BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ

Địa chỉ:
- **Số 1:** Số 24, Thái Học 2, phường Sao Đỏ, thành phố Chí Linh, tỉnh Hải Dương.
- **Số 2:** Số 72, đường Nguyễn Thái Học/Quốc lộ 37, phường Thái Học, thành phố Chí Linh, tỉnh Hải Dương.
- **Điện thoại:** (0220) 3882 269 **Fax:** (0220) 3882 921 **Website:** http://saodo.edu.vn **Email:** info@saodo.edu.vn

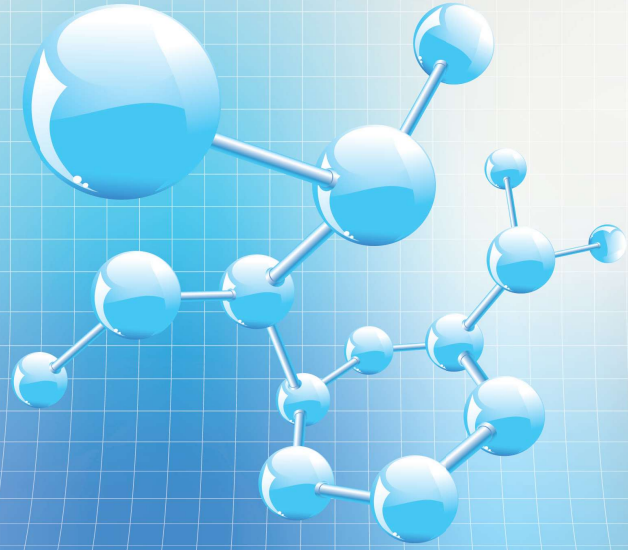
P. ISSN 1859-4190
E. ISSN 2815-553X

Số 3 (78)
2022

SỐ 3 (78) 2022
TẠP CHÍ NGHIÊN CỨU KHOA HỌC
P.ISSN 1859-4190 - E.ISSN 2815-553X

Tạp chí
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ
SCIENTIFIC JOURNAL - SAO DO UNIVERSITY

P. ISSN 1859-4190
E. ISSN 2815-553X



Địa chỉ Tòa soạn:
Trường Đại học Sao Đỏ.
Số 24, Thái Học 2, phường Sao Đỏ, thành phố Chí Linh, tỉnh Hải Dương.
Điện thoại: (0220) 3587213, Fax: (0220) 3882 921, Hotline: 0912 107858/0936 847980.
Website: http://tapchikhcn.saodo.edu.vn/Email: tapchikhcn@saodo.edu.vn.

Giấy phép xuất bản số: 620/GP-BTTTT ngày 17/9/2021 của Bộ Thông tin và Truyền thông.
In 2.000 bản, khổ 21 × 29,7cm, tại Công ty TNHH in Tre Xanh, cấp ngày 17/02/2011.

Số 3 (78)
2022

P. ISSN 1859-4190
E. ISSN 2815-553X

Địa chỉ Tòa soạn:
Trường Đại học Sao Đỏ.
Số 24, Thái Học 2, phường Sao Đỏ, thành phố Chí Linh, tỉnh Hải Dương.
Điện thoại: (0220) 3587213, Fax: (0220) 3882 921, Hotline: 0912 107858/0936 847980.
Website: http://tapchikhcn.saodo.edu.vn/Email: tapchikhcn@saodo.edu.vn.

Giấy phép xuất bản số: 620/GP-BTTTT ngày 17/9/2021 của Bộ Thông tin và Truyền thông.
In 2.000 bản, khổ 21 × 29,7cm, tại Công ty TNHH in Tre Xanh, cấp ngày 17/02/2011.

THỂ LỆ GỬI BÀI
TẠP CHÍ NGHIÊN CỨU KHOA HỌC, TRƯỜNG ĐẠI HỌC SAO ĐỎ

Tạp chí Nghiên cứu khoa học, Trường Đại học Sao Đỏ (P. ISSN 1859-4190, E. ISSN 2815-553X), thường xuyên công bố kết quả, công trình nghiên cứu khoa học và công nghệ của các nhà khoa học, cán bộ, giảng viên, nghiên cứu sinh, học viên cao học, sinh viên ở trong và ngoài nước.

1. Tạp chí xuất bản 01 số/quý bằng hai ngôn ngữ tiếng Việt và tiếng Anh. Tạp chí nhận đăng các bài báo khoa học thuộc các lĩnh vực: Điện - Điện tử - Tự động hóa; Cơ khí - Động lực; Kinh tế; Triết học - Xã hội học - Chính trị học; Các lĩnh vực khác gồm: Công nghệ thông tin; Hóa học - Công nghệ thực phẩm; Ngôn ngữ học; Toán học; Vật lý; Văn hóa - Nghệ thuật - Thể dục thể thao...
2. Bài nhận đăng là những công trình nghiên cứu khoa học chưa công bố trong bất kỳ ấn phẩm khoa học nào.
3. Tòa soạn chỉ nhận bài báo gửi online trên website http://tapchikhcn.saodo.edu.vn. Bài báo gửi về toà soạn dưới dạng file điện tử (*.doc *.docx và *.pdf); cuối bài báo, tác giả ghi rõ thông tin địa chỉ liên hệ, số điện thoại, email và cập nhật thông tin trên website. Bài báo phải được trình bày đúng định dạng, rõ ràng; Trường hợp bài báo phải chỉnh sửa theo thể lệ hoặc theo yêu cầu của Phản biện thì tác giả sẽ cập nhật trên website. Người phản biện sẽ do toà soạn mời. Toà soạn không gửi lại bài nếu không được đăng.
4. Các công trình thuộc đề tài nghiên cứu có Cơ quan quản lý cần kèm theo giấy phép cho công bố của cơ quan (Tên đề tài, mã số, tên chủ nhiệm đề tài, cấp quản lý,...).
5. Tên bài báo trình bày bằng hai ngôn ngữ (tiếng Việt và tiếng Anh), font Arial, cỡ chữ 14, in đậm, căn giữa.
6. Tên tác giả (không ghi học hàm, học vị), font Arial, cỡ chữ 10, in đậm, căn lề phải; cơ quan công tác của các tác giả, font Arial, cỡ chữ 9, in nghiêng, căn lề phải.
7. Chữ "Tóm tắt" in đậm, font Arial, cỡ chữ 10; Nội dung tóm tắt của bài báo không quá 10 dòng, trình bày bằng hai ngôn ngữ (tiếng Việt và tiếng Anh), font Arial, cỡ chữ 10, in thường.
8. Chữ "Từ khóa" in đậm, nghiêng, font Arial, cỡ chữ 10; Có từ 03÷05 từ khóa, font Arial, cỡ chữ 10, in nghiêng, ngăn cách nhau bởi dấu chấm phẩy, cuối cùng là dấu chấm.
9. Nội dung bài báo viết bằng tiếng Việt hoặc tiếng Anh; Nếu là bài báo viết bằng tiếng Việt: Tiêu đề tiếng Việt trước, tiếng Anh sau; Tóm tắt tiếng Việt trước, tiếng Anh sau; Từ khóa tiếng Việt trước, tiếng Anh sau; Nếu là bài báo viết bằng tiếng Anh: Tiêu đề tiếng Anh trước, tiếng Việt sau; Tóm tắt tiếng Anh trước, tiếng Việt sau; Từ khóa tiếng Anh trước, tiếng Việt sau.
10. Bài báo được đánh máy trên khổ giấy A4 (21 × 29,7cm) có độ dài không quá 8 trang, font Arial, cỡ chữ 10, giãn dòng At least 12pt, Before 3pt, After 3pt; căn lề trên 2.5cm, dưới 2.5cm, trái 3cm, phải 2cm; hình vẽ phải rõ ràng, đủ nét và được định dạng dưới dạng file ảnh (*.jpg); Phương trình, công thức phải soạn thảo bằng Mathtype hoặc Equation; Phần nội dung bài báo được chia thành 02 cột, khoảng cách cột là 1cm; Trong trường hợp hình vẽ, hình ảnh có kích thước lớn, bảng biểu có độ rộng lớn hoặc công thức, phương trình dài thì cho phép trình bày dưới dạng 01 cột.
11. Tài liệu tham khảo được sắp xếp theo thứ tự tài liệu được trích dẫn trong bài báo.
    - Nếu là sách/luận án: Tên tác giả (năm), Tên sách/luận án/luận văn, Nhà xuất bản/Trường/Viện, lần xuất bản/tái bản.
    - Nếu là bài báo/báo cáo khoa học: Tên tác giả (năm), Tên bài báo/báo cáo, Tạp chí/Hội nghị/Hội thảo, Tập/Kỳ yếu, số, trang.
    - Nếu là trang web: Phải trích dẫn đầy đủ tên website và đường link, ngày cập nhật.
12. Định dạng mẫu bài báo tham khảo tại địa chỉ http://tapchikhcn.saodo.edu.vn/news/detail/198/format_paper
    Bài báo sau khi xuất bản sẽ được công bố trên http://tapchikhcn.saodo.edu.vn.

