuất hiện từ dữ liệu phi cấu trúc này. Đó là những gì làm cho phân tích văn bản.

Sự khác biệt phân tích văn bản và tìm kiếm

Chú ý rằng đây là giải nén văn bản, không phải trên từ khóa tìm kiếm. Tìm kiếm là lấy về một tài liệu dựa trên những gì người dùng đã biết họ đang tìm kiếm. Phân tích văn bản là việc khám phá ra thông tin. Trong khi phân tích văn bản khác với tìm kiếm, nó có thể làm tăng thêm kỹ thuật tìm kiếm. Ví dụ, phân tích văn bản kết hợp với tìm kiếm có thể được sử dụng để cung cấp phân loại tốt hơn.

Query, Data Mining, Search and Text Analytics		
	Retrieval	**Insight**
Structured	Query: Return data	Data mining: Insight from structured data
Unstructured	Search: Returns documents	Text analytics: Insight from text

Bảng 2-2: Truy vấn, khai thác dữ liệu, tìm kiếm và phân tích văn bản

Ở phía bên trái của bảng là truy vấn và tìm kiếm, mà là cả hai về thu hồi. Ví dụ, một người dùng cuối có thể truy vấn một cơ sở dữ liệu để tìm ra bao nhiêu khách hàng ngừng sử dụng dịch vụ của công ty trong tháng vừa qua. Các truy vấn sẽ trả về một số duy nhất. Chỉ bằng cách hỏi nhiều hơn và khác nhau truy vấn sẽ cho người dùng cuối có được các thông tin cần thiết để xác định lý do tại sao khách hàng đang rời. Tương tự như vậy, từ khóa tìm kiếm cho phép người dùng cuối để tìm các tài liệu có chứa tên của các đối thủ cạnh tranh của công ty. Việc tìm kiếm sẽ trả về một nhóm tài liệu. Chỉ có bằng cách đọc các tài liệu này sẽ cho người dùng cuối đến với bất kỳ câu trả lời liên quan đến câu hỏi của mình.

Các công nghệ trên thu thập các mảnh thông tin và yêu cầu tương tác của con người để tổng hợp và phân tích các thông tin đó. Các công nghệ trên bên phải: khai thác dữ liệu và phân tích văn bản cung cấp cái nhìn sâu sắc nhanh hơn nhiều.

2.3 Phân tích và kỹ thuật khai thác

Nhìn chung, giải pháp phân tích văn bản sử dụng một sự kết hợp của kỹ thuật thống kê và xử lý ngôn ngữ tự nhiên (NLP) để trích xuất thông tin từ dữ liệu phi cấu trúc. NLP là một lĩnh vực rộng lớn và phức tạp đã được phát triển trong vòng 20 năm qua. Mục tiêu chính của NLP là ý nghĩa từ văn bản. Xử lý ngôn ngữ tự nhiên thường sử dụng các khái niệm ngôn ngữ học như cấu trúc ngữ pháp và các bộ phận của bài phát biểu. Thông thường, ý tưởng đằng sau kiểu phân tích này là xác định đã làm gì với ai, khi nào, ở đâu, như thế nào, và tại sao.

NLP thực hiện phân tích văn bản ở các cấp độ khác nhau:

- Từ vựng / phân tích hình thái xem xét các đặc điểm của một từ cụ thể - bao gồm các tiền tố, hậu tố, nguồn gốc, và thành phần của câu (danh từ, động từ, tính từ, vv) - thông tin sẽ góp phần vào sự hiểu biết trong bối cảnh của văn bản cung cấp. Phân tích từ vựng phụ thuộc vào từ điển, từ điển đồng nghĩa, hoặc bất kỳ danh sách cung cấp thông tin về những từ đó. Trong trường hợp xúc tiến bán hàng của một công ty truyền thông không dây, một từ điển có thể cung cấp các thông tin về vị trí, một nỗ lực quảng cáo, hoặc một nỗ lực để khuyến khích sự phát triển của một ai đó. Phân tích từ vựng cũng sẽ cho phép một ứng dụng để nhận ra rằng việc thúc đẩy, khuyến mãi, và phát huy được tất cả các phiên bản của cùng một từ và ý tưởng.
- Phân tích cú pháp sử dụng cấu trúc ngữ pháp để phân tích các văn bản và đưa từ riêng lẻ vào ngữ cảnh. Ở đây bạn được mở rộng cái nhìn từ một từ duy nhất đến các cụm từ hoặc toàn bộ văn bản. Bước này có thể lập sơ đồ mối quan hệ giữa các từ (ngữ pháp) hoặc tìm kiếm các trình tự hình thành câu đúng hay cho chuỗi số đại diện cho ngày tháng hoặc giá trị tiền tệ.
- Phân tích ngữ nghĩa xác định nghĩa của một câu. Điều này có thể bao gồm kiểm tra trật tự từ và cấu trúc câu và làm cho chúng rõ ràng bằng cách liên hệ cú pháp tìm thấy trong các cụm từ, câu, đoạn văn.
- Phân tích cấp độ văn bản cố gắng xác định ý nghĩa của văn bản vượt quá mức độ câu.

Trong thực tế, để trích xuất thông tin từ các nguồn tài liệu khác nhau, các tổ chức cần phải phát triển các quy tắc. Tất nhiên, các quy tắc có thể phức tạp hơn nhiều. Các tổ chức có thể tạo ra quy tắc tay, tự động, hoặc bằng cách kết hợp cả hai phương pháp:

- Trong cách tiếp cận sử dụng, ai đó sử dụng một ngôn ngữ độc quyền xây dựng một loạt các quy tắc để khai thác. Người này cũng có thể xây dựng các từ điển và danh sách từ đồng nghĩa. Trong khi các phương pháp thủ công có thể tốn nhiều thời gian, nó có thể cung cấp kết quả rất chính xác.

- Cách tiếp cận tự động có thể sử dụng máy học hoặc kỹ thuật thống kê khác. Phần mềm này tạo quy tắc dựa trên một tập hợp của dữ liệu văn bản. Đầu tiên, hệ thống xử lý một tập hợp các tài liệu tương tự (ví dụ, các bài báo) để phát triển có nghĩa là học các quy tắc. Sau đó, người dùng chạy một tập hợp các dữ liệu thử nghiệm để kiểm tra tính chính xác của các quy tắc.

2.3.1 Tìm hiểu thu thập thông tin

Các kỹ thuật được mô tả trước đó trong chương này thường được kết hợp với các kỹ thuật thống kê hoặc ngôn ngữ khác, tự động gắn thẻ và đánh dấu các tài liệu văn bản để trích xuất các loại thông tin sau đây:

- Định danh: Một cái tên khác thay cho từ khóa.
- Đối tượng: Thường được gọi tên là các thực thể, đây là những ví dụ cụ thể trừu tượng hóa (hữu hình hay vô hình). Ví dụ như tên người, tên công ty, vị trí địa lý, thông tin liên lạc, ngày, giờ, tiền tệ, chức danh và vị trí, ... Ví dụ: phần mềm phân tích văn bản có thể trích xuất các thực thể Jane Doe là một người được đề cập trong văn bản được phân tích. Các tổ chức ngày 03 tháng 03 năm 2007 có thể được chiết xuất như một ngày, ...
- Sự việc: Còn được gọi là các mối quan hệ, sự kiện chỉ ra ai / cái gì / nơi mối quan hệ giữa hai thực thể
- Sự kiện: Trong khi một số chuyên gia sử dụng các điều khoản thực tế, mối quan hệ, và sự kiện thay thế cho nhau, những người khác phân biệt giữa sự việc và sự kiện, nói rằng sự kiện thường chứa một chiều thời gian và gây ra sự việc thay đổi. Ví dụ như một sự thay đổi trong quản lý trong công ty hoặc tình trạng của một quá trình bán hàng cũng được gọi là các mối quan hệ, sự kiện chỉ ra ai / cái gì / nơi mối quan hệ giữa hai thực thể.
- Khái niệm: Đây là bộ các từ và cụm từ chỉ ra một ý tưởng cụ thể hoặc chủ đề mà người dùng quan tâm. Điều này có thể được thực hiện bằng tay hoặc bằng cách sử dụng thống kê, hoặc phương pháp tiếp cận để phân loại. Ví dụ, khái niệm khách hàng không hài lòng có thể bao gồm những lời giận dữ, thất vọng và các cụm từ ngắt kết nối dịch vụ, không gọi lại, và lãng phí tiền bạc. Do đó, khái niệm khách hàng không hài lòng có thể được chiết xuất thậm chí không có những lời phàn nàn của khách hàng hoặc xuất hiện trong văn bản. Các khái niệm có thể được định nghĩa bởi người sử dụng cho phù hợp với nhu cầu cụ thể của họ.

- Cảm xúc: Phân tích tâm lý thị trường được sử dụng để xác định những quan điểm hoặc cảm xúc trong văn bản cơ bản. Một số kỹ thuật làm điều này bằng cách phân loại văn bản như: chủ quan (ý kiến) hay khách quan (thực tế), sử dụng học máy hoặc kỹ thuật NLP. Phân tích cảm xúc đã trở nên rất phổ biến trong "tiếng nói của khách hàng" các loại ứng dụng.

2.3.2 Nguyên tắc phân loại

Nguyên tắc phân loại thường quan trọng đối với văn bản phân tích. Phân loại tư duy là một phương pháp để tổ chức thông tin vào các mối quan hệ thứ bậc. Nó đôi khi được gọi là một cách tổ chức phạm trù. Bởi vì một nguyên tắc phân loại xác định mối quan hệ giữa các điều khoản một công ty sử dụng, nó làm cho dễ dàng hơn để tìm và sau đó phân tích văn bản.

Ví dụ, một nhà cung cấp dịch vụ viễn thông cung cấp cả dịch vụ có dây và không dây. Trong dịch vụ không dây, các công ty có thể hỗ trợ điện thoại di động và truy cập Internet. Sau đó công ty có thể có hai hay nhiều cách phân loại các dịch vụ điện thoại di động, chẳng hạn như các kế hoạch và các loại điện thoại. Việc phân loại có thể đạt được tất cả các con đường xuống đến các bộ phận của một chiếc điện thoại riêng của mình.

Tất cả các nguyên tắc phân loại cũng có thể sử dụng từ đồng nghĩa và biểu thức thay thế, nhận ra rằng điện thoại di động đều giống nhau. Những nguyên tắc phân loại có thể khá phức tạp và có thể mất một thời gian dài để phát triển.

Một số nhà cung cấp cho rằng phân loại là không cần thiết khi sử dụng sản phẩm của họ và người dùng doanh nghiệp có thể phân loại các thông tin đã được chiết xuất. Điều này sẽ thực sự phụ thuộc vào đối tượng bạn quan tâm. Thông thường, các chủ đề có thể rất phức tạp, sắc thái, hoặc cụ thể cho một ngành công nghiệp nhất định. Điều đó sẽ đòi hỏi một phân loại tập trung.

2.4 Đưa kết quả cùng với dữ liệu có cấu trúc

Sau khi dữ liệu phi cấu trúc của bạn có cấu trúc, bạn có thể kết hợp nó với các thông tin có cấu trúc khác mà có thể tồn tại trong kho dữ liệu của bạn, và sau đó áp dụng các công cụ kinh doanh thông minh hoặc khai thác dữ liệu để thu thập cái nhìn sâu sắc hơn nữa.

Marrying Structured and Unstructured Data				
Identifier	**Entity**	**Issue**	**Sentiment**	**Segment**
Cust XYZ	Plan A	Roll-over minutes	Neutral	Gold
Cust ABC	Plan A	Roll-over minutes	Negative	Silver
XXXX	Plan A	Expensive	Neutral	XXX
Cust XYT	Plan A	Data plan	Negative	Bronze

Bảng 2-3: Kết hợp dữ liệu có cấu trúc và dữ liệu không có cấu trúc

Ví dụ, kết quả phân tích văn bản được sáp nhập với thông tin thanh toán có cấu trúc. Về cơ bản, bạn có thể kết hợp thông tin từ các khách hàng sống trong hệ thống thanh toán với các thông tin từ các ghi chú trung tâm cuộc gọi. Tất nhiên, khi khách hàng gọi tới, không có thông tin để phù hợp; đây là lý do tại sao "XXXX" xuất hiện trong những hàng này.

Trong ví dụ này, dữ liệu có cấu trúc cùng với các dữ liệu phi cấu trúc cho thấy ít nhất một trong những khách hàng của bạn là một khách hàng vàng, do đó, nó sẽ có giá trị cho công ty để thực hiện một nỗ lực để giữ chân họ. Tất nhiên, trong thực tế, bạn sẽ có nhiều dữ liệu hơn để làm việc.

2.5 Đưa dữ liệu lớn sử dụng

Các trường hợp sử dụng khuyến mại không dây chỉ là một ví dụ về cách phân tích văn bản có thể được sử dụng giúp hiểu sâu hơn về dữ liệu. Một trường hợp sử dụng dữ liệu lớn có nghĩa là các dữ liệu phi cấu trúc được phân tích hoặc là khối lượng lớn, tốc độ cao, hoặc cả hai. Các phần sau đây mô tả một vài ví dụ.

a. Tiếng nói của khách hàng

Tối ưu hóa các trải nghiệm của khách hàng và cải thiện duy trì khách hàng là động lực chủ đạo cho nhiều ngành công nghiệp dịch vụ. Các tổ chức có liên quan với các vấn đề này có thể hỏi những câu hỏi như:

- Khách hàng thường phàn nàn về những vấn đề gì và làm cách nào để có thể thay đổi theo thời gian?
- Mức độ hài lòng của khách hàng với các dịch vụ cụ thể là gì?
- Các vấn đề thường gặp dẫn đến sự không hài lòng của khác hàng?

