Tên bài báo: Đánh giá: Kiến trúc cơ sở dữ liệu Big Data và NoSQL

Moko, Anasuodei

Khoa Khoa học Máy tính và Tin học,

Đại học Liên bang Otuoke,

Bang Bayelsa, Ni-giê-ri-a.

Asagba, Hoàng tử Oghenekaro

Khoa Khoa học Máy tính,

Đại học Cảng Harcourt,

Bang Rivers, Nigeria.

Trừu tượng:

Dữ liệu trong thời gian gần đây ngày càng lớn hơn để xử lý ngày càng tăng đặc biệt trong khối lượng, khiến dữ liệu phi cấu trúc cụ thể trở nên khó xử lý và quản lý, bởi vì cơ sở dữ liệu có cấu trúc gặp khó khăn khi xử lý dữ liệu phi cấu trúc do kích thước của nó, được gọi là Dữ liệu lớn. Do đó để hiểu được Dữ liệu lớn, kiến ​​trúc mới và các phương pháp là cần thiết, bài báo này đánh giá Dữ liệu lớn và các tính năng của nó, bốn NoSQL có sẵn kiến trúc để xử lý Dữ liệu lớn, điểm mạnh, điểm yếu và các loại có sẵn là thảo luận.

Từ khóa: Dữ liệu lớn,Phi cấu trúc, NoSQL, Cơ sở dữ liệu, Có cấu trúc

I. Giới thiệu

Một thách thức công nghệ lớn mà thế giới đang phải đối mặt là quản lý dữ liệu và lưu trữ, trong đó hàng triệu dữ liệu trong thời gian gần đây đang được tạo ra ở các khoảng thời gian ngắn hơn hơn nano giây. Do đó, việc xử lý một lượng lớn dữ liệu là một thách thức lớn và do đó, với sự gia tăng dân số, cần có dữ liệu tiên tiến nhất công nghệ quản lý và thu gom. (Sultana, et al, 2017) nhấn mạnh rằng nhu cầu xử lý và tạo dữ liệu nhanh chóng trong thời gian gần đây đã dẫn đến thực tế là hơn 2,6 nghìn tỷ dữ liệu đang được tạo ra

hằng ngày. Họ dự đoán thêm rằng trong tương lai sẽ có sự gia tăng dữ liệu theo cấp số nhân cách sử dụng và thế hệ trong tương lai ghi nhớ cách nó đang được sử dụng ngày nay. nêu rõ ràng rằng việc quản lý và lưu trữ dữ liệu trong thời đại dữ liệu lớn mà phần mềm thông thường đang bị đẩy đến giới hạn bởi lượng dữ liệu khổng lồ, tạo ra nhu cầu đáng kể thay đổi đầu tư, thu thập và lưu trữ dữ liệu để sử dụng trong tương lai, do đó tạo ra sự khác biệt các tổ chức nỗ lực có ý thức để bảo mật và giữ mọi phần dữ liệu tiềm năng. Tuy nhiên dữ liệu thường không có cấu trúc, nó có thể được tạo từ nhiều nguồn khác nhau, chẳng hạn như đăng lên mạng xã hội, nội dung đa phương tiện với kho lưu trữ tự động. Email, công cụ tìm kiếm truy vấn, kho lưu trữ tài liệu quản lý nội dung, dữ liệu cảm biến thuộc các loại khác nhau, chứng khoán trao đổi, ảnh vệ tinh, hệ thống giám sát và ứng dụng e-health.etc.

Đôi khi nó được hiểu là 'Không chỉ SQL', để nó có thể diễn đạt thực tế rằng các các công nghệ được sử dụng trong các ứng dụng web phân tán hàng loạt, bên cạnh dữ liệu quan hệ công nghệ. Quan trọng nhất, các công nghệ NoSQL được yêu cầu nếu tính sẵn sàng cao theo yêu cầu của dịch vụ web. Nó là một kiến ​​trúc lưu trữ được phân phối rộng rãi, có các chức năng cơ bản cấu trúc hệ thống quản lý cơ sở dữ liệu hữu ích. Dữ liệu thực tế của các giá trị chính được lưu trữ theo cặp, cột hoặc họ cột, tài liệu và đồ thị. Do đó, dự phòng khác nhau các khái niệm được hỗ trợ để đảm bảo tránh lỗi và tăng tính khả dụng của NoSQL các hệ thống cơ sở dữ liệu. (Phanxicô, 2019)