THÔNG TIN LIÊN HỆ:

Ban Biên tập Tạp chí Nghiên cứu khoa học, Trường Đại học Sao Đỏ
Phòng 203, Tầng 2, Nhà B1, Trường Đại học Sao Đỏ.
Địa chỉ: Số 24, Thái Học 2, phường Sao Đỏ, thành phố Chí Linh, tỉnh Hải Dương.
Điện thoại: (0220) 3587213, Fax: (0220) 3882921, Hotline: 0912 107858/0936 847980.
Website: http://tapchikhcn.saodo.edu.vn
Email: tapchikhcn@saodo.edu.vn

TẠP CHÍ
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ

TRONG SỐ NÀY

Số 3(78) 2022

LIÊN NGÀNH ĐIỆN - ĐIỆN TỬ - TỰ ĐỘNG HÓA

NGÀNH CÔNG NGHỆ THÔNG TIN

LIÊN NGÀNH CƠ KHÍ - ĐỘNG LỰC

TẠP CHÍ
NGHIÊN CỨU KHOA HỌC
ĐẠI HỌC SAO ĐỎ

TRONG SỐ NÀY

Số 3(78) 2022

SCIENTIFIC JOURNAL SAO DO UNIVERSITY

CONTENTS

No 3(78) 2022

TITLE FOR ELECTRICITY - ELECTRONICS - AUTOMATION

TITLE FOR INFORMATION TECHNOLOGY INDUSTRY

TITLE FOR MECHANICAL AND DRIVING POWER ENGINEERING

SCIENTIFIC JOURNAL SAO DO UNIVERSITY

CONTENTS

No 3(78) 2022

Xây dựng nền tảng lưu trữ và phân tích dữ liệu lớn với Apache Hadoop và Spark

Building big data platform storage and analytics with Apache Hadoop and Spark

Vũ Bảo Tạo[1*], Đặng Văn Nam[2], Nông Thị Oanh[2],
Hoàng Thị Ngát[1], Nguyễn Thị Ánh Tuyết[1]

**Email: taovb2006@gmail.com*

[1]Trường Đại học Sao Đỏ
[2]Trường Đại học Mỏ - Địa chất Hà Nội

Ngày nhận bài: 24/01/2022
Ngày nhận bài sửa sau phản biện: 27/6/2022
Ngày chấp nhận đăng: 30/9/2022

Tóm tắt

Hiện nay, dữ liệu đã và đang trở nên ngày càng quan trọng. Dữ liệu là yếu tố quyết định, ảnh hưởng tới hầu hết các lĩnh vực như tài chính - ngân hàng, y tế, giáo dục, nông nghiệp, năng lượng…. Tốc độ sinh dữ liệu ngày càng nhanh với khối lượng ngày càng lớn và thuật ngữ Dữ liệu lớn (Big data) cũng ra đời. Dữ liệu lớn là một trong những công nghệ chủ chốt của cuộc Cách mạng công nghiệp 4.0. Tuy nhiên, việc lưu trữ và phân tích dữ liệu lớn cũng đòi hỏi những kiến thức và công nghệ phù hợp. Chúng ta không thể sử dụng các kỹ thuật lưu trữ và phân tích dữ liệu truyền thống với dữ liệu lớn được. Trong bài báo này, nhóm tác giả sẽ trình bày việc triển khai xây dựng nền tảng lưu trữ dữ liệu lớn sử dụng Apache Hadoop trên một cụm (cluster) các máy tính và Apache Spark để phân tích dữ, trích rút các thông tin có ích (insights) từ tập dữ liệu lưu trữ trên các máy tính này.

Từ khóa: *Dữ liệu lớn; phân tích dữ liệu lớn; cụm máy tính; Hadoop; Spark.*

Abstract

Data has become more and more important. Data is the decisive factor, affecting almost all fields such as finance, banking, healthcare, education, agriculture, energy…The speed of data generation is getting faster and faster with increasing volume and the term Big data was also born. Big data is one of the key technologies of the Industrial Revolution 4.0. However, storing and analyzing big data also requires the right knowledge and technology. We can't use traditional data storage and analysis techniques with big data. In this article, the authors will present the implementation of building a big data storage platform using Apache Hadoop on a cluster of computers and Apache Spark to analyze data, extract valuable from the dataset stored on these compters.

Keywords: *Big data; big data analytics; cluster; Hadoop; Spark.*

1. ĐẶT VẤN ĐỀ

Sự bùng nổ về dữ liệu đang được thể hiện rất rõ trong những năm gần đây; Thuật ngữ "Dữ liệu không bao giờ ngủ - Data never sleep" đã không còn xa lạ. Lượng dữ liệu được tạo ra trong 2 năm gần đây bằng toàn bộ dữ liệu được tạo ra trước đó. Theo ước tính, đến năm 2025, sẽ có 463 Exabytes dữ liệu sẽ được sinh ra mỗi ngày và dự kiến đạt 180 Zettabyte [1]. Lượng dữ liệu khổng lồ này được tạo ra không chỉ bởi các Email, tin nhắn, website, mạng xã hội, hình ảnh, video... mà còn một lượng lớn dữ liệu được tạo ra bởi máy móc mà không cần tác động của người dùng, như các thiết bị cảm biến kết nối Internet (IoT).

Khi nói tới dữ liệu lớn là nói tới khối lượng lớn dữ liệu, bao gồm cả dữ liệu có cấu trúc và không có cấu trúc mà chúng ta đang tạo ra và đối mặt hàng ngày. Thuật ngữ "Dữ liệu lớn" là thuật ngữ được sử dụng cho tập hợp các dữ liệu quá lớn và phức tạp khiến cho việc xử lý các dữ liệu này trở nên khó khăn khi sử dụng các kỹ thuật truyền thống [1]. Dữ liệu lớn thường được xác định thông qua mô hình 5V [2] (Hình 1) bao gồm:

Volume (Dung lượng): Đây là đặc tính đầu tiên và phổ biến nhất của dữ liệu lớn, đặc tính này cho biết độ lớn của dữ liệu được sinh ra.