Thông tin, chẳng hạn như e-mail cho các công ty, các cuộc khảo sát sự hài lòng của khách hàng, ghi chú trung tâm cuộc gọi, và các tài liệu nội bộ khác, giữ rất nhiều thông tin về mối quan tâm của khách hàng và tình cảm. Phân tích văn bản có thể giúp xác định nguyên nhân của sự không hài lòng của khách hàng một cách kịp thời. Nó có thể giúp cải thiện hình ảnh thương hiệu bằng cách chủ động giải quyết các vấn đề trước khi chúng trở thành một điều rắc rối lớn với khách hàng.

Đây có phải là một vấn đề dữ liệu lớn? Nó có thể. Nó phụ thuộc vào khối lượng của thông tin. Bạn có thể có một khối lượng lớn thông tin được cung cấp trong chế độ hàng loạt. Các công ty có thể muốn kết hợp dữ liệu này với dữ liệu có cấu trúc.

b. Phân tích truyền thông xã hội

Một hình thức thể hiện tiếng nói của khách hàng là phân tích truyền thông xã hội. Nó đã thể hiện được rất nhiều khả năng trong thời gian gần đây, và trong thực tế, giúp thị trường phân tích văn bản. Trong phân tích truyền thông xã hội, dữ liệu qua Internet đang tập hợp lại với nhau. Điều này bao gồm văn bản phi cấu trúc từ các blog, microblog, các bài báo, văn bản từ các diễn đàn trực tuyến... Luồng dữ liệu khổng lồ sau đó được phân tích - thường sử dụng phân tích văn bản - để có được câu trả lời cho những câu hỏi như:

- Người ta đang nói về thương hiệu của tôi là gì?
- Họ thích những gì về thương hiệu của tôi?
- Họ không thích điều gì về thương hiệu của tôi?
- Làm thế nào để thương hiệu của tôi so sánh với các đối thủ cạnh tranh?

Phương tiện truyền thông xã hội không chỉ được sử dụng bởi các nhà tiếp thị liên quan về thương hiệu của họ. Chính phủ đang sử dụng nó để tìm kiếm cuộc hội thoại khủng bố. Cơ quan y tế đang sử dụng nó để xác định các mối đe dọa sức khỏe cộng đồng trên toàn thế giới.

Đó là một trường hợp sử dụng dữ liệu lớn, đặc biệt là khi bạn có thể làm việc với một nhà cung cấp dịch vụ có thể lắp ráp tất cả các tweet từ Twitter, cùng với tất cả các dữ liệu khác.

2.6 Công cụ phân tích văn bản cho Big data

2.6.1 Attensity

Attensity (www.attensity.com) là một trong những công ty phân tích văn bản ban đầu mà đã bắt đầu phát triển và bán các sản phẩm hơn mười năm trước đây. Tại thời điểm này, nó có hơn 150 khách hàng doanh nghiệp và là một trong nhóm phát triển NLP lớn nhất thế giới. Attensity cung cấp nhiều công cụ để phân tích văn bản. Chúng bao gồm tự động phân loại, trích xuất thực thể, và khai thác đầy đủ. Khai thác đầy đủ là công nghệ hàng đầu của Attensity, tự động trích xuất các dữ kiện từ văn bản phân tích cú pháp (người đã làm gì với ai, khi nào, ở đâu, dưới những điều kiện) và tổ chức các thông tin này.

Công ty đang tập trung vào phân tích xã hội đa kênh và tham gia bằng cách phân tích văn bản để báo cáo từ các nguồn nội bộ và bên ngoài, sau đó định tuyến cho người dùng doanh nghiệp để tham gia. Gần đây họ đã mua Biz360, một công ty truyền thông xã hội mà tập hợp các luồng khổng lồ của phương tiện truyền thông xã hội. Nó đã phát triển một hệ thống tính toán lưới cung cấp khả năng highperformance cho xử lý một lượng lớn các văn bản thời gian thực. Attensity sử dụng một khuôn khổ Hadoop (MapReduce, HDFS, và HBase) để lưu trữ dữ liệu. Nó cũng có một hệ thống dữ liệu hàng đợi mà và điều chỉnh phương pháp qua nhiều máy chủ khi cần thiết.

2.6.2 Clarabridge

Clarabridge là một nhà cung cấp phân tích văn bản. Clarabridge là một sản phẩm trí tuệ doanh nghiệp của công ty tư vấn (gọi là Claraview) mà nhận ra sự cần thiết để đối phó với các dữ liệu phi cấu trúc. Mục tiêu của nó là để giúp các công ty nâng cao giá trị kinh doanh đo lường trước được bằng cách nhìn vào các khách hàng một cách tổng thể, xác định rõ những kinh nghiệm quan trọng và các vấn đề, giúp đỡ tất cả mọi người trong một tổ chức có những hành động và hợp tác trong thời gian thực. Điều này bao gồm việc xác định thời gian thực của tình cảm và phân loại các thông tin phản hồi của khách hàng về dữ liệu văn bản vào hệ thống Clarabridge.

Tại thời điểm này, Clarabridge là cung cấp cho khách hàng một số tính năng phức tạp và thú vị, bao gồm nhấp chuột đơn để phân tích, xác định những gì đang gây ra một sự thay đổi trong khối lượng văn bản, tình cảm, hay sự hài lòng liên quan đến các vấn đề đang nổi lên. Nó cũng cung cấp các giải pháp như là một phần mềm dịch vụ (SaaS).

2.6.3 IBM

Phần mềm khổng lồ IBM (www.ibm.com) cung cấp một số giải pháp trong không gian phân tích văn bản dựa trên chiến lược thông minh Planet. Ngoài Watson và IBM SPSS, IBM cũng cung cấp phân tích nội dung với tìm kiếm doanh nghiệp (ICAES). Phân tích nội dung IBM được phát triển dựa trên công việc thực hiện tại Viện nghiên cứu của IBM.

Phân tích nội dung của IBM được sử dụng để chuyển đổi nội dung vào các thông tin phân tích, và điều này là có sẵn cho các phân tích chi tiết tương tự như cách cấu trúc dữ liệu sẽ được phân tích trong một bộ công cụ BI. Phân tích nội dung của IBM và tìm kiếm doanh nghiệp là hai sản phẩm riêng biệt. Các mục tiêu giải pháp hội tụ cả hai tăng cường tìm kiếm doanh nghiệp có sử dụng phân tích văn bản, cũng như phân tích nội dung độc nhu cầu. ICAES có tích hợp chặt chẽ với các nền tảng InfoSphere BigInsights IBM, cho phép các bộ sưu tập tìm kiếm và phân tích nội dung rất lớn.

2.6.4 OpenText

OpenText (www.opentext.com), một công ty trụ sở tại Canada, có lẽ là nổi tiếng nhất với vai trò lãnh đạo của mình trong quản lý thông tin giải pháp doanh nghiệp (EIM). Tầm nhìn của nó xoay quanh việc quản lý, bảo vệ và giải nén giá trị từ các dữ liệu phi cấu trúc của các doanh nghiệp. Nó cung cấp về "ngữ nghĩa trung gian". Theo công ty, phát triển công nghệ ngữ nghĩa của nó được bắt nguồn từ khả năng của mình để cho phép phân tích thời gian thực với độ chính xác cao trên các bộ dữ liệu lớn (nội dung) trên ngôn ngữ, định dạng, và lĩnh vực công nghiệp. Ý tưởng đằng sau trung gian ngữ nghĩa là ngữ nghĩa có thể được tiếp xúc ở các cấp độ khác nhau và làm việc với các công nghệ khác nhau (ví dụ, quản lý tài liệu, phân tích dự đoán, vv) để giải quyết vấn đề kinh doanh. Nói cách khác, các phân tích văn bản có thể được kích hoạt và sử dụng khi cần thiết. OpenText cung cấp trung gian này như là một sản phẩm độc lập được sử dụng trong một loạt các giải pháp cũng như nhúng trong sản phẩm của mình.

2.6.5 SAS

SAS (www.sas.com) đã giải quyết được vấn đề phức tạp của dữ liệu lớn trong một thời gian dài. Vài năm trước đây, họ mua nhà cung cấp phân tích văn bản Teragram để tăng cường chiến lược của mình, sử dụng cả dữ liệu có cấu trúc và phi cấu trúc trong phân tích và tích hợp dữ liệu này cho mô hình mô tả và tiên đoán. Bây giờ, khả năng phân tích văn bản của họ là một phần của phân tích nền tảng và văn bản dữ liệu tổng thể, được xem đơn giản chỉ như là một nguồn dữ liệu.

SAS tiếp tục đổi mới trong lĩnh vực phân tích hiệu suất cao để đảm bảo rằng hiệu suất đáp ứng mong đợi của khách hàng. Mục đích là giải những vấn đề phải mất vài tuần có thể giải quyết chúng trong ngày, hoặc giải quyết vấn đề trong nhiều ngày như trước đây có thể giải quyết trong vài phút. Ví dụ, các máy chủ phân tích SAS hiệu suất cao là một giải pháp trong bộ nhớ cho phép bạn phát triển các mô hình phân tích sử dụng dữ liệu hoàn chỉnh, không chỉ là một tập hợp con của dữ liệu tổng hợp. SAS nói rằng bạn có thể sử dụng hàng ngàn biến và hàng triệu tài liệu như là một phần của phân tích này. Các giải pháp chạy trên EMC Greenplum hoặc các thiết bị Teradata cũng như trên phần cứng hàng hóa sử dụng hệ thống phân phối tập tin Hadoop (HDFS).

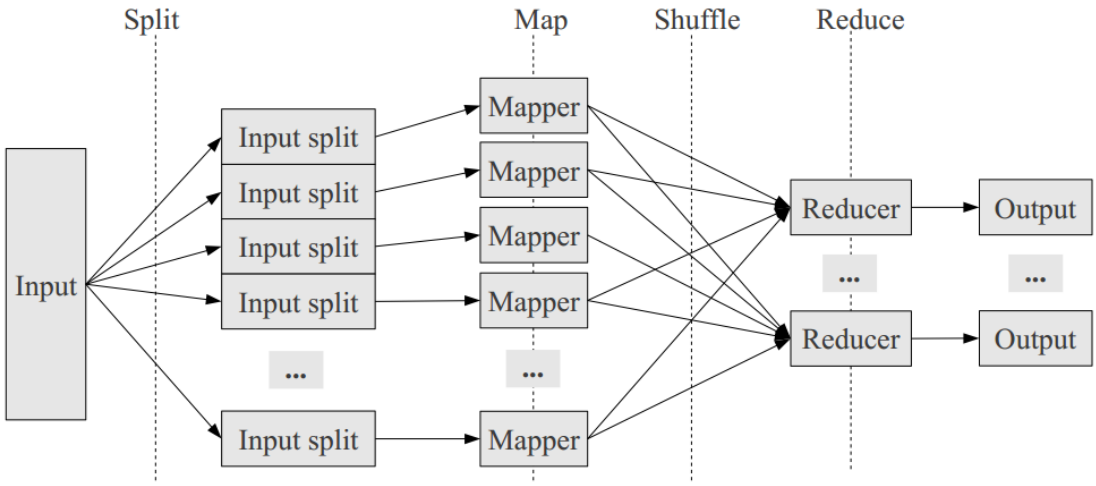
CHƯƠNG 3: HADOOP VÀ THỰC NGHIỆM

3.1 Giới thiệu hệ thống Hadoop

3.1.1 Mô hình xử lý dữ liệu phân tán Mapreduce

3.1.1.1 Giới thiệu chung

Năm 2004, Google công bố mô hình xử lý dữ liệu phân tán MapReduce, Mô hình này là sáng kiến của một nhóm các kỹ sư Google, khi nghiên cứu tìm kiếm giải pháp mở rộng cỗ máy tìm kiếm của họ. Có thể coi MapReduce là một mô hình lập trình, hay một giải thuật lập trình, chuyên dùng để giải quyết vấn đề về xử lý dữ liệu lớn. Mô hình này cơ bản gồm hai thao tác chính là Map và Reduce, với ý tưởng là chia công việc lớn ra thành nhiều công việc nhỏ, giao cho nhiều máy tính cùng thực hiện - thao tác Map, sau đó tổng hợp kết quả lại - thao tác Reduce.