Đối với tất cả các phương pháp quản lý dữ liệu phi quan hệ, thuật ngữ NoSQL được sử dụng để đáp ứng tiêu chí liệt kê dưới đây:

Thứ nhất: Thông tin không được lưu trong bảng

Thứ hai: Ngôn ngữ SQL không phải là ngôn ngữ cơ sở dữ liệu.

Tuy nhiên, công nghệ cơ sở dữ liệu tương đối với công nghệ NoSQL cần được mở rộng sang tạo quyền truy cập toàn cầu không đổi đối với các dịch vụ mà nó cung cấp cho Web sâu rộng ứng dụng hoặc ứng dụng xử lý cơ sở dữ liệu lớn. (Ali, 2019)

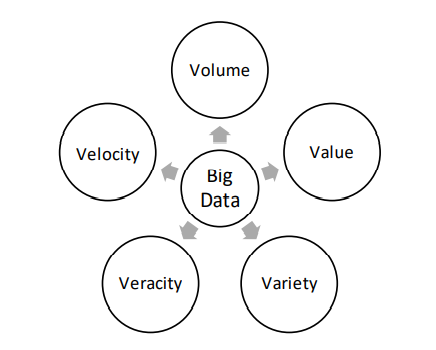
Hiểu công cụ "Dữ liệu lớn" trong khái niệm tạo dữ liệu là hữu ích, vì nó giúp lưu trữ và quản lý lượng thông tin vô tận được tạo ra mỗi giây, mỗi ngày. Mặc dù đã có các hệ quản trị cơ sở dữ liệu quan hệ (RDMS) cho dữ liệu cố định lưu trữ trong một số năm, khả năng mở rộng, tính nhất quán, hiệu quả hệ thống, thu thập dữ liệu và dữ liệu tích hợp trích xuất vẫn chưa được giải quyết.

Các công nghệ Dữ liệu lớn là các công cụ cho phép chúng ta hiểu được nhiều ý nghĩa hơn từ máy dữ liệu học tập và những thứ cho phép chúng tôi duy trì khối lượng dữ liệu cao hơn ở mức độ chi tiết cao hơn bao giờ hết. Các công nghệ Dữ liệu lớn này đã được Google đi tiên phong và họ đã tìm thấy cách dưới dạng Hadoop vào cộng đồng CNTT rộng lớn hơn. NoSQL hoạt động để giúp giải quyết các yêu cầu về khối lượng, sự đa dạng và tốc độ dữ liệu lớn (Andreas

& Michael, 2019) giải thích rằng Dữ liệu lớn vẫn chưa có định nghĩa ràng buộc. Tuy nhiên nhiều chuyên gia dữ liệu sẽ đồng ý về ba chữ V (3V): Khối lượng cho khối lượng dữ liệu lớn, Đa dạng nhiều định dạng, dữ liệu có cấu trúc, bán cấu trúc và phi cấu trúc và Tốc độ cao, tốc độ và xử lý dữ liệu thời gian thực.

(Nzar & Dashne 2019) đã phân loại các tính năng của Dữ liệu lớn thành ―5Vs‖: Dung lượng, tốc độ, sự đa dạng, tính xác thực và giá trị như mô tả dưới đây:

--------hình ảnh--------



Hình 1: Dữ liệu lớn 5 V (Nguồn: Nzar & Dashne, 2019)

1. Khối lượng: Hiển thị lượng dữ liệu khổng lồ, chẳng hạn như dữ liệu cho thiết bị di động, được sử dụng cho chức năng khác nhau.