Velocity (Tốc độ): Không chỉ có nhiều nguồn dữ liệu khác nhau từ các thiết bị, máy móc, con người mà tốc độ sinh dữ liệu cũng tăng lên liên tục. Khía cạnh tốc độ trong dữ liệu lớn bên cạnh quan tâm tới tốc độ sinh dữ liệu còn là tốc độ truyền dữ liệu.

Người phản biện: 1. PGS. TS. Nguyễn Long Giang
2. PGS. TS. Lê Hồng Anh

Variety (Đa dạng): Dữ liệu tồn tại ở nhiều định dạng khác nhau từ dữ liệu có cấu trúc, dữ liệu số trong các cơ sở dữ liệu truyền thống, cho đến các tài liệu văn bản không có cấu trúc, email, âm thanh, hình ảnh, các giao dịch tài chính...

Veracity (Chính xác): Đặc tính này đề cập tới độ tin cậy của dữ liệu thu thập được từ các nguồn khác nhau, giúp ta hiểu rõ hơn về các rủi ro của các phân tích và các quyết định đưa ra dựa trên một tập dữ liệu cụ thể.

Value (Giá trị): Đây là đặc trưng được coi là quan trọng nhất của dữ liệu lớn. Nó đề cập tới những tri thức (insights), giá trị có thể trích rút ra được từ dữ liệu. Dữ liệu lớn sẽ không có ý nghĩa gì nếu chúng ta không chuyển được chúng thành những thứ có giá trị.

Hình 1. *Mô hình 5V của dữ liệu lớn*

Ngày nay, thị trường công nghệ về dữ liệu lớn liên tục phát triển. Mọi khía cạnh trong đời sống của chúng ta đều sẽ bị ảnh hưởng bởi dữ liệu lớn. Việc thu thập, lưu trữ và phân tích dữ liệu lớn để phát hiện ra các tri thức tiềm ẩn bên trong dữ liệu là rất quan trọng, giúp hỗ trợ quá trình ra quyết định mang lại nhiều giá trị to lớn [3]. Các lĩnh vực đang sử dụng dữ liệu lớn đem lại hiệu quả có thể chỉ ra như trong lĩnh vực giáo dục, y tế, quản lý nhà nước, tài chính - ngân hàng, giao thông vận tải, truyền thông và giải trí, quản lý tài nguyên và giám sát thiên tai.... Ngoài mô hình 5V như ở trên, các nhà nghiên cứu hiện nay còn đề xuất các mô hình 10V, 14V để đề cấp tới nhiều đặc trưng và bao quát hơn của Dữ liệu lớn [4].

Không ai có thể phủ nhận được tầm quan trọng và lợi ích của Dữ liệu lớn đã, đang và sẽ mang lại. Tuy nhiên, để có thể triển khai và vận hành một hệ thống dữ liệu lớn trong thực tế sẽ phải đối mặt với một loạt các thách thức từ chi phí tới hạ tầng thiết bị, công nghệ và kỹ thuật [5], [6].

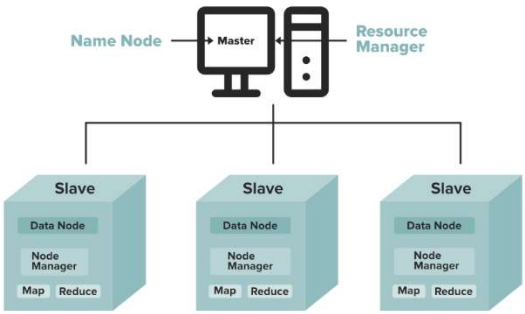
Cấu trúc của bài báo bao gồm các nội dung từ nghiên cứu lý thuyết tới triển khai thực nghiệm xây dựng một hệ thống dữ liệu lớn hoàn chỉnh. Trong đó, nội dung phần 2 của bài báo sẽ được trình bày tiếp theo đây tập trung vào việc giới thiệu các nền tảng lưu trữ và phân tích dữ liệu lớn mạnh mẽ và phổ biến hiện nay là Apache Hadoop và Apache Spark. Phần 3 cũng là phần trọng tâm của bài báo, tập trung vào việc thiết kế và triển khai xây dựng một cụm với 3 máy tính trong đó một máy đóng vai trò là Namenode và 2 máy Datanode, sử dụng hệ điều hành Ubuntu 20.04. Các máy tính trong cụm sẽ được kết nối, cho phép lưu trữ dữ liệu phân tán. Hệ thống này đang được vận hành ổn định, có khả năng năng mở rộng theo cả chiều ngang và chiều dọc một cách nhanh chóng. Đồng thời, chúng tôi cũng thực hiện lưu trữ, phân tích trên một tập dữ liệu cụ thể là Data_Uber sử dụng Spark SQL để truy vấn, trích rút thông tin có ích và Spark MLLib để xây dựng mô hình học máy, thực hiện phân cụm dữ liệu với thuật toán Kmeans.

2. NỀN TẢNG LƯU TRỮ VÀ PHÂN TÍCH DỮ LIỆU LỚN

2.1. Apache Hadoop

Như đã trình bày trong phần đặt vấn đề, chúng ta không thể sử dụng các công nghệ và kỹ thuật truyền thống để lưu trữ và phân tích dữ liệu lớn được. Do đó, cần phải triển khai và ứng dụng các công nghệ phù hợp đáp ứng được các yêu cầu về hiệu năng, tính sẵn sàng, độ an toàn và khả năng chịu lỗi cao của một hệ thống dữ liệu lớn. Apache Hadoop là một trong những hệ thống đáp ứng được các yêu cầu đó và đang được sử dụng rộng rãi.

Apache Hadoop là một framework mã nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán [7]. Nó được thiết kế để mở rộng quy mô tới hàng ngàn máy tính khác nhau trong một cụm. Apache Hadoop được thiết kế theo kiến trúc chủ - khách (Master - Slave) bao gồm một nút chủ (Namenode) và nhiều nút khách (Datanode) như mô tả trong Hình 2.

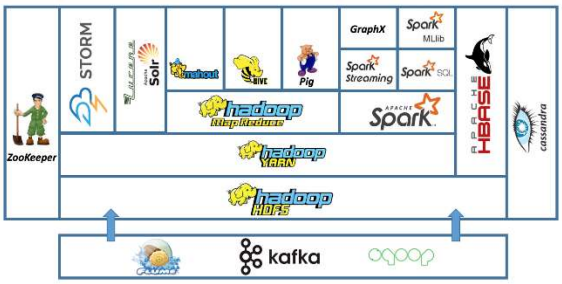


Hình 2. *Kiến trúc Master – Slave của Hadoop*

Chức năng của nút chủ là gán tác vụ cho nhiều nút khách khác nhau, lưu trữ siêu dữ liệu và quản lý tài nguyên. Các nút khách (Datanode) lưu trữ dữ liệu thực và thực hiện các tính toán.