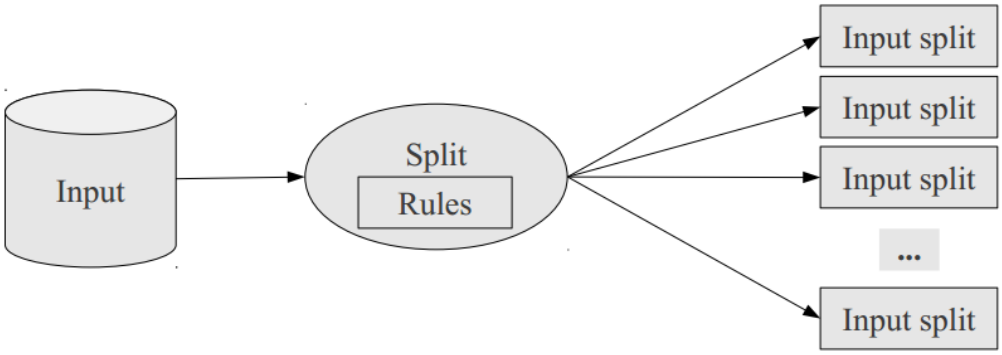
Hình 3-1: Mô hình tổng quát của Mapreduce

Trong mô hình trên, ngoài hai quá trình cơ bản là Map và Reduce đã được trình bày, còn có thêm hai quá trình nữa là Split và Shuffle, hai quá trình này lần lượt giữ vai trò: phân chia dữ liệu đầu vào, tạo tiền đề cho quá trình Map và gom nhóm dữ liệu đầu ra của quá trình Map, tạo tiền đề cho quá trình Reduce.

MapReduce định nghĩa dữ liệu dưới dạng các cặp <key, value> - <khóa, giá trị>; ví dụ, key có thể là tên của tập tin và value nội dung của tập tin, hoặc key là địa chỉ URL và value là nội dung tại URL, v.v. Dữ liệu được định nghĩa theo dạng này linh hoạt hơn các bảng dữ liệu quan hệ hai chiều truyền thống (quan hệ cha - con hay còn gọi là khóa chính - khóa phụ).

3.1.1.2 Quá trình Split

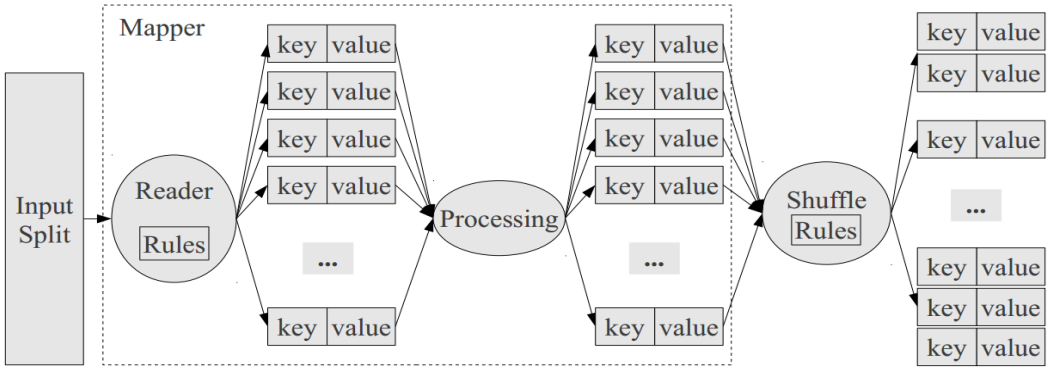
Để có thể phân tán công việc trên hệ thống máy tính, trước tiên cần phải phân nhỏ khối dữ liệu đầu vào cần xử lý ra thành nhiều phần, rồi sau đó mới có thể phân công cho mỗi máy xử lý một phần trong số đó. Quá trình phân chia dữ liệu này được gọi là Split, Split sẽ dựa vào một bộ tiêu chí được đặt ra trước để chia nhỏ dữ liệu, mỗi mảnh dữ liệu được chia nhỏ như vậy gọi là một input split.



Hình 3-2: Quá trình Split

3.1.1.3 Quá trình Map và Shuffle

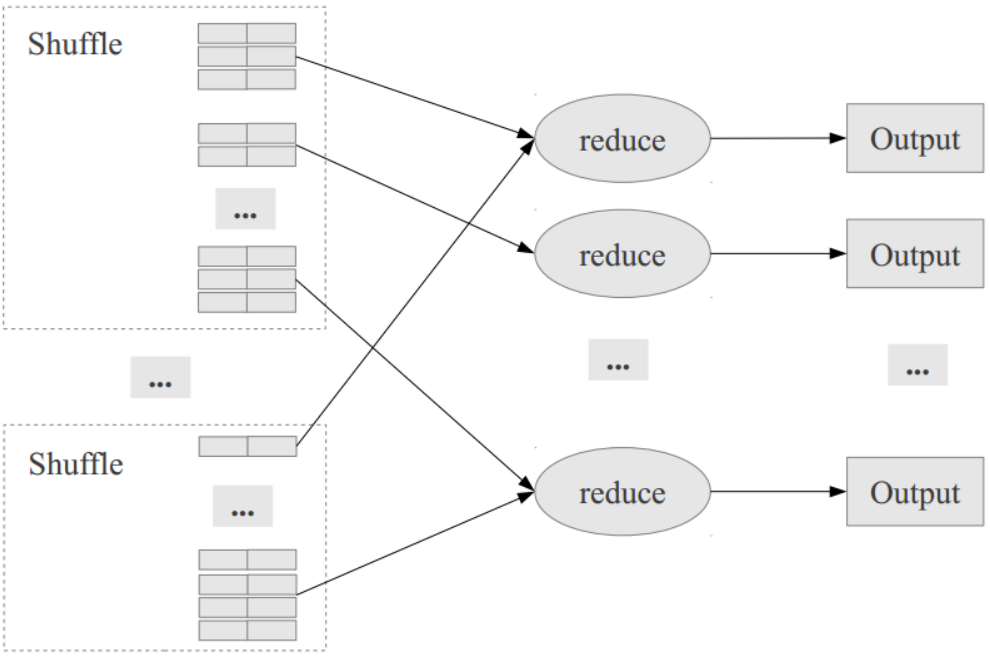
Sau khi các input split được tạo ra, Quá trình Map được thực hiện - hệ thống sẽ phân bố các input split về các máy xử lý, các máy được phân công sẽ tiếp nhận và xử lý input split được giao, ta gọi quá trình diễn ra trên nội bộ mỗi máy trong quá trình Map là Mapper. Trước khi được xử lý, input split được định dạng lại thành dữ liệu chuẩn của MapReduce - dữ liệu có dạng các cặp <key, value>. Kết thúc quá trình Mapper trên mỗi máy, dữ liệu đầu ra cũng có dạng các cặp <key, value>, chúng sẽ được chuyển sang cho quá trình Shuffle để phân nhóm theo tiêu chí đã được định trước, chuẩn bị cho bước xử lý phân tán tiếp theo. Như vậy, quá trình Shuffle sẽ được thực hiện một cách nội bộ trên mỗi máy chạy Mapper.



Hình 3-3: Quá trình Mapper và Shuffle trên một máy

3.1.1.4 Quá trình Reduce

Quá trình Shuffle diễn ra trên nhiều máy nhưng do sử dụng chung một tiêu chí đã được định trước, nên việc phân nhóm dữ liệu trên các máy có sự thống nhất. Các nhóm dữ liệu tương ứng với nhau trên tất cả các máy chạy Shuffle sẽ được gom lại chuyển về cho cùng một máy xử lý, cho ra kết quả cuối cùng. Toàn bộ quá trình này được gọi là *Reduce*, quá trình xử lý trên từng máy là quá trình Reducer.



Hình 3-4: Quá trình Reduce

3.1.1.5 Một số bài toán ứng dụng mô hình Mapreduce

- *Grep* phân tán: *Grep* là một tiện ích dòng lệnh dùng cho việc tìm kiếm trên tập dữ liệu văn bản. Khi áp dụng mô hình MapReduce, trong quá trình Map, mỗi Mapper sẽ làm việc với một tập con của tập dữ liệu văn bản, công việc của mỗi Mapper là tìm kiếm và đánh dấu những dòng khớp với biểu thức tìm kiếm trong tập dữ liệu văn bản mà mình phụ trách. Kết quả của các Mapper sẽ được quá trình Reduce gom lại tạo thành kết quả cuối cùng.
- Sắp xếp phân tán: Mô hình MapReduce rất phù hợp với bài toán sắp xếp dữ liệu. Trong quá trình Map, mỗi Mapper sẽ chỉ giữ nhiệm vụ đọc dữ liệu lên, Shuffle sẽ phân nhóm dữ liệu theo từng khoảng giá trị, Quá trình Reduce sẽ chịu trách nhiệm sắp xếp dữ liệu, mỗi Reducer sẽ sắp xếp dữ liệu trên khoảng giá trị được phân công. Một ví dụ cụ thể về việc sắp xếp dữ liệu: Bài toán sắp xếp một tập dữ liệu nhiệt độ. Thay vì sắp xếp toàn bộ dữ liệu bằng một chương trình tuần tự, ta có thể xử lý

bằng mô hình MapReduce như sau: Tại quá trình Map, dữ liệu sẽ được đọc lên và định dạng thành các cặp <ngày đo, nhiệt độ>. Tại quá trình Shuffle, ta sẽ phân chia dữ liệu theo từng khoảng giá trị của trường "nhiệt độ", trước khi chuyển qua cho Reduce sắp xếp. Như vậy, mỗi Reducer sẽ chỉ sắp xếp những dữ liệu nằm trong một khoảng nhất định. Ta dễ dàng tổng hợp kết quả sắp xếp của các Recducer, để tạo ra kết quả sắp xếp toàn cục.

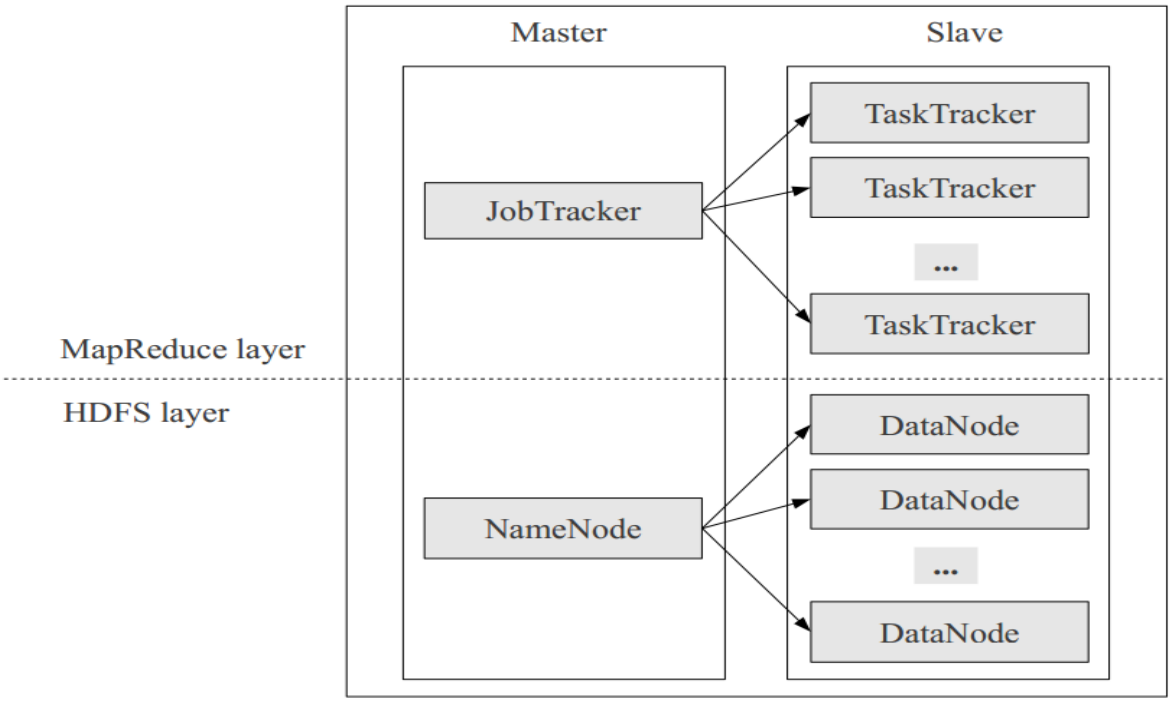
3.1.2 Hadoop – nền tảng lập trình theo mô hình Mapreduce

3.1.2.1 Giới thiệu chung

Hadoop là một nền tảng nguồn mở, được Dough Cutting tạo ra khi ông nghiên cứu về Nutch - một ứng dụng tìm kiếm. Hadoop được viết bằng Java, dùng hỗ trợ xây dựng, thực thi các ứng dụng tính toán phân tán theo mô hình MapReduce. Hadoop cluster là hệ thống máy tính đã được triển khai nền tảng Hadoop, một Hadoop cluster bao gồm hai thành phần cơ bản là kiến trúc MapReduce và hệ thống tập tin phân tán HDFS, Trong đó:

- Kiến trúc MapReducer gồm hai phần: TaskTracker - trực tiếp thực thi các tác vụ xử lý dữ liệu, JobTracker - quản lý và phân chia công việc cho các TaskTracker.
- Hệ thống HDFS gồm hai phần: DataNode - nơi trực tiếp lưu trữ dữ liệu, mỗi DataNode chịu trách nhiệm lưu trữ một phần dữ liệu của hệ thống, NameNode - quản lý các DataNode, dẫn đường cho các yêu cầu truy xuất dữ liệu.

Kiến trúc của Hadoop cluster là kiến trúc Master-Slave, và cả hai thành phần MapReduce và HDFS đều tuân theo kiến trúc này.