2. Tốc độ: chỉ định tốc độ hoặc tần suất tạo, cập nhật, xử lý và truy cập vào dữ liệu.

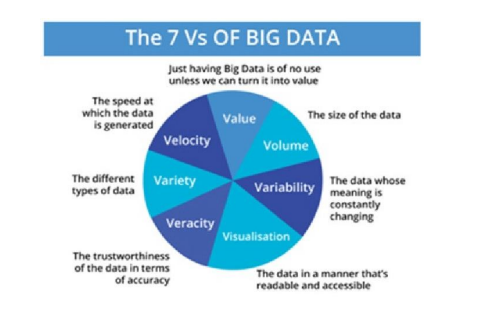
3. Đa dạng: dữ liệu được truy cập thông qua nhiều loại thiết bị khác nhau, chẳng hạn như video, ảnh, vân vân.

4. Giá trị: giải thích cách rút ra kiến ​​thức hữu ích từ các tập dữ liệu khổng lồ. nhiều nhất khía cạnh quan trọng của bất kỳ công cụ dữ liệu lớn nào là giá trị, vì nó cho phép tạo ra giá trị hiểu biết.

5. Tính xác thực: Đề cập đến độ chính xác và giá trị rất lớn của thông tin.

(Khan, et al, 2017) phân loại 1 - 6 v của dữ liệu lớn, dữ liệu này phát triển thành giá trị dữ liệu, khiến nó trở thành 7 V của dữ liệu lớn. Bảy chữ V là - Dung lượng, Tốc độ, Đa dạng, Độ biến thiên, Độ chính xác, Trực quan hóa và Giá trị.

---hình ảnh---



Hình 2: Bảy chữ V của Dữ liệu lớn (Nguồn: Syed et al, 2019)

II. Đánh giá các tác phẩm liên quan

Kalid et al (2017) Bài báo nêu bật tài liệu giải quyết các vấn đề và tìm giải pháp thông qua việc sử dụng các phương pháp tiếp cận Dữ liệu lớn tương phản để quản lý Không có SQL cơ sở dữ liệu; BigTable, DybanoDB và Cassandra. Tác phẩm cũng vạch ra tầm lớn đó các công ty không có hệ thống lưu trữ phù hợp không cần lưu trữ và quản lý số lượng lớn dữ liệu. Nghiên cứu cho thấy rằng BigTable của Google và DynamoDB của Amazon rất quan trọng và mang lại lợi nhuận cho chính họ và Casandra là sự kết hợp của cả hai hệ thống.

Md. Razu và cộng sự (2018) đã nghiên cứu cơ sở dữ liệu NoSQL để xử lý Dữ liệu lớn, bao gồm các vấn đề về giao dịch và cấu trúc. Công trình cũng nêu bật các hướng nghiên cứu và những thách thức liên quan đến việc xử lý Dữ liệu lớn mà nghiên cứu tin rằng thông tin thật tuyệt vời khi xem xét tài liệu về Cơ sở dữ liệu dữ liệu lớn NoSQL, bao gồm cấu trúc dữ liệu, vấn đề và phương pháp thu thập thông tin hữu ích được đo lường trong thời gian thực.

Vahid (2016) trong Dữ liệu lớn: Thỉnh thoảng, cho rằng sự tăng trưởng của dữ liệu và tri thức là phụ thuộc vào sự sẵn có của nó trong tay người tiêu dùng. Đơn giản thôi, như điện thoại di động, máy tính xách tay, PC, v.v. có thể dễ dàng truy cập. Trong trường hợp này, tri thức phát triển nhanh chóng tràn ngập—vì nội dung mới được tạo bởi chính người dùng nên nhu cầu quản lý, sử dụng, phân loại và bảo mật dữ liệu đó phải được đáp ứng. Bài báo cố gắng tổ chức và phân tích các quy trình, các vấn đề tiềm ẩn, nghiên cứu và sáng kiến ​​Dữ liệu lớn và hiển thị hiện tượng tăng mạnh này từ một chân trời triển vọng.