Hệ sinh thái của Hadoop bao gồm nhiều thành phần khác nhau từ thu thập cho tới lưu trữ và phân tích một lượng dữ liệu lớn tính bằng Petabytes (Hình 3). Trong đó, hai thành phần chính của Hadoop bao gồm: Hệ thống file phân tán (Hadoop Distributed File System - HDFS), đây là phần lõi và là xương sống của hệ sinh thái Hadoop, có khả năng lưu trữ các bộ dữ liệu lớn với nhiều định dạng. Dữ liệu được lưu trữ trên các

Datanode và sẽ được sao lưu, nhân bản trên các Datanode khác nhau để làm tăng khả năng chịu lỗi của hệ thống. MapReduce và Spark là nền tảng xử lý và tính toán dữ liệu phân tán. Nó cho phép thực hiện các tính toán ở kích cỡ lớn một cách dễ dàng. Chi tiết về Spark sẽ được trình bày trong phần 2.2 dưới đây.

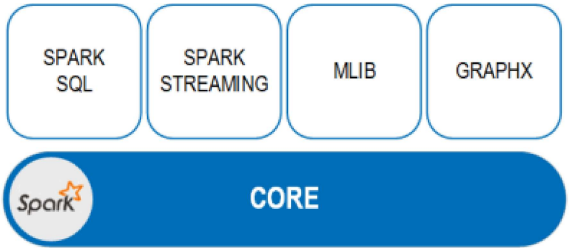


Hình 3. *Hệ sinh thái của Apache Hadoop*

2.2. Apache Spark

Apache Spark là một dự án xử lý dữ liệu phân tán mã nguồn mở [8]. Spark cho phép xử lý dữ liệu lớn với tốc độ nhanh hơn MapReduce tới 100 lần, hỗ trợ xử lý nhiều định dạng và nguồn dữ liệu, đồng thời tích hợp nhiều bộ thư viện mạnh mẽ để xử lý dữ liệu [9]. Nhờ tính dễ sử dụng nên Spark trở nên dễ dàng đối với các nhà khoa học dữ liệu, phân tích dữ liệu, lập trình viên. Spark hỗ trợ nhiều ngôn ngữ khác nhau như Java, Scale, Python, R vì thế người dùng có thể dễ dàng lựa chọn công cụ để phát triển các ứng dụng để giải quyết các bài toán xử lý dữ liệu có kích thước lớn.

Apache Spark rất linh hoạt, nó cung cấp một nền tảng hợp nhất để xử lý dữ liệu và có thể sử dụng cho nhiều dạng xử lý khác nhau như xử lý theo lô, thực hiện các tương tác truy vấn, xử lý sử dụng các giải thuật học máy và xử lý thời gian thực. Trước khi có Spark, người dùng phải sử dụng nhiều công nghệ khác nhau để giải quyết những bài toán này, việc sử dụng một nền tảng hợp nhất sẽ giúp làm giảm chi phí và tài nguyên, tối ưu hóa hệ thống. Apache Spark có 5 thành phần chính (Hình 4), bao gồm:



Hình 4. *Các thành phần của Apache Spark*

Spark Core: Là phần lõi và nền tảng để xây dựng tất cả các chức năng của Apache Spark. Spark Core bao gồm 2 thành phần là hạ tầng tính toán phân tán và lập trình RDD.

Spark SQL: Là thành phần nền tảng phân tán cho xử lý dữ liệu có cấu trúc, tương thích hoàn toàn với Hive, có thể truy xuất đến nhiều nguồn dữ liệu khác nhau như Avro, Parquet, ORC, JSON, CSV... Spark SQL là thành phần phát triển nhanh nhất, người dùng có thể dễ dàng sử dụng các câu lệnh SQL để thao tác, xử lý dữ liệu phân tán.

Spark Streaming: Là thành phần cho phép xử lý dữ liệu luồng thời gian thực với băng thông cao, khả năng chịu lỗi lớn.

MLLib: Thành phần này cung cấp cơ chế để quản lý và đơn giản hóa nhiệm vụ xây dựng các mô hình học máy, cung cấp thư viện với nhiều giải thuật học máy phổ biến.

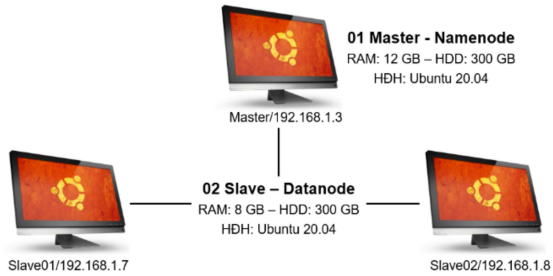
GraphX: Thư viện xử lý đồ thị phân tán, cung cấp các tính toán đồ thị và trừu tượng hóa đồ thị dựa trên các RDD.

3. XÂY DỰNG CỤM MÁY TÍNH LƯU TRỮ VÀ PHÂN TÍCH DỮ LIỆU LỚN

3.1. Thiết kế và xây dựng cụm máy tính với Apache Hadoop để lưu trữ dữ liệu

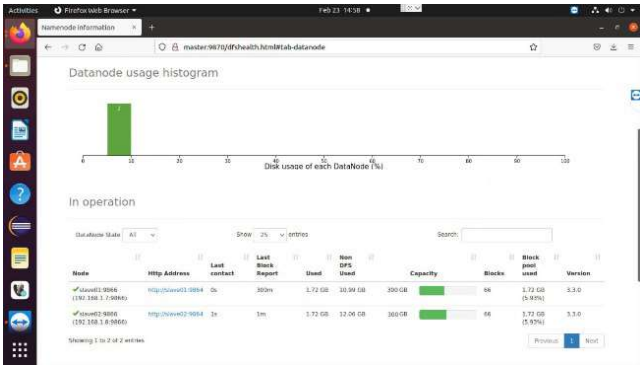
Trong phần 2, chúng tôi đã trình bày tổng quan về nền tảng được sử dụng để lưu trữ và phân tích dữ liệu lớn mạnh mẽ hiện nay là Apache Hadoop và Spark. Trong phần 3 này, chúng tôi sẽ tiến hành triển khai thiết kế và xây dựng một cụm (cluster) các máy tính để tạo thành một hệ thống lưu trữ và phân tích dữ liệu phân tán trên nền tảng Apache Hadoop.

Do điều kiện về hạ tầng thiết bị có hạn, chúng tôi sử dụng 3 máy tính khác nhau để tạo thành một cụm theo mô hình Master - Slave trong đó: 01 máy tính đảm nhận vai trò Master, 02 máy tính đóng vai trò Slave. Các thông số cụ thể của cụm các máy tính được mô tả như Hình 5 dưới đây. Máy Master đảm nhận chức năng của một Namenode, không chứa dữ liệu thực mà chỉ chứa bảng tham chiếu tới địa chỉ chứa dữ liệu, lập lịch và quản lý tài nguyên vì thế cần ít bộ nhớ lưu trữ và các tài nguyên tính toán cao. Máy Slave đảm nhận chức năng của Datanode là nơi lưu trữ dữ liệu thực trong một môi trường phân tán.