Hình 3-5: Các thành phần của Hadoop cluster

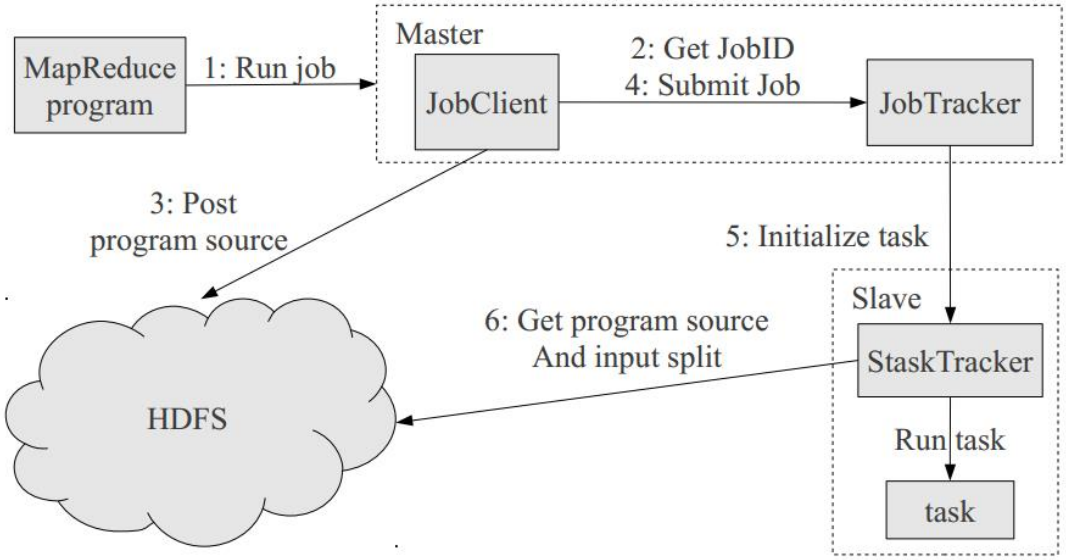
3.1.2.2 Mapreduce Layer

JobTracker và TaskTracker

Trong Hadoop, mỗi quá trình xử lý MapReduce được gọi là một job. Việc thực hiện job sẽ được quản lý bởi hai đối tượng là JobTracker và TaksTracker. JobTracker hoạt động tại máy master có nhiệm vụ quản lý toàn bộ hệ thống gồm việc tạo và quản lý job, phân bố dữ liệu và phân công công việc cho các TaskTracker, xử lý lỗi, v.v.Tại mỗi máy slave có một TaskTracker hoạt động để tạo các task xử lý theo yêu cầu của JobTracker. Ngoài ra, định kỳ mỗi khoảng thời gian, TaskTracker phải gửi tín hiệu HeartBeat về JobTracker để thông báo rằng nó vẫn đang còn hoạt động. Điều này đảm bảo JobTracker lập thời biểu công việc chính xác và hiệu quả cho cả hệ thống.

Cơ chế hoạt động mô hình MapReduce trong Hadoop

Mỗi khi có yêu cầu thực thi một ứng dụng MapReduce, JobTracker sẽ tạo ra một JobClient và chép toàn bộ code thực thi cần thiết của job đó lên hệ thống tập tin phân tán HDFS, mỗi JobClient sẽ được gán một jobID duy nhất. Tiếp theo JobClient sẽ gửi một yêu cầu thực thi job lên JobTracker, JobTracker dựa theo yêu cầu của JobClient, sẽ gửi yêu cầu khởi tạo task kèm theo các thông tin phân công công việc đến các TaskTracker. Mỗi TaskTracker sẽ dựa vào thông tin phân công lần lượt thực hiện: Khởi tạo map task hoặc reduce task, chép toàn bộ code thực thi trên HDFS về, thực hiện công việc được phân công. Sau khi thực hiện xong, TaskTracker sẽ thông báo cho JobTracker và tự giải phóng.



Hình 3-6: Cơ chế hoạt động của JobTracker và TaskTracker trong Hadoop

3.1.2.3 Hadoop Distributed File System Layer

a. Giới thiệu chung

Hadoop Distributed File System (HDFS) là một hệ thống tập tin phân tán, được thiết kế để chạy trên hệ thống nhiều máy tính được nối mạng với nhau, có khả năng chịu lỗi cao và có thể triển khai trên hệ thống phần cứng không đòi hỏi cấu hình đắt tiền. Có rất nhiều đặc điểm giống nhau giữa HDFS và những hệ thống tập tin phân tán khác. Tuy nhiên, HDFS có những đặc điểm nổi bật riêng giúp nó có khả năng hỗ trợ tốt cho các ứng dụng xử lý dữ liệu lớn.

b. Đặc điểm của HDFS

Dữ liệu lưu trữ cực lớn: HDFS được thiết kế để lưu trữ những tập tin với kích thước hàng trăm megabyte, gigabyte hay terabyte. Ngày nay, hệ thống HDFS có thể lưu trữ lên đến petabyte dữ liệu.

Xử lý lỗi phần cứng: HDFS không đòi hỏi phần cứng cấu hình cao đối với hệ thống máy tính. Vì vậy, việc xảy ra lỗi trên các thiết bị phần cứng là hoàn toàn có thể xảy ra một cách thường xuyên. Một hệ thống HDFS có thể bao gồm hàng ngàn máy xử lý, mỗi máy (node) lưu trữ một phần của dữ liệu. HDFS có cơ chế quản lý toàn bộ các node đang chạy trên hệ thống để nhận biết node nào đang rảnh và node nào bị lỗi. Những công việc và dữ liệu được xử lý tại node bị lỗi đó sẽ được chuyển sang node rảnh của hệ thống để xử lý lại.

Dữ liệu chặt chẽ: HDFS hoạt động theo cơ chế ghi một lần - đọc nhiều lần. Mỗi tập tin sẽ được tạo, ghi dữ liệu và đóng lại hoàn toàn. Việc cập nhật ghi thêm dữ liệu vào tập tin là không thể thực hiện trên HDFS. Dữ liệu có thể được truy xuất nhiều lần nhưng vẫn đảm bảo tính nhất quán. Cơ chế thích hợp cho những ứng dụng đọc dữ liệu theo dạng tuần tự.

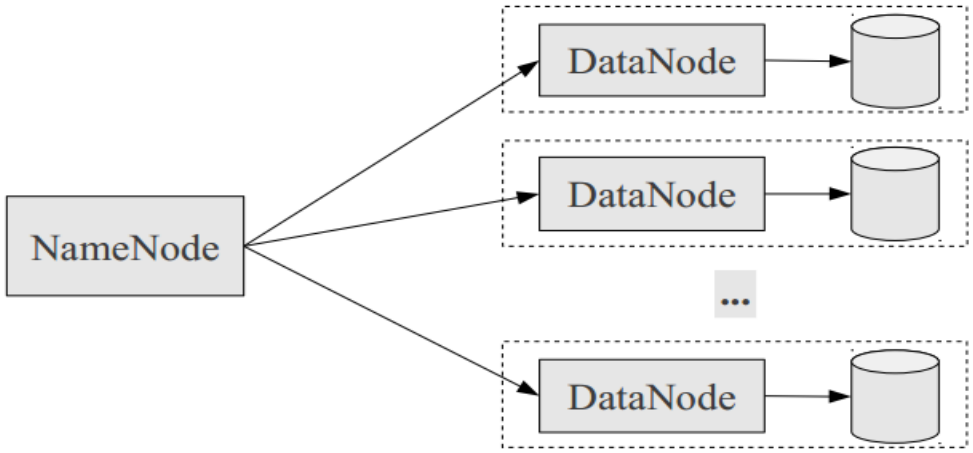
Di chuyển tính toán thay vì di chuyển dữ liệu: những yêu cầu tính toán của ứng dụng sẽ được thực hiện tại node chứa dữ liệu gần nhất với nó. Điều này càng hiệu quả đối với dữ liệu lớn và hệ thống mạng băng thông hẹp. HDFS cung cấp giao diện cho ứng dụng tìm kiếm và di chuyển chính nó đến vị trí dữ liệu gần nhất.

Chạy trên nhiều nền tảng và thiết bị: HDFS được thiết kế để dễ dàng di chuyển từ nền tảng này sang nền tảng khác, thiết bị này sang thiết bị khác. Điều này tạo điều kiện thuận lợi cho việc ứng dụng HDFS một cách rộng rãi.

c. Các khái niệm trên HDFS

Block: Mỗi đĩa cứng có một kích thước block nhất định. Đó là kích thước dữ liệu nhỏ nhất có thể được ghi và đọc trên đó. Kích thước block cho những tập tin hệ thống cho những đĩa lưu trữ đơn thường khoảng vài kilobyte. Việc này giúp người dùng dễ dàng đọc hoặc ghi tập tin với chiều dài đó. HDFS cũng có quy định về kích thước block. Mặc định mỗi block trên HDFS có kích thước là 64MB.

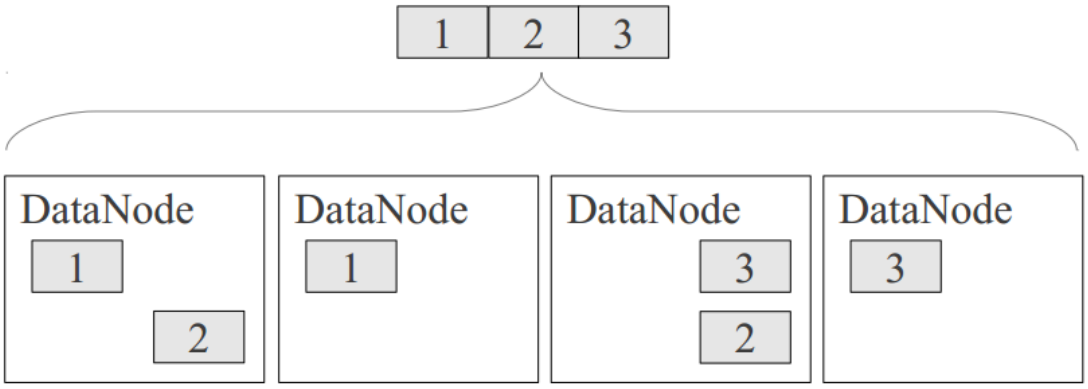
NameNode và DataNode: HDFS là một kiến trúc Master/Slave. HDFS cluster là hệ thống máy tính đã được triển khai HDFS. Trong một cluster HDFS có duy nhất một NameNode đóng vai trò giống như một master, dùng quản lý không gian tên của hệ thống tập tin (file system namespace) và điều phối việc truy xuất dữ liệu. Ngoài ra, còn có các DataNode đóng vai trò như các slave. Mỗi DataNode chịu trách nhiệm quản lý thông tin lưu trữ của các dữ liệu trên máy mà nó đang chạy. HDFS cung cấp một không gian tên cho phép dữ liệu của người dùng lưu trữ trong các tập tin. Mỗi tập tin sẽ được tách thành một hoặc nhiều block được lưu trữ trong một tập hợp những DataNode. Dựa vào không gian trên, NameNode có thể thực hiện các thao tác như đóng, mở và đổi tên tập tin và thư mục. NameNode cũng xác định sơ đồ lưu trữ các block của DataNode. Ngoài nhiệm vụ đáp ứng yêu cầu đọc, ghi dữ liệu do Namenode chuyển đến, các DataNode cũng có nhiệm vụ tạo, xóa và nhân rộng các blocks theo những chỉ thị từ NameNode.



Hình 3-7: Kiến trúc Master/Slave của hệ thống tập tin phân tán Hadoop

File System Namespace: HDFS tổ chức tập tin phân cấp theo mô hình truyền thống. Với HDFS người dùng cũng có thể tạo, xóa, di chuyển, đổi tên tập tin thư mục như các hệ thống tập tin truyền thống thông thường.

Data Replication: HDFS được thiết kế để lưu trữ các tập tin cực lớn. Trên một Hadoop cluster mỗi tập tin được chia thành nhiều block có thứ tự và lưu trữ trên nhiều máy. Việc nhân bản các block nhằm tăng khả năng chịu lỗi cho hệ thống. Mỗi block sẽ được nhân bản bao nhiêu lần tùy theo cấu hình của hệ thống. Các DataNode có nhiệm vụ lưu trữ các block mà nó được NameNode phân công.