Ali và cộng sự (2019) đã gửi rằng NoSQL bị thay đổi bởi hầu hết các cơ sở dữ liệu quan hệ được sử dụng cho dữ liệu lưu trữ, nhưng nó không thay thế hoàn toàn SQL, như tên cho thấy. Giấy thảo luận về cơ sở dữ liệu SQL và NoSQL, sự tương phản của SQL thông thường với Big Data Analytics

Cơ sở dữ liệu NoSQL, mẫu dữ liệu NoSQL, biểu mẫu lưu trữ dữ liệu NoSQL, các tính năng và tính năng của từng kho lưu trữ dữ liệu, ưu nhược điểm của NoSQL và RDBMS.

III. Ngôn ngữ truy vấn có cấu trúc (SQL)

Mặt khác, SQL (Ngôn ngữ truy vấn có cấu trúc) theo truyền thống là ngôn ngữ nổi tiếng nhất cơ sở dữ liệu từ đầu những năm 1970. Một ví dụ là cơ sở dữ liệu quan hệ (được truy cập bằng cách sử dụng SQL

- ví dụ. MySQL) nơi dữ liệu được lưu trữ trong một bảng có hàng và cột. nhà phát triển trở lại sau đó triển khai chủ yếu các thiết kế của họ theo mô hình phát triển phần mềm thác nước.

Điều này có nghĩa là mọi giai đoạn phát triển phần mềm đều được lên kế hoạch tốt trước khiquá trình phát triển bắt đầu bằng cách sử dụng một mối quan hệ thực thể phức hợp toàn diện để đảm bảo rằng tất cả những gì đượccần thiết trong cơ sở dữ liệu đã được cân nhắc và cung cấp cẩn thận (Schaefer, 2015). Thậm chí mặc dù cơ sở dữ liệu quan hệ rất hữu ích, vẫn có những thách thức đối với phần mềm các nhà phát triển chẳng hạn như nếu có một cải tiến nhỏ trong chu kỳ phát triển của phần mềm, các nhà phát triển sẽ phải đấu tranh để giữ cho dự án dưới mức ngân sách quy định và phần mềm có thể không đáp ứng được nhu cầu của người dùng. Một lý do khác là đã có một sự đột biến trong lượng dữ liệu được tạo ra, đó là do một loại cơ sở dữ liệu mới NoSQL.

Một phương pháp quản lý mới được coi là cần thiết để hỗ trợ các ứng dụng như xem xét tệp nhật ký theo thời gian thực, giao dịch thương mại điện tử và dữ liệu được đăng lên phương tiện truyền thông xã hội khổng lồ về khối lượng. Một cách tiếp cận thay thế phải được đưa vào thực tế để xử lý vấn đề này sự gia tăng phi thường về dữ liệu được tạo ra và để vượt qua tất cả những thách thức nêu trên.

Cơ sở dữ liệu được giới thiệu, đó là cơ sở dữ liệu NoSQL, tuy nhiên có một số mức độ không hiệu quả do khối lượng sản xuất dữ liệu lớn và thiếu sự hỗ trợ cho

tính chất AXIT.

IV. Dữ liệu lớn và cơ sở dữ liệu

Dữ liệu lớn có thể được lưu trữ bằng cả Cơ sở dữ liệu có cấu trúc (MySQL cơ sở dữ liệu quan hệ) và Cơ sở dữ liệu phi cấu trúc (MongoDB một cơ sở dữ liệu phi quan hệ). Xem xét các sự thay đổi về thời gian phản hồi của từng loại cơ sở dữ liệu, các thuật toán khác nhau cần được phân tích để nâng cao hiệu suất trong việc giám sát hệ thống theo thời gian thực đối với cả SQL và cập nhật NoSQL và chèn dữ liệu lớn.

Các hệ thống cơ sở dữ liệu NoSQL được gọi là hệ thống lưu trữ dựa trên web miễn là chúng đáp ứng các yêu cầu sau:

 Mô hình: Mô hình cơ sở dữ liệu cơ bản không phải là quan hệ.

 Ít nhất 3 Vs: Lượng dữ liệu lớn (khối lượng), cấu trúc dữ liệu linh hoạt (đa dạng) và xử lý thời gian thực được đưa vào hệ thống cơ sở dữ liệu (tốc độ).

 Lược đồ: Một lược đồ cơ sở dữ liệu được thiết lập không bị ràng buộc bởi hệ thống quản lý cơ sở dữ liệu.