Hình 5. *Thiết kế 3 máy tính tạo thành một Cluster theo mô hình Master - Slave*

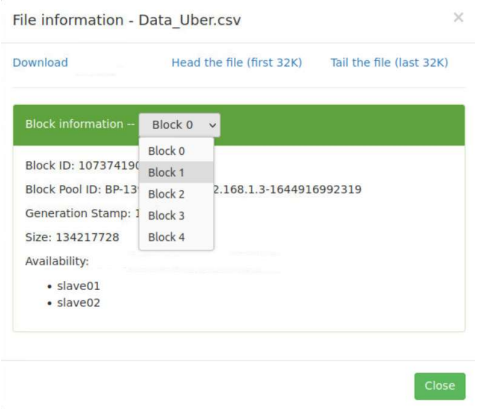
Sau khi thiết kế xong, triển khai cài đặt Hadoop trên các máy tính và cấu hình các tham số của hệ thống. Chúng tôi sử dụng phiên bản Apache Hadoop 3.3, đây là phiên bản mới nhất hiện nay. Hệ thống đã được cài đặt và kết nối với nhau thành công, tạo được một nền tảng cho phép lưu trữ và xử lý một khối lượng lớn dữ liệu hàng terabytes và petabytes (Hình 6).



Hình 6. *Thông tin phiên bản Hadoop và danh sách các máy Datanode trong cụm*

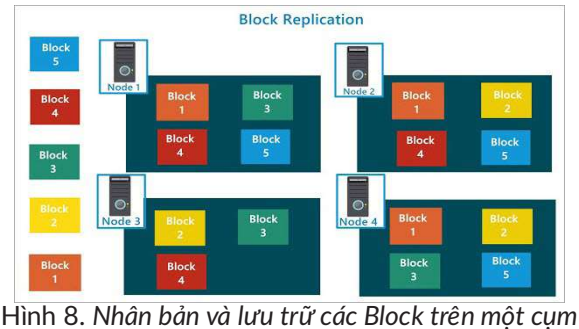
Với kiến trúc lưu trữ HDFS, một file có kích thước lớn bất kỳ hàng Gigabytes dữ liệu khi lưu trữ trên hệ thống Hadoop sẽ được chia ra thành các khối (mặc định 128Mb trong phiên bản 3.3, người dùng có thể thiết lập kích thước khối cho phù hợp) và được lưu trữ như các đơn vị độc lập phân tán trên các Datanode của cụm. Việc lưu trữ dữ liệu dưới dạng các khối sẽ giải quyết được vấn đề khi cần lưu trữ một file có kích thước lớn, đồng thời nó giúp đơn giản hóa hệ thống lưu trữ, phù hợp với việc nhân bản dữ liệu để làm tăng khả năng chịu lỗi và tính sẵn sàng.

File dữ liệu Data_Uber.csv [10] mà chúng tôi sử dụng để phân tích sẽ được trình bày trong 3.2 có kích thước 558MB khi lưu trữ trên hệ thống HDFS của Hadoop sẽ được chia ra thành 5 khối, từ Block0 đến Block4. Hệ thống không quan tâm tới định dạng file mà sẽ thực hiện tách thành từng khối, trong đó các khối Block0 đến Block3 có kích thước 128MB, khối Block4 chứa phần dữ liệu còn lại của file tương ứng với 46MB (Hình 7).



Hình 7. *Phân tách file dữ liệu Data_Uber.csv thành các khối và lưu trữ trên HDFS*

Trong trường hợp một cụm có nhiều Datanode thì các khối sẽ được lưu trữ phân tán trên các node này. Đồng thời, Hadoop cũng thực hiện nhân bản các khối (Block replication) để giúp làm tăng khả năng chịu lỗi của hệ thống. Hệ số nhân bản mặc định là 2, nghĩa là với mỗi một block sẽ được nhân bản thêm một block để lưu trữ trên các node khác nhau. Tùy thuộc các yêu cầu về an toàn dữ liệu và nền tảng phần cứng, người dùng có thể thiết lập hệ số nhân bản cho phù hợp [11]. Hình 8 dưới đây minh họa việc lưu trữ 5 Block trên một cụm gồm 4 Datanode với hệ số nhân bản là 3. Mỗi một block sẽ được nhân bản thành 3 bản sao và được lưu trữ trên các Datanode khác nhau, việc này đảm bảo rằng khi sự cố xảy ra ở một hoặc hai datanote bất kỳ trong cụm thì hệ thống vẫn hoạt động bình thường.



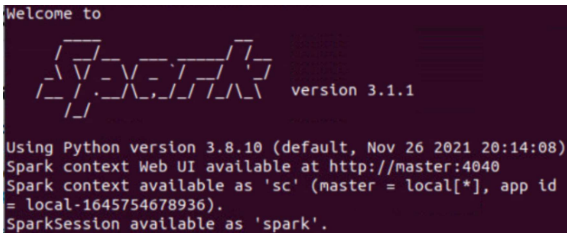
Hình 8. *Nhân bản và lưu trữ các Block trên một cụm gồm 4 Datanode*

Việc chia thành các khối, nhân bản và lưu trữ trên các Datanode được thực hiện "trong suốt" với người dùng. Người dùng khi đọc, ghi các file dữ liệu trên Hadoop được thực hiện như là trên một file duy nhất. Việc thực hiện đọc và phân tích file dữ liệu lưu trữ trên hệ thống phân tán HDFS sẽ được chúng tôi trình bày trong nội dung tiếp theo.

3.2. Truy vấn và phân tích dữ liệu lớn với Apache Spark

Apache Spark như đã trình bày trong nội dung 2.2 là một dự án xử lý dữ liệu phân tán mã nguồn mở. Spark được viết bằng Scala trên nền JVM và Java runtime, có thể chạy đa nền tảng trên Linux và cả Windows. Nó hỗ trợ xử lý nhiều định dạng và nguồn dữ liệu khác nhau, đồng thời tích hợp nhiều bộ thư viện mạnh để xử lý dữ liệu. Spark hỗ trợ nhiều ngôn ngữ lập trình khác nhau, trong nội dung bài báo này chúng tôi sử dụng ngôn ngữ lập trình Python để đọc và phân tích dữ liệu, đây hiện tại cũng là ngôn ngữ lập trình được sử dụng nhiều nhất với Spark.

Thực hiện cài đặt và cấu hình PySpark lên hệ thống, chúng tôi sử dụng Spark phiên bản 3.3.1, Python phiên bản 3.8.10 và Jupyter notebook để truy vấn dữ liệu.



Hình 9. Phiên bản Spark và Python sử dụng

Để có thể thao tác được với dữ liệu lưu trữ trên HDFS của Hadoop cần tạo một phiên (Spark Session) kết nối với máy chủ Master. Hình 10 dưới đây là đoạn mã nguồn thực hiện khởi tạo một session với máy Master.

```
In [2]: #Khởi tạo sparksession
        import findspark
        findspark.init()
        from pyspark.sql import SparkSession
        spark = SparkSession.builder.master("master[9870]").getOrCreate()
        spark

Out[2]: SparkSession - hive
        SparkContext

        Spark UI
        Version
        v3.1.1
        Master
        local[2]
        AppName
        PySparkShell
```