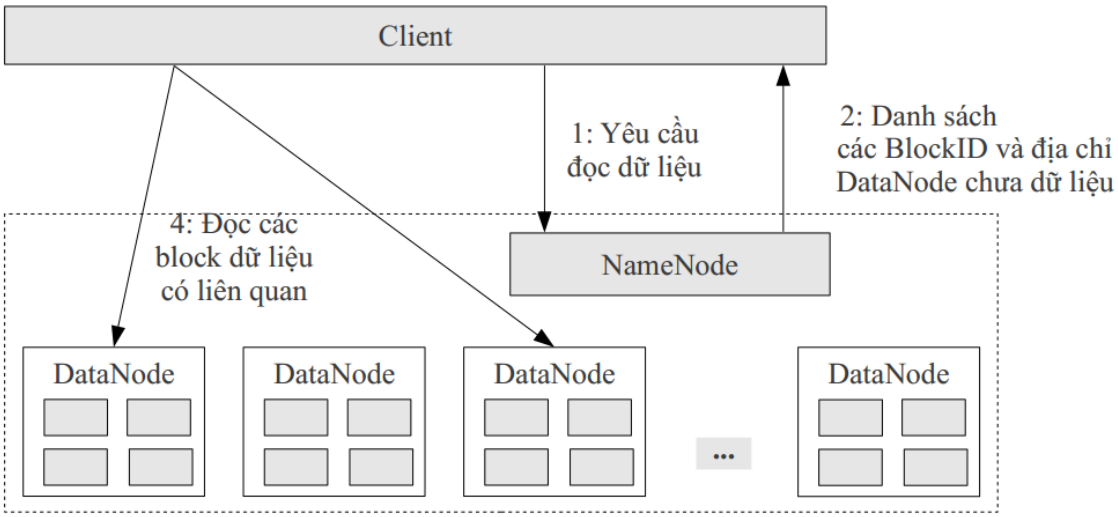
Hình 3-8: Nhân bản block trong HDFS

d. Lỗi đĩa, thông điệp HeartBeat và nhân bản tại các block

Định kỳ, mỗi DataNode sẽ gởi đến NameNode một thông điệp gọi là HeartBeats để xác định tình trạng hoạt động của DataNode. Nếu sau một khoảng thời gian quy định mà không thấy DataNode gởi HeartBeats, NameNode sẽ đánh dấu là DataNode đó bị lỗi và không ra bất kỳ thao tác nào cho nó nữa. Bất kỳ dữ liệu gì do DataNode bị lỗi quản lý đều xem như không còn nữa. NameNode sẽ tìm một bản sao khác của các block trên DataNode bị lỗi và sao chép toàn bộ công việc của DataNode bị lỗi cùng bản sao của các block cho DataNode nào còn rảnh để thực hiện tiếp công việc. Khi số lượng bản sao của một block nhỏ hơn giá trị được chỉ định trước, NameNode sẽ khởi tạo quá trình nhân bản bất cứ khi nào có thể.

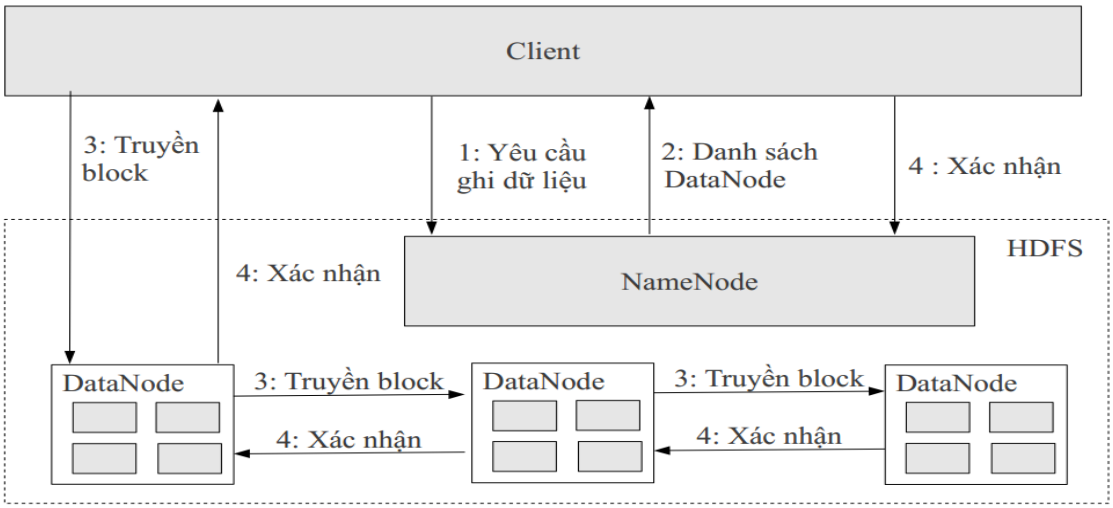
e. Truy xuất dữ liệu trên HDFS

Đọc dữ liệu trên HDFS: Khi chương trình client gởi một yêu cầu đọc dữ liệu tới HDFS, hệ thống sẽ gởi một yêu cầu đến NameNode để hỏi về vị trí các block có liên quan đến dữ liện cần đọc. Sau khi có được danh sách vị trí các block có liên quan, Client sẽ kết nối trực tiếp đến DataNode để đọc dữ liệu.



Hình 3-9: Quá trình đọc dữ liệu trên HDFS

Ghi dữ liệu trên HDFS: Khi chương trình client có tập tin dữ liệu cần ghi lên HDFS, đầu tiên tập tin dữ liệu phải được ghi vào bộ nhớ cục bộ của máy tính chạy chương trình client. Khi tập tin cục bộ đã tích lũy đủ dung lượng của một block, hoặc quá trình ghi tập tin cục bộ đã hoàn toàn kết thúc, chương trình client sẽ nhận về một danh sách những DataNode được NameNode chỉ định sẽ chứa tập tin dữ liệu này. Sau đó, chương trình client sẽ gởi bản sao của tập tin cần ghi đến DataNode đầu tiên. DataNode đầu tiên sẽ trích lấy một phần của tập tin để ghi xuống bộ nhớ của mình, sau đó chuyển toàn bộ phần còn lại cho DataNode khác, DataNode khác sẽ thực hiện lại công việc mà DataNode đầu tiên đã làm, cứ tiếp tục như vậy cho đến khi toàn bộ nội dung của tập tin được ghi hết. Có thể nói, việc ghi dữ liệu trên hệ thống HDFS được thực hiện theo cơ chế ống dẫn.



Hình 3-10: Quy trình ghi dữ liệu trên HDFS theo cơ chế ống dẫn

f. SecondaryNameNode

Trong HDFS cluster có duy nhất một NameNode, hệ thống không thể hoạt động được nếu như không có NameNode. Vì tính chất quan trọng đó, việc sao lưu dự phòng cho NameNode là rất cần thiết. Đó chính là nhiệm vụ của Secondary NameNode. Định kỳ sau mỗi khoảng thời gian, Secondary NameNode sẽ kết nối đến NameNode để cập nhật các tập tin về cấu hình, trạng thái của hệ thống, toàn bộ các tập tin này sẽ được lưu lại thành một CheckPoint, dùng cho việc khôi phục lại NameNode nếu NameNode bị lỗi. Đồng thời, NameNode là một dịch vụ phải phục vụ rất nhiều DataNode trong hệ thống. Secondary NameNode còn có khả năng chia tải với NameNode. Ngoài ra, việc định kỳ tạo CheckPoint giúp hệ thống chuyển sang trạng thái hoạt động nhanh hơn.

3.1.2.4 Kiểm tra tình trạng hoạt động của hệ thống Hadoop

Ta có thể kiểm tra tình trạng hoạt động của các MapReduce job thông qua trình duyệt web, với địa chỉ http:<JobTracker>:50030, giá trị <JobTracker> chính là IP hoặc tên miền của máy đang chạy JobTrakcer, tức máy master của hệ thống. Nhờ đó, ta có thể biết được MapReduce job đã hoàn tất, bị lỗi hay đang chạy, chạy được bao nhiêu phần, node nào đang chạy các TaskTracker, bao nhiêu tác vụ Reducer, bao nhiêu tác vụ Mapper đang chạy trên các node, số phần trăm kết quả đạt được của các tác vụ, mỗi máy đã xử lý bao nhiêu input split, v.v. Tương tự với HDFS, ta có thể sử dụng trình duyệt web, truy cập tới địa chỉ http:<NameNode>:50070, với <NameNode> là địa chỉ IP, hoặc tên miền của máy chạy NameNode, để biết được các thông tin về trạng thái hoạt động của hệ thống.

3.1.3 Xây dựng một chương trình chạy trên nền Hadoop

3.1.3.1 Các lớp cơ bản trong chương trình Hadoop

a. Các kiểu dữ liệu cơ bản

Để xây dựng chương trình, Hadoop xây dựng gói org.apache.hadoop.io hỗ trợ các kiểu dữ liệu phù hợp với Hadoop cho Java, gồm có:

- NullWritale: tương ứng với kiểu dữ liệu Null trong Java.
- Text: tương ứng với kiểu dữ liệu String trong Java.
- BytesWritable: tương ứng với kiểu Byte trong Java.
- BooleanWritable: tương ứng với kiểu Boolean trong Java.
- IntWritable: tương ứng với kiểu Integer trong Java.
- LongWritable: tương ứng với kiểu Long trong Java.

- FloatWritable: tương ứng với kiểu Float trong Java.
- DoubleWritable: tương ứng với kiểu Double trong Java

b. Lớp Mapper

Đây là lớp hỗ trợ thực hiện quá trình Map trong hệ thống. Lập trình viên sẽ viết một lớp mới, thừa kế lại lớp Mapper. Có thể định nghĩa lại các phương thức trong lớp Mapper cho phù hợp, có hai phương thức qua trọng cần phải quan tâm là:

- Phương thức run(): Lập trình viên có thể định nghĩa lại phương thức này để kiểm soát việc đọc và phân phát dữ liệu từ input split.
- Phương thức map(): Đây là phương thức quan trọng nhất, trong hầu hết các trường hợp Lập trình viên phải định nghĩa lại phương thức này, phương thức này được thiết kế để mỗi lần nhận vào và xử lý một cặp <key, value>.

c. Lớp Partitioner

Sử dụng lớp Partitioner giúp chúng ta có thể tùy biến, phân nhóm các cặp <key, value> đầu ra của quá trình Mapper trên mỗi map task. Nếu không sử dụng lớp này trong chương trình MapReduce, dữ liệu đầu ra của quá trình Mapper sẽ được gom lại thành một nhóm duy nhất.

d. Lớp hỗ trợ Combiner

Combiner có thể được hoặc không được sử dụng trong chương trình MapReduce, mục đích của tác vụ này là giảm lượng dữ liệu gởi đi từ các map task tới các reduce task. Bản chất của tác vụ Combiner là thực hiện tác vụ Reducer tại từng map task trước khi gửi đi thực hiện Reducer một lần nữa tại reduce task. Mỗi map task sẽ thực hiện một hoặc nhiều tác vụ Combiner, mỗi Combiner sẽ phụ trách xử lý một nhóm dữ liệu đầu ra của Mapper. Việc xây dựng lớp Combiner tương tự như xây dựng lớp Reducer.

e. Lớp Reduce

Lớp Reducer hỗ trợ thực hiện quá trình Reduce. Tương tự như lớp Mapper, Lập trình viên sẽ thiết kế lớp mới thừa kế lại lớp Reducer, và định nghĩa lại các phương thức có sẵn nếu cần thiết, hai phương thức thường được định nghĩa lại là:

- Phương thức run(): Lập trình viên có thể định nghĩa lại phương thức này để kiểm soát việc đọc và phân phát dữ liệu từ quá trình Map gửi tới.

- Phương thức reduce(): Như phương thức map() trong lớp Mapper, Lập trình viên thường phải định nghĩa lại phương thức này, phương thức này được thiết kế để mỗi lần nhận vào và xử lý một loạt các cặp <key, value> có cùng chung thuộc tính key.

f. Lớp WritableComparator

Dữ liệu được tạo ra từ tác vụ Map sau khi được phân nhóm, Combine và lưu trữ vào bộ nhớ cục bộ của các máy chạy map task, sẽ được các reduce task chép về bộ nhớ cục bộ của mình, mỗi reduce task chỉ chép về những dữ liệu thuộc nhóm được phân công xử lý. Tại đây, dữ liệu trước khi được xử lý tại phương thức reduce() sẽ được gom nhóm lại một lần nữa theo thuộc tính key và tổ chức sắp xếp trong từng nhóm nếu có yêu cầu. Lớp WritableComparator cho phép chúng ta định nghĩa lại hàm compare() tạo ra tiêu chí sắp xếp cho các cặp <key, value>. Nếu không khai báo và sử dụng lớp này thì mặc định các cặp <key, value> sau khi được gom nhóm sẽ không được sắp xếp theo bất kỳ tiêu chí nào.

3.1.3.2 Quy trình hoạt động

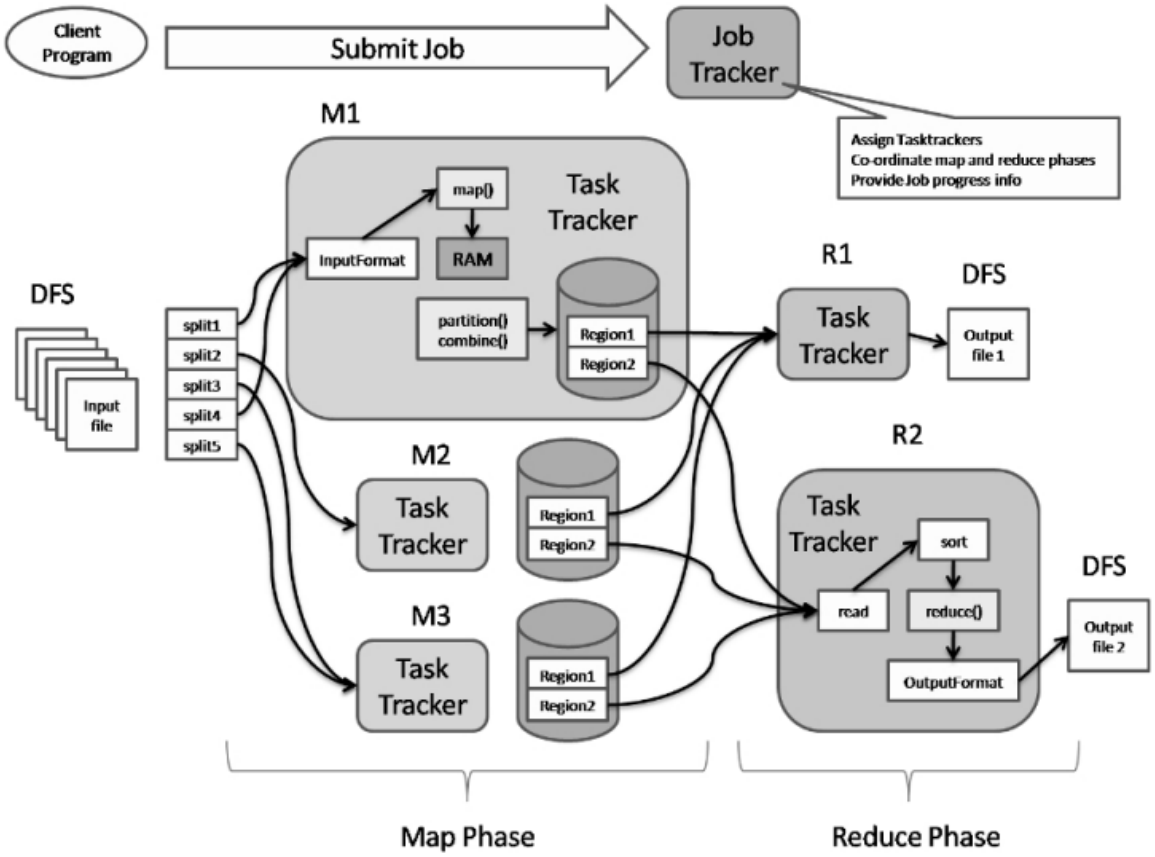
Khi Hadoop cluster được nạp một chương trình - một job, JobTracker sẽ thực hiện việc khởi tạo một job mới trên hệ thống. Nó sẽ đọc số lượng input file mà chương trình cần thực thi, thực hiện việc chia thành các input split. Tùy theo số lượng input split, JobTracker sẽ yêu cầu các TaskTracker khởi tạo đủ số lượng map task cần thiết cho việc xử lý.

Thực thi tại Map Task

Mỗi map task sẽ đọc vào một input split và phân nó thành những record trong hàm run(), mỗi record là một cặp <key, value>. Sau đó, phương thức map() được gọi để thực hiện việc tính toán xử lý trên từng cặp <key, value>. Kết quả sau khi được xử lý sẽ không được chuyển ngay đến reduce task mà được lưu trữ tại bộ nhớ cục bộ của map task. Khi kích thước dữ liệu đạt đến ngưỡng quy định, map task thực hiện quá trình Shuffle để phân nhóm dữ liệu. Nếu trong chương trình có thiết lập sử dụng lớp Combine, thì map task sẽ thực hiện việc Combiner cho từng nhóm dữ liệu. Kết quả sau khi thực hiện sẽ được ghi vào một tập tin tràn và đăng ký với TaskTracker. Khi kích thước tập tin đủ lớn sẽ thực hiện việc chuyển dữ liệu sang reduce task.

Thực thi tại Reduce Task

Đầu tiên reduce task sẽ chép dữ liệu từ các map task về bộ nhớ cục bộ của nó. Mỗi reduce task chỉ thực hiện việc chép những dữ liệu thuộc một nhóm nhất định. Tiếp theo, dữ liệu sẽ được gom nhóm theo key, mỗi nhóm có dạng <key, list(values)>, nếu được yêu cầu sắp xếp, dữ liệu trong mỗi nhóm sẽ được sắp xếp trước khi gửi qua phương thức reduce() để xử lý và ghi dữ liệu ra HDFS.



Hình 3-11: Quá trình hoạt động của một tác vụ MapReduce trên Hadoop

3.2 Thực nghiệm

3.2.1 Hướng dẫn cài đặt Hadoop cluster

Những phiên bản phần mềm, hệ điều hành dùng trong cài đặt:

- Hệ điều hành Ubuntu-16.04-1-desktop-amd64
- Java
- SSH cài sẵn trên hệ điều hành
- Hadoop phiên bản 2.2.0

Những cài đặt và thiết lập chung trên hệ thống (được thực hiện trong Terminal)

3.2.1.1 Cài đặt Java

Hadoop yêu cầu các máy tính trên cluster phải được cài Java với phiên bản thấp nhất là 1.5, và phiên bản khuyên dùng là 1.6. Do đó, để hệ thống hoạt động một cách tốt nhất, phiên bản Java 1.6 sẽ được hướng dẫn cài đặt. Ngoài ra, trong hướng dẫn sau, việc cài đặt Java sẽ được thực hiện thông qua một nhà phân phối, nếu thấy đó là không an toàn, bạn có thể tải và cài đặt Java theo hướng dẫn tại trang chủ của Oracle.

- Kiểm tra java đã được cài đặt trên ubuntu bằng câu lệnh: `$ java version`
- Nếu máy chưa được cài đặt java, ta sẽ cài đặt bằng câu lệnh: `$ sudo apt-get install openjdk-9-jdk`
- Cài đặt ssh: `$ sudo apt-get install openssh-server`

3.2.1.2 Tạo người dùng riêng cho Hadoop

- Tạo nhóm người dùng hadoop: `$ sudo addgroup hadoop`
- Thêm người dùng hduser thuộc nhóm người dùng hadoop: `$ sudo adduser -- ingroup hadoop hduser`
- Đăng nhập vào tài khoản người dùng hduser: `$ su -- hduser`

3.2.1.3 Cấu hình ssh

Tạo ra khóa chứng thực SSH cho hduser:

```
$ ssh-keygen -t rsa -P ""
```

```
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_key
```

Kiểm tra bằng lệnh: `$ ssh localhost`

3.2.1.4 Cài đặt và cấu hình Hadoop

- Giải nén gói Hadoop: `$ sudo tar vxzf hadoop-2.2.0.tar.gz`
- Di chuyển thư mục cài đặt hadoop về thư mục /usr/local: `$ sudo mv hadoop-2.2.0 /usr/local/hadoop`
- Chuyển quyền sử dụng cho người dùng hduser: `$ sudo chown -R hduser:hadoop hadoop`
- Chèn nội dung sau vào cuối tập tin /home/hduser/.bashrc để thêm các thiết lập đường dẫn cho người dùng hduser:

```
# Java Path
export JAVA_HOME = /usr/lib/jvm/java-9-openjdk
# Hadoop Variables
export HADOOP_HOME =  /usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR =$HADOOP_HOME/etc/hadoop
```

- Chèn nội dung sau vào cuối tập tin hadoop/conf/hadoop-env.sh để thêm thiết lập đường dẫn Java cho Hadoop
  export JAVA_HOME = /usr/lib/jvm/java-9-openjdk

- Thêm vào giữa 2 thẻ <configuration> và </configuration> trong tập tin hadoop/etc/hadoop/core-site.xml

```
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
```

- Thêm vào giữa 2 thẻ <configuration> và </configuration> trong tập tin hadoop/etc/hadoop/hdfs-site.xml

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</val
ue>
</property>
<property>
<name>dfs.namenode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</val
ue>
</property>
```

- Thêm vào giữa 2 thẻ <configuration> và </configuration> trong tập tin hadoop/etc/hadoop/yarn-site.xml

```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-
services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.shuffleHandler</valu
e>
</property>
```

- Thêm vào giữa 2 thẻ <configuration> và </configuration> trong tập tin hadoop/etc/hadoop/mapred-site.xml

```
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

- Tạo thư mục chứa namenode và datanode

```
$ sudo mkdir -p /usr/local/hadoop_tmp
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

- Chuyển quyền cho người dung hduser

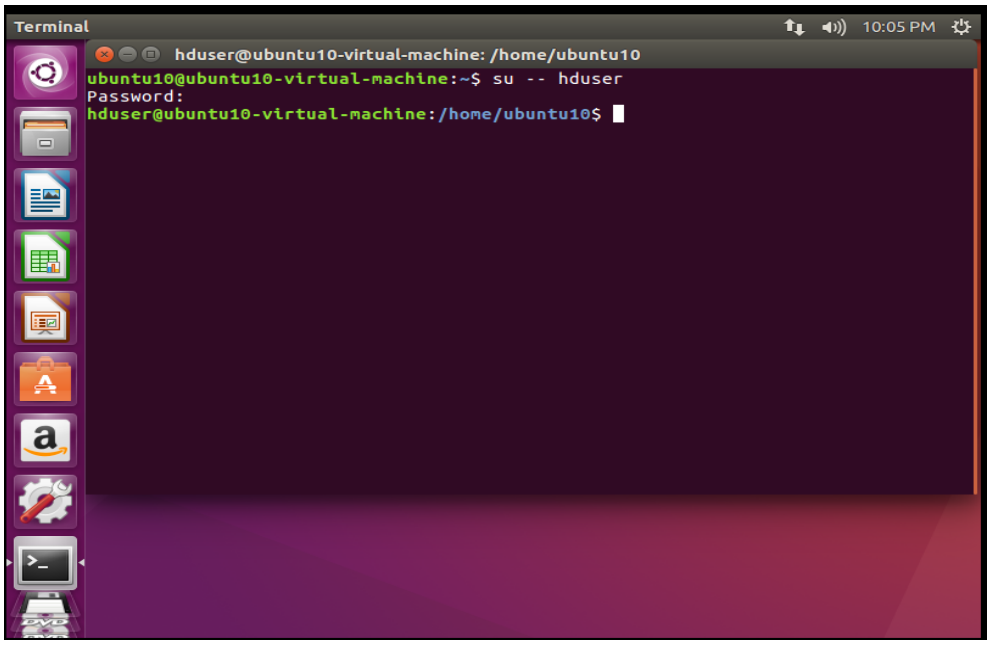
```
$ sudo chown –R hduser /usr/local/hadoop_tmp
```

- Định dạng namenode