Hình 10. *Khởi tạo SparkSession tới máy Master*

File dữ liệu Data_Uber.csv được sử dụng để phân tích đã được lưu trữ trên HDFS trong cụm máy tính. Chúng ta sẽ sử dụng Spark để đọc tập dữ liệu này vào biến kiểu DataFrame, DataFrame là tập hợp dạng bảng với các hàng và cột. Tập dữ liệu lưu trữ thông tin thời điểm, vị trí đón khách và mã công ty quản lý phương tiện bao gồm 4 cột: Datetime - Thời điểm đón khách; lat - Kinh độ; lon - Vĩ độ vị trí đón khách; base - Mã đơn vị quản lý phương tiện chở khách. Đây là tập dữ liệu chở khách của Uber từ ngày 01/8 đến hết 31/8/2014.

```
#Đọc dữ liệu từ file .csv lưu trữ trên HDFS của Hadoop
path = '/Data_HUMG/db_uber/Data_Uber.csv'
df_uber = spark.read.schema(schema).load(path,
                                         format='csv',
                                         inferSchema=True,
                                         header=False)
df_uber.printSchema()  #Hiển thị lược đồ dữ liệu
df_uber.show(5)        #Hiển thị 5 bản ghi đầu tiên

root
 |-- datetime: timestamp (nullable = true)
 |-- lat: float (nullable = true)
 |-- lon: float (nullable = true)
 |-- base: string (nullable = true)

+-------------------+-------+--------+------+
|           datetime|    lat|     lon|  base|
+-------------------+-------+--------+------+
|2014-08-01 00:00:00| 40.729|-73.9422|B02598|
|2014-08-01 00:00:00|40.7476|-73.9871|B02598|
|2014-08-01 00:00:00|40.7424|-74.0044|B02598|
|2014-08-01 00:00:00| 40.751|-73.9869|B02598|
|2014-08-01 00:00:00|40.7406|-73.9902|B02598|
+-------------------+-------+--------+------+
only showing top 5 rows
```

Hình 11. *Đọc tập dữ liệu lưu trữ trên HDFS của Hadoop với PySpark*

Như đã trình bày, việc chia tập dữ liệu thành các khối (block), nhân bản và lưu trữ vào các Datanode trong cụm cũng như việc tổng hợp lại để truy vấn dữ liệu Hadoop sẽ thực hiện "trong suốt" với người dùng, người dùng thực hiện giống như trên một file vật lý duy nhất. File dữ liệu Data_Uber.csv được chia thành 5 block (block0 tới block4) và được lưu trữ phân tán trên 2 Datanode của cụm, hệ số nhân bản mặc định là 2. Khi truy xuất dữ liệu, máy Namenode sẽ ánh xạ tham chiếu tới các khối tương ứng trên các Datanode khác nhau để tạo thành một file dữ liệu hoàn chỉnh.

Phần 2.2 đã trình bày 5 thành phần chính của Apache Spark, trong đó SparkCore là phần lõi và nền tảng để xây dựng tất cả các chức năng của Apache Spark, cung cấp khả năng tính toán trong bộ nhớ, xử lý tập dữ liệu lớn song song và phân tán. Các thành phần còn lại bao gồm SparkSQL, Spark Streaming, Spark MLLib và GraphX sẽ chạy trên SparkCore. Chúng tôi sẽ sử dụng SparkSQL để truy vấn tập dữ liệu và thực hiện một số thống kê số trên tập dữ liệu Data_Uber.

SparkSQL là thành phần nền tảng phân tán cho xử lý dữ liệu có cấu trúc, có thể truy xuất đến đa dạng nguồn dữ liệu như: Avro, Parquet, ORC, JSON, CSV, JDBC. SparkSQL mang đến một sự tiện lợi, mềm dẻo, hiệu năng cao đối với dữ liệu có cấu trúc kích cỡ Petabytes. Chúng ta có thể sử dụng các câu lệnh SQL quen thuộc để thao tác xử lý dữ liệu. Theo thống kê, SparkSQL là thành phần phát triển nhanh nhất vì nó cho phép những người sử dụng SQL có thể tiếp cận sức mạnh của nền tảng xử lý dữ liệu phân tán.

Để thực hiện thống kê dữ liệu với SparkSQL chúng ta có thể sử dụng các toán tử mà SparkSQL cung cấp như select, where, groupBy, orderBy… để truy vấn dữ liệu trong DataFrame; Hình 12 minh họa việc thống kê đếm số lượng chuyến theo từng mã hãng quản lý phương tiện. Ở đây, chúng tôi sử dụng toán tử groupBy để nhóm dữ liệu theo cột 'base', orderBy để sắp xếp theo thứ tự giảm dần về số lượng.

```
#Thống kê số chuyến theo từng hãng vận chuyển:
from pyspark.sql.functions import col, asc,desc
df_base = df_uber.groupBy(col("base")).count().orderBy(col("count").desc())
df_base.show()

+------+------+
|  base| count|
+------+------+
|B02617|355803|
|B02598|220129|
|B02682|173280|
|B02764| 48591|
|B02512| 31472|
+------+------+
```

Hình 12. *Thống kê số chuyến theo từng hãng với SparkSQL*

Kết quả câu truy vấn dữ liệu cho thấy có 5 đơn vị quản lý phương tiện, trong đó đơn vị có mã 'B02617' có số lượt chở khách nhiều nhất với trên 350 nghìn lượt, thấp nhất là mã 'B02512' với trên 31 nghìn lượt. Như vậy, trong tháng 8/2014 đơn vị 'B02617' có số lượt chở khách cao hơn 10 lần so với 'B02512'.

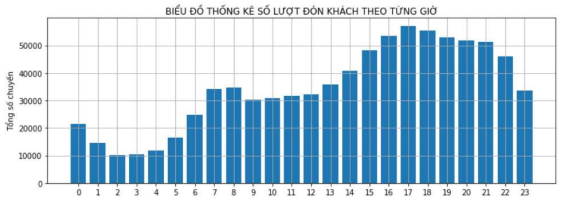
Ngoài việc sử dụng các toán tử mà SparkSQL cung cấp, chúng ta cũng có thể viết trực tiếp các câu truy vấn SQL như với cơ sở dữ liệu quan hệ. Trong Hình 13 dưới đây thực hiện truy vấn dữ liệu sử dụng SparkSQL để thống kê số lượt đón khách theo từng giờ trong ngày. Chúng ta thấy ngay rằng, người dùng hoàn toàn có thể sử dụng các câu lệnh SQL quen thuộc khi làm việc với các hệ quản trị cơ sở dữ liệu để truy vấn dữ liệu phân tán với SparkSQL. Đây cũng là một trong số lý do giúp cho SparkSQL có tốc độ phát triển nhanh, được đông đảo mọi người sử dụng.

Nhìn vào biểu đồ thống kê trên toàn tập dữ liệu Data_Uber đã thực hiện, chúng ta có thể thấy ngay rằng khung thời gian từ 16 giờ đến 21 giờ hàng ngày có số lượng khách đi xe cao hơn các khung thời gian khác, cao nhất là thời điểm 17 giờ với trên 55 nghìn lượt đi xe. Vào buổi sáng, khách chủ yếu tập trung vào khung 7 giờ, 8 giờ cao hơn các khung giờ còn lại. Những thông tin chúng ta trích rút được từ dữ liệu sẽ đem lại nhiều lợi ích trong quá trình quản lý, vận hành và tối ưu hóa hoạt động của doanh nghiệp.

Các nội dung ở trên, chúng tôi tập trung vào việc khai thác SparkSQL của Apache Spark để truy vấn dữ liệu phân tán lưu trữ trên HDFS của Hadoop, chúng ta hoàn toàn có thể sử dụng các thư viện khác như

Spark MLIB, Spark Streaming… để xây dựng các mô hình học máy, xử lý dữ liệu thời gian thực với dữ liệu phân tán.

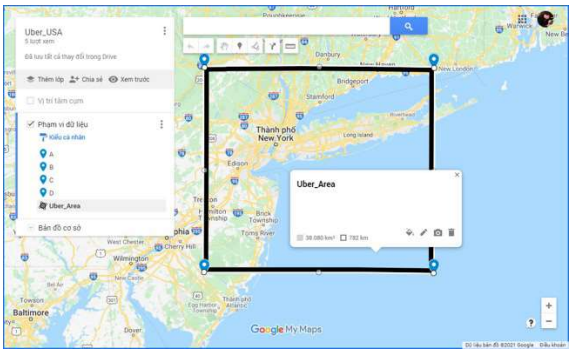
```
#Tạo một bảng tạm tbl_uber từ DataFrame:
df_uber.createOrReplaceTempView("tbl_uber")
#Truy vấn dữ liệu với SparkSQL
sql1 = """SELECT hour(tbl_uber.datetime) as hr, count(tbl_uber.base) as ct
          FROM tbl_uber GROUP BY hr ORDER BY hr"""
a = spark.sql(sql1)
#Trực quan hóa dữ liệu
import matplotlib.pyplot as plt
x=a.toPandas()["hr"].values.tolist()
y=a.toPandas()["ct"].values.tolist()
plt.bar(list(map(str,x)),y)
plt.title('BIỂU ĐỒ THỐNG KÊ SỐ LƯỢT ĐÓN KHÁCH THEO TỪNG GIỜ')
plt.xlabel("Giờ - Hours")
plt.ylabel("Tổng số chuyến")
plt.show()
```



Hình 13. *Truy vấn dữ liệu với SparkSQL*

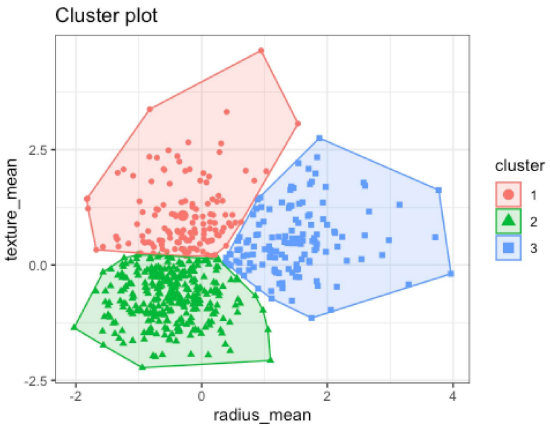
MLIB là một phần của hệ sinh thái Spark, cung cấp cơ chế quản lý và đơn giản hóa nhiệm vụ xây dựng các mô hình học máy. Trong phần này chúng tôi nghiên cứu sử dụng Spark MLIB xây dựng một mô hình học máy trên nền tảng dữ liệu lớn được lưu trữ phân tán áp dụng cho dữ liệu Uber đã phân tích ở trên.

Mỗi một điểm đón khách được lưu trữ bao gồm cả kinh độ và vĩ độ nó sẽ được biểu diễn thành một điểm trên bản đồ. Hình 14 cho biết phạm vi và giới hạn các vị trí đón khách trong tập dữ liệu. Để biết được khu vực nào có lượng đón khách nhiều nhất, chúng tôi sẽ sử dụng mô hình phân cụm (Clustering) để gom nhóm các điểm gần nhau lại dựa trên thông tin lat, lon của điểm đó. Phân cụm là một trong những thuật toán phổ biến nhất thuộc lớp bài toán học không giám sát (unsupervised learning).



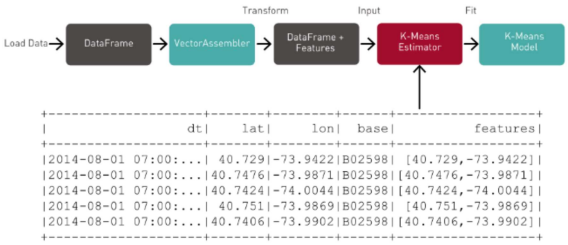
Hình 14. *Phạm vi các vị trí đón khách trong tập dữ liệu trên bản đồ*

Có rất nhiều thuật toán phân cụm khác nhau, trong đó K-means là một trong những thuật toán phổ biến và được sử dụng nhiều nhất [12]. Ý tưởng chung của K-means dựa trên việc xác định các tâm cụm và phân các mẫu vào cụm có tâm gần nhất. Hình 15 minh họa thuật toán phân cụm các điểm dữ liệu vào 3 nhóm.



Hình 15. *Phân cụm sử dụng K-means*

Để sử dụng được thuật toán K-means trong Spark MLIB, cần thực hiện VectorAssember để kết hợp các đặc trưng (features) sẽ sử dụng cho phân cụm [13]. Với bài toán này, 2 thuộc tính lat, lon sẽ được dùng để phân cụm. Hình 16 thể hiện các bước chuẩn bị dữ liệu cho việc phân cụm.




```
+--------------------+-------+--------+------+--------------------+
|                  dt|    lat|     lon|  base|            features|
+--------------------+-------+--------+------+--------------------+
|2014-08-01 07:00:...| 40.729|-73.9422|B02598| [40.729,-73.9422]|
|2014-08-01 07:00:...|40.7476|-73.9871|B02598|[40.7476,-73.9871]|
|2014-08-01 07:00:...|40.7424|-74.0044|B02598|[40.7424,-74.0044]|
|2014-08-01 07:00:...| 40.751|-73.9869|B02598| [40.751,-73.9869]|
|2014-08-01 07:00:...|40.7406|-73.9902|B02598|[40.7406,-73.9902]|
+--------------------+-------+--------+------+--------------------+
```

Hình 16. *VectorAssembler dữ liệu lat, lon*

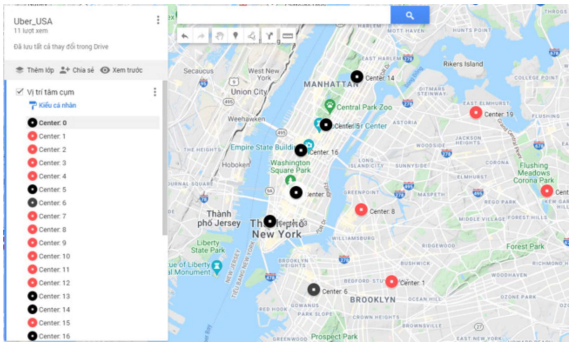
Từ module MLIB, cần import thuật toán Kmeans để phân cụm, tham số bắt buộc cần thiết lập đó là số cụm, chúng tôi thiết lập số cụm bằng 20. Như vậy, thuật toán sẽ thực hiện gom tất cả các điểm đón khách trong tập dữ liệu vào 20 cụm được đánh số từ 0-19 (cid) dựa trên thông số lat, lon (Hình 17).

```
#Sử dụng thuật toán KMeans để phân cụm dữ liệu
#Vị trí (kinh độ, vĩ độ) - features được sử dụng để phân thành 20 cụm

from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator

# Trains a k-means model.
kmeans = KMeans().setK(20).setFeaturesCol("features").setPredictionCol("cid").setSeed(1)
model = kmeans.fit(df_uber_2)

# Make predictions
predictions = model.transform(df_uber_2)

predictions.show()
```

```
+--------------------+-------+--------+------+--------------------+---+
|            datetime|    lat|     lon|  base|            features|cid|
+--------------------+-------+--------+------+--------------------+---+
|2014-08-01 00:00:00| 40.729|-73.9422|B02598|[40.7290000915527...|  8|
|2014-08-01 00:00:00|40.7476|-73.9871|B02598|[40.7476000554199...| 16|
|2014-08-01 00:00:00|40.7424|-74.0044|B02598|[40.7424011230468...| 16|
|2014-08-01 00:00:00| 40.751|-73.9869|B02598|[40.7509994506835...| 16|
|2014-08-01 00:00:00|40.7406|-73.9902|B02598|[40.7406005859375...| 16|
|2014-08-01 00:00:00|40.6994|-73.9591|B02617|[40.6994018554687...|  8|
|2014-08-01 00:00:00|40.6917|-73.9398|B02617|[40.6916999816894...|  1|
|2014-08-01 00:00:00|40.7063|-73.9223|B02617|[40.706298828125,...|  1|
|2014-08-01 00:00:00|40.6759|-74.0168|B02617|[40.6758995056152...|  6|
+--------------------+-------+--------+------+--------------------+---+
```