```
$ hadoop namenode –format
```

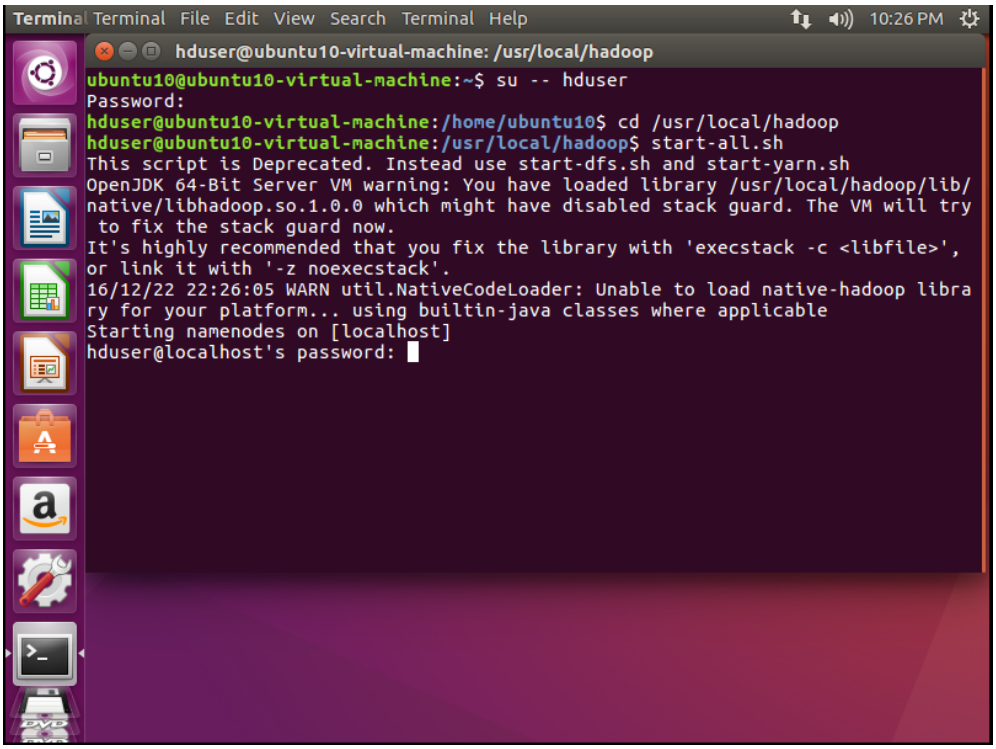
3.2.2 Khởi động hệ thống

Đăng nhập vào tài khoản người dùng hduser: `$ su -- hduser`



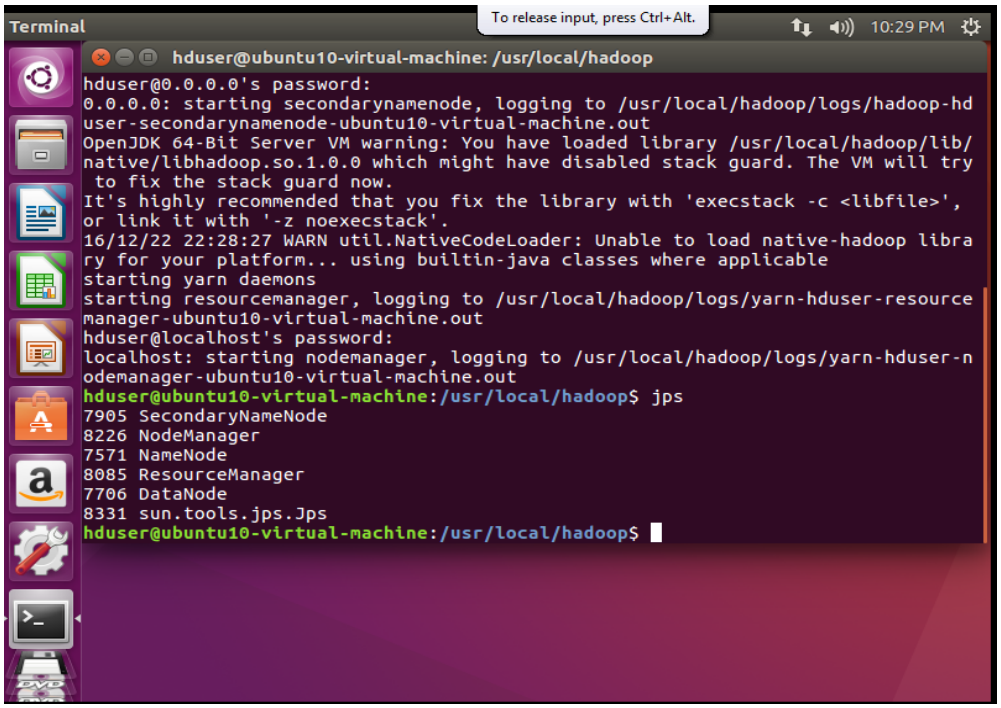
Hình 3-12: Đăng nhập vào tài khoản người dùng hduser

Start Hadoop: `$ start-all.sh`



Hình 3-13: Khởi động Hadoop

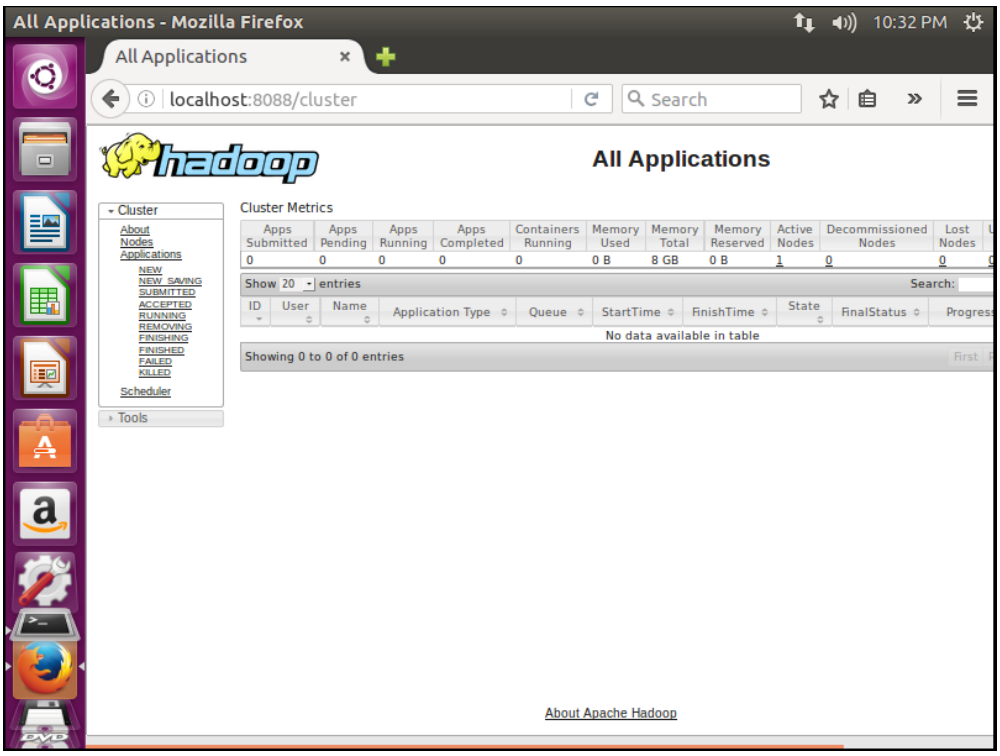
Kiểm tra Hadoop đã được chạy thành công: `$ jps`



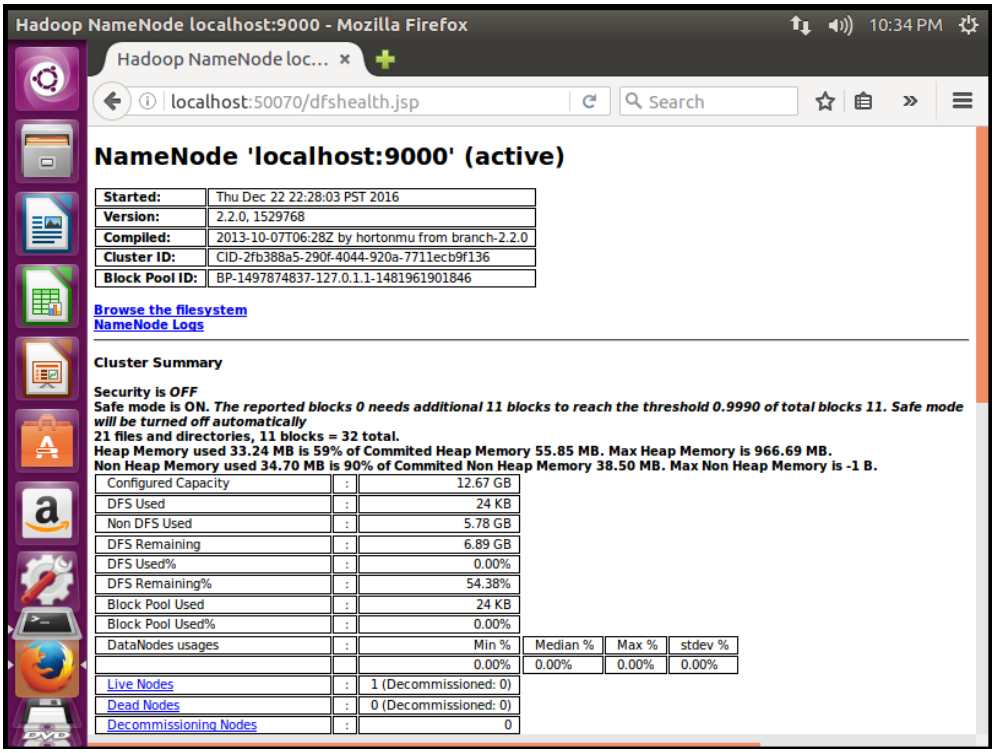
Hình 3-14: Kiểm tra Hadoop

Các trang quản lý của Hadoop:

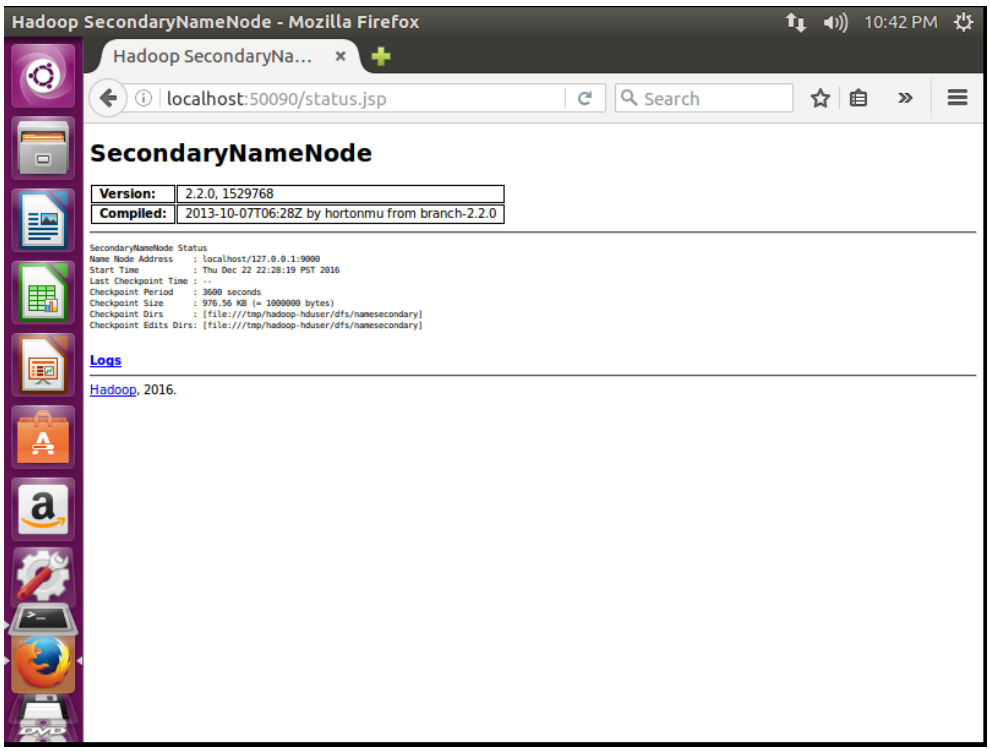
- All Aplications: localhost:8088
- Hadoop Namenode: localhost:50070
- Secondary Namenode: localhost:50090
- Content of directory: localhost:50075



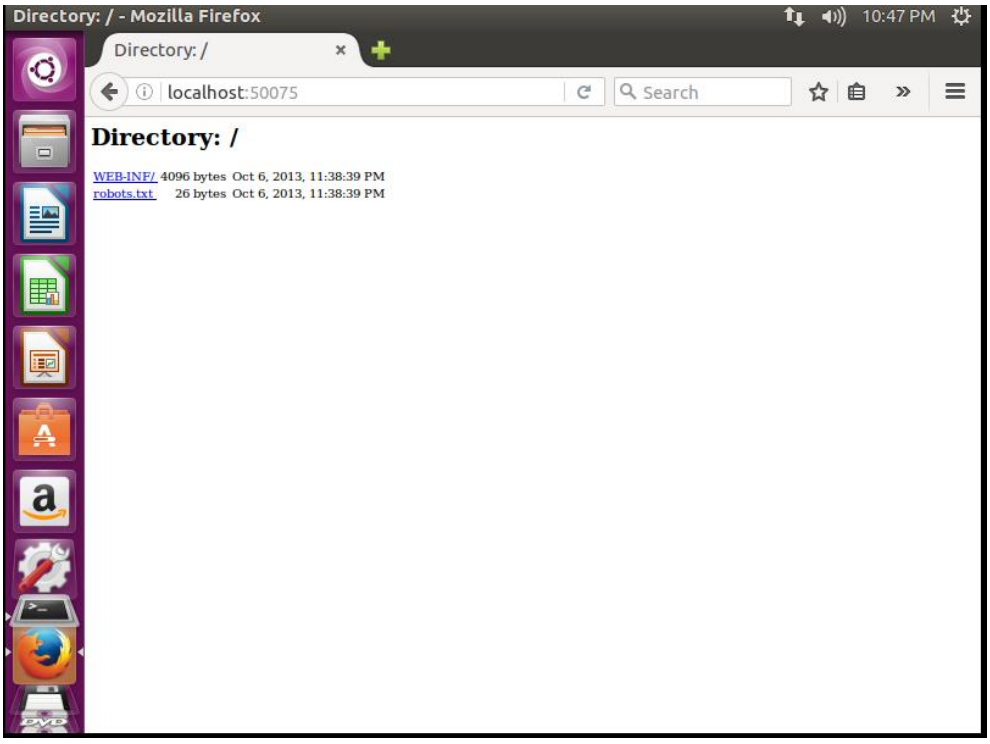
Hình 3-15: Trang quản lý Hadoop All Aplications



Hình 3-16: Trang quản lý Hadoop Namenode

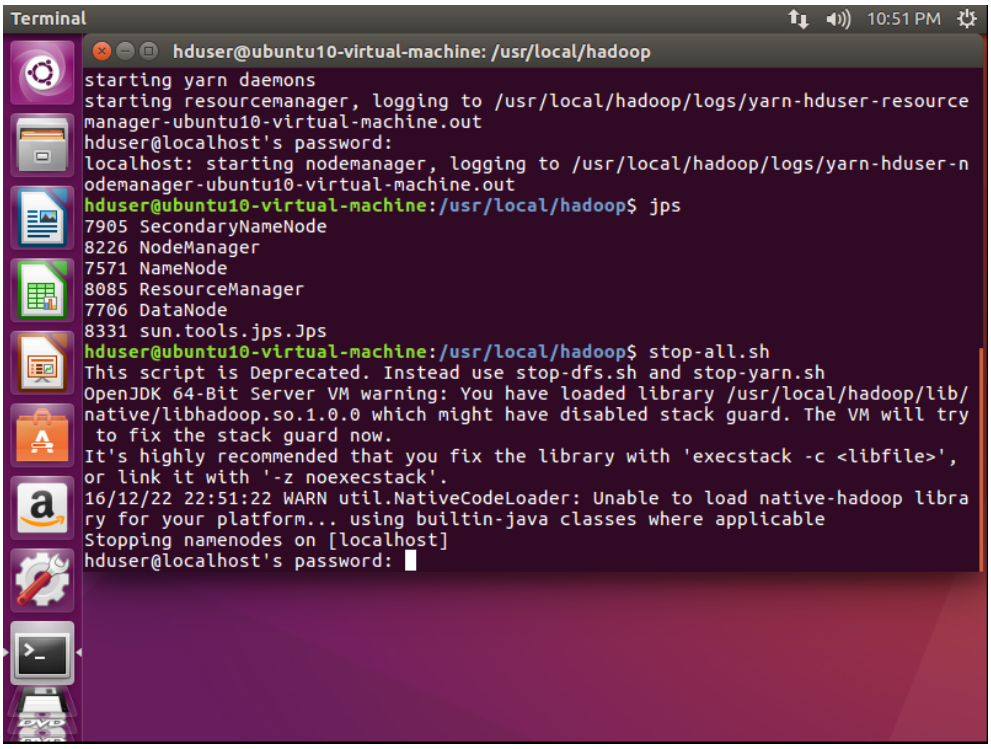


Hình 3-17: Trang quản lý Hadoop SecondaryNamenode



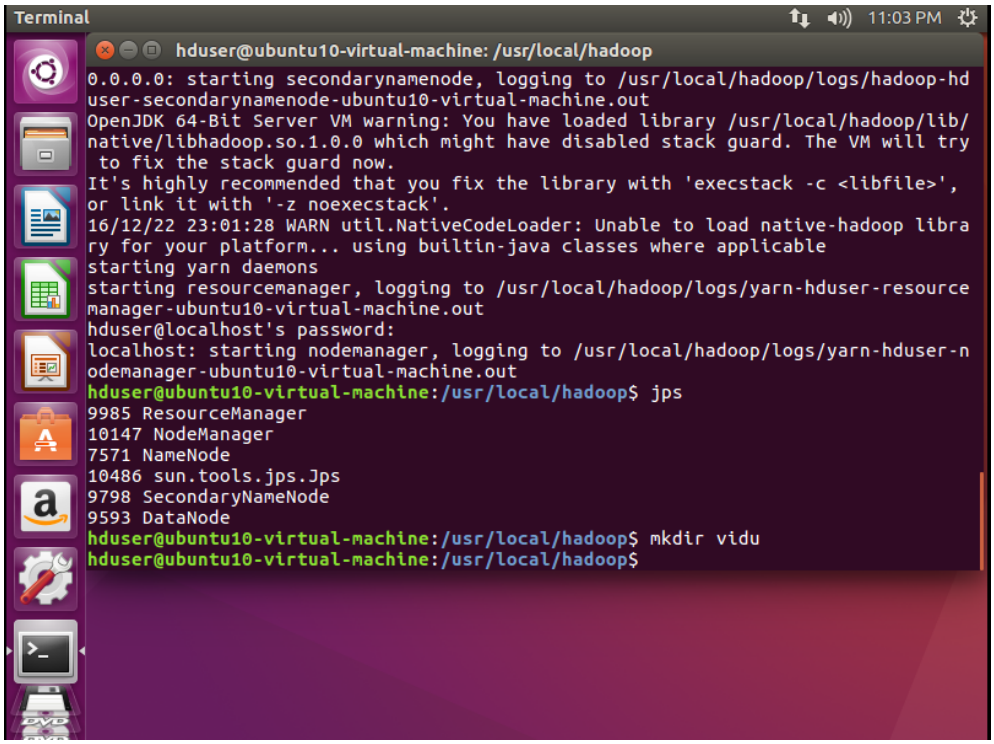
Hình 3-18: Trang quản lý Hadoop Directory

Stop Hadoop:`$ stop-all.sh`



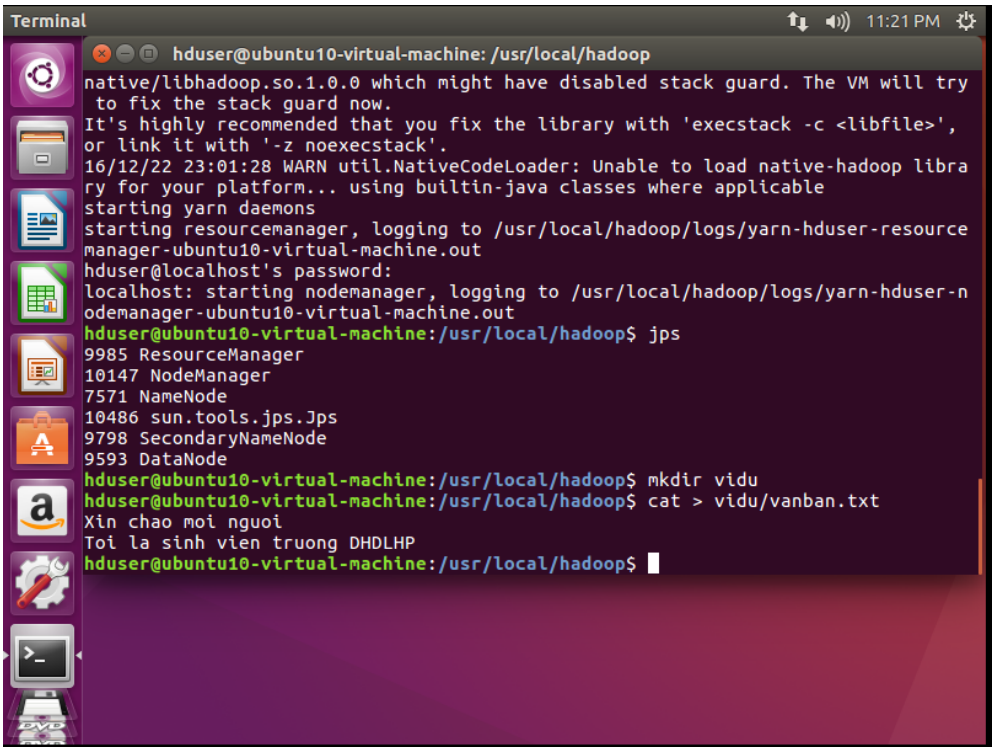
Hình 3-19: Tắt Hadoop

Tạo 1 thư mục tên là vidu: `$ mkdir vidu`

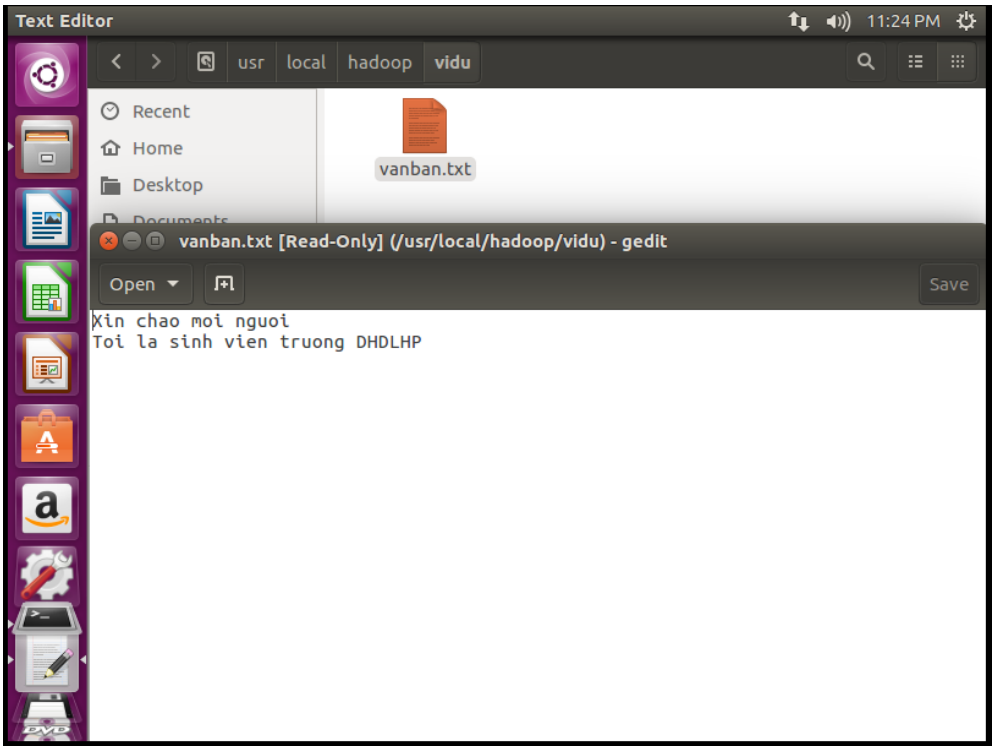


Hình 3-20: Tạo thư mục vidu

Thêm 1 file văn bản có tên vanban.txt vào trong thư mục vidu và nhập nội dung văn bản: $ cat > vidu/vanban.txt

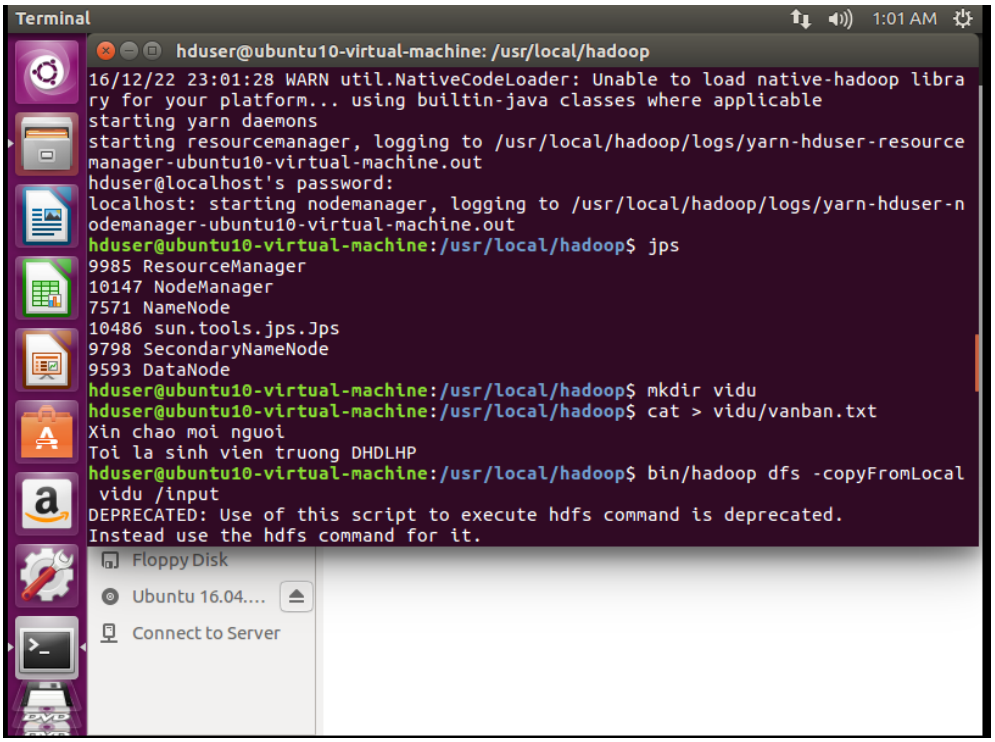


Hình 3-21: Thêm file văn bản vào trong thư mục vidu



Hình 3-22: Thư mục vidu và file vanban.txt được tạo

Copy thư mục vidu vào hdfs: `$ bin/hadoop dfs –copyFromLocal vidu /input`



Hình 3-23: Copy thư mục vidu vào hdfs