Hình 17. *Sử dụng Kmeans trong Spark MLIB*

Từ kết quả phân cụm, chúng ta sẽ xác định được vị trí tâm của các cụm này và hiển thị vị trí các tâm cụm lên trên bản đồ. Hình 18, thể hiện vị trí một số tâm cụ trên bản đồ. Dựa trên kết quả phân cụm, chúng ta có thể thực hiện các phân tích sử dụng Spark SQL để thống kê và xác định các cụm có mật độ đón khách cao nhất, mật độ đón khách theo từng khung thời gian trong ngày của từng cụm.



Hình 18. *Hiển thị vị trí tâm cụm trên bản đồ*

4. KẾT LUẬN

Cùng với sự phát triển mạnh mẽ của các công nghệ mới, đặc biệt là sự phát triển của mạng Internet, dữ liệu trở nên rất đa dạng và dữ liệu gia tăng với tốc độ cũng ngày càng nhanh. Do đó, việc nghiên cứu các công nghệ, công cụ để xây dựng và triển khai các hệ thống Dữ liệu lớn là rất cấp thiết. Trong nội dung bài báo này, chúng tôi đã nghiên cứu và tiến hành thiết kế, xây dựng một cụm máy tính theo kiến trúc Master-Slave và triển khai hệ thống mã nguồn mở Apache Hadoop cho phép lưu trữ và tính toán trên hệ thống phân tán. Đồng thời bài báo cũng thực hiện triển khai và truy vấn dữ liệu với Apache Spark. Đây đều là các nền tảng công nghệ mạnh mẽ, phổ biến nhất hiện nay trong việc triển khai và làm việc với Dữ liệu lớn. Từ các bước triển khai này, có thể dễ dàng mở rộng các node trong cụm theo cả chiều ngang và chiều dọc để gia tăng không gian lưu trữ, tính sẵn sàng và sức mạnh tính toán phục vụ cho các bài toán có quy mô lớn hơn.

TÀI LIỆU THAM KHẢO

[1]. Konstantinos Vassakis, Emmanuel Petrakis and Ioannis Kopanakis (2018), *Big Data Analytics: Applications*, Prospects and Challenges, Lecture Notes on Data Engineering and Communications Technologies.

[2]. Ishwarappa and J. Anuradha (2015), *A brief introduction on big data 5Vs characteristics and hadoop technology*, Procedia Comput, *Sci*, vol. 48, pp. 319-324.

[3]. Intel (2015), *White paper: Turn Big Data into Big Value*.

[4]. Arockia Panimalar, Varnekha Shree, Veneshia Kathrine (2017), T*he 17 V's Of Big Data, International Research,* Journal of Engineering and Technology (IRJET), Volume. 04, Issue.09, pp. 329-333.

[5]. Oguntimilehin A., Ademola E.O. (2014), *A Review of Big Data Management, Benefits and Challenges*, Journal of Emerging Trends in Coumputing and Information Sciences, vol-5, pp. 433-437.

[6]. Stephen Kaisler, Frank Armour, Jalberto Espinosa and Wolliam Money (2013), *Big Data: Issues and Challenges Moving Forward*, Hawaii International Conference on System Sciences 46th, pp. 995-1003.

[7]. Piyush Sewal, Hari Singh (2021), *A Critical Analysis of Apache Hadoop and Spark for Big Data Processing*, International Conference on Signal Processing, Computing and Control (ISPCC).

[8]. Carol McDonald (2018), *Getting Started with Apache Spark from Inception to Production*, Ebook, MapR Technologies.

[9]. Eman S.Abead, Mohamed H.Khafagy, Fatma A.Omara (2015), *A Comparative Study of HDFS Replication Approaches*, International Journal in IT and Engineering, Vol.03 Issue-08.

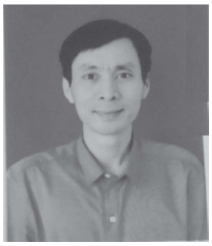
[10]. Chris Albon (2018), *Python Machine Learning Cookbook*, practical solutions from preprocessing to deep learning, O'Reilly Media.

[11]. Adi Polak (2022), *Machine Learning with Spark*, O'Reilly Media.

[12]. https://spark.apache.org/

[13]. https://hadoop.apache.org/

THÔNG TIN TÁC GIẢ



Vũ Bảo Tạo

- Năm 2013: Tốt nghiệp Thạc sĩ ngành Công nghệ thông tin chuyên ngành Công nghệ phần mềm Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên, khoa Công nghệ thông tin, Trường Đại học Sao Đỏ.
- Lĩnh vực quan tâm: Quản trị mạng máy tính, kỹ nghệ phần mềm.
- Điện thoại: 0912519702 Email: taovb2006@gmail.com

Đặng Văn Nam

- Năm 2012: Tốt nghiệp Thạc sĩ, ngành Hệ thống Thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên khoa Công nghệ thông tin, Trường Đại học Mỏ - Địa chất Hà Nội.
- Lĩnh vực quan tâm: Dữ liệu lớn; trí tuệ nhân tạo.
- Điện thoại: 0986226651 Email: dangvannam@humg.edu.vn

Nông Thị Oanh

- Năm 2013: Tốt nghiệp Thạc sĩ, ngành Hệ thống Thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên khoa Công nghệ thông tin, Trường Đại học Mỏ - Địa chất Hà Nội.
- Lĩnh vực quan tâm: Web ngữ nghĩa, khoa học dữ liệu.
- Điện thoại: 0983085852 Email: nongthioanh@humg.edu.vn

Hoàng Thị Ngát

- Năm 2013: Tốt nghiệp Thạc sĩ, ngành Khoa học máy tính, Trường Học viện Kỹ thuật Quân sự.
- Tóm tắt công việc hiện tại: Giảng viên khoa Công nghệ thông tin, Trường Đại học Sao Đỏ.
- Lĩnh vực quan tâm: Xử lý ảnh, học máy.
- Điện thoại: 0976940598 Email: htngat1985@gmail.com

Nguyễn Thị Ánh Tuyết

- Năm 2013: Tốt nghiệp Thạc sĩ, ngành Hệ thống thông tin, Trường Học viện Kỹ thuật Quân sự.
- Tóm tắt công việc hiện tại: Giảng viên khoa Công nghệ thông tin, Trường Đại học Sao Đỏ.
- Lĩnh vực quan tâm: Cơ sở dữ liệu phân tán, phân tích hệ thống thông tin.
- Điện thoại: 0972384332 Email: anhtuyet13381@gmail.com