KẾT LUẬN

Trong quá trình nghiên cứu, tìm hiểu và hoàn thành đề tài đồ án tốt nghiệp "Áp dụng các kỹ thuật trong Big data vào lưu trữ dữ liệu", em đã có cơ hội để thu nhận được thêm rất nhiều kiến thức về Big data. Big data là một đề tài rộng lớn có tác động mạnh mẽ đến mọi lĩnh vực trong đời sống xã hội. Trong thời gian tới chắc chắn Big data sẽ ngày càng thể hiện được sức mạnh và tầm ảnh hưởng.

Trong đề tài đồ án tốt nghiệp này em đã trình bày về công nghệ nền tảng quản lý dữ liệu lớn, mô hình phân tán dữ liệu Mapreduce và mô hình Hadoop. Do thời gian thực hiện đề tài đồ án hạn chế nên vẫn còn rất nhiều những kiến thức khác liên quan đến Big data mà em chưa tìm hiểu và khai thác. Trong thời gian tới em sẽ cố gắng tiếp tục nghiên cứu và tìm hiểu sâu hơn về lĩnh vực này.

Sinh viên

Nguyễn Chí Thanh

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1]. Nguyễn Minh Thuận, Nguyễn Trọng Thức - *Nghiên cứu nền tảng tính toán song song với MapReduce và Hadoop. Áp dụng cho việc xây dựng wordnet tiếng Việt và tạo chỉ mục tài liệu,* Cần Thơ, 2012.

[2]. Bùi Thị Hồng Phúc - *Xây dựng một ứng dụng minh họa cho khả năng của MongdoDB* , Cần Thơ, 2012.

Tài liệu Tiếng Anh

[3]. Tom White - *Hadoop The Definitive Guide 3rd Edition*.

[4]. Judith S. Hurwitz, Alan F. Nugent, Dr.Fern Halper, Marcia A. Kaufman - *Big data for dummies*

[5]. Marcello Trovati, Richard Hill, Ashiq Anjum, Shao Ying Zhu, Lu Liu - *Big data Analytics and Cloud Computing*

Tài liệu trực tuyến

[6]. Website Hadoop: *https://hadoop.apache.org/*

[7]. *https://dinhnn.com/category/big-data/*