 Kiến trúc: Kiến trúc cơ sở dữ liệu hỗ trợ mở rộng theo chiều ngang và đầy đủ

ứng dụng web phân tán.

 Nhân bản: Hệ quản trị cơ sở dữ liệu hỗ trợ nhân bản dữ liệu.

 Đảm bảo tính nhất quán: tính nhất quán có thể được đảm bảo với độ trễ để ưu tiên cho tính sẵn sàng cao và dung sai của các phân vùng. (2019 Andreas & Michael)

V. Kiến trúc NoSQL dữ liệu lớn

So với 'Không phải SQL', NoSQL có nghĩa là 'Không chỉ SQL' vì nhiều người coi nó là một loại của cơ sở dữ liệu giúp thực hiện các hoạt động dữ liệu lớn và lưu trữ chúng ở định dạng hợp lệ. Nó là thường được sử dụng vì tính đơn giản và phạm vi dịch vụ rộng. Mô hình kiến ​​trúc của nó cung cấp một cách logic để dữ liệu được xử lý trên cơ sở dữ liệu. (Nzar&Dashne,2019)

Các tính năng NoQuery:

 Lược đồ miễn phí

 Cuối cùng nhất quán (như trong thuộc tính BASE)

 Sao chép kho lưu trữ dữ liệu để loại bỏ một điểm lỗi duy nhất.

 Có khả năng xử lý nhiều loại dữ liệu và khối lượng dữ liệu lớn

VI. Các mẫu kiến ​​trúc của NoSQL

Dữ liệu được lưu trữ trong NoSQL tuân theo bất kỳ mẫu nào trong bốn mẫu kiến ​​trúc dữ liệu.

a. Cơ sở dữ liệu lưu trữ khóa-giá trị

b. Cơ sở dữ liệu lưu trữ cột

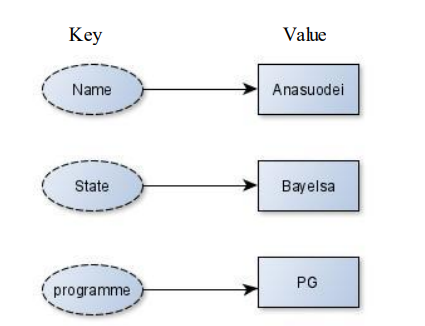
c. Cơ sở dữ liệu tài liệu

đ. Cơ sở dữ liệu đồ thị

A. Cơ sở dữ liệu lưu trữ khóa-giá trị

Một trong những mô hình cơ sở dữ liệu NoSQL cơ bản nhất là mô hình này. Dữ liệu được thu thập trong mẫu của các cặp khóa-giá trị, như tên ngụ ý. Một chuỗi các chuỗi, số nguyên hoặc ký tự là thường là khóa, nhưng nó cũng có thể là một dạng dữ liệu nâng cao hơn. Thông thường, giá trị là được kết nối hoặc đồng liên quan đến khóa. Cơ sở dữ liệu để lưu trữ cặp khóa-giá trị thường lưu trữ thông tin dưới dạng bảng băm trong đó mỗi khóa là duy nhất. Giá trị có thể ở bất kỳ dạng nào (Ký hiệu đối tượng JavaScript (JSON), Đối tượng lớn nhị phân (BLOB), chuỗi, v.v.). Phong cách này kiến trúc thường được sử dụng trong các trang web mua sắm hoặc ứng dụng thương mại điện tử và tài sản quan trọng là khả năng quản lý rộng rãi khối lượng dữ liệu, tải nặng và dễ dàng với những phím nào được sử dụng để truy xuất dữ liệu

---hình ảnh---



Hình 3. Một ví dụ về Key-Value

Các ràng buộc liên quan đến cơ sở dữ liệu lưu trữ khóa-giá trị là sự phức tạp của nó trong việc xử lý các truy vấn sẽ cố gắng bao gồm nhiều cặp khóa-giá trị có thể làm chậm đầu ra và có thể khiến dữ liệu xung đột với các mối quan hệ nhiều-nhiều.

Ví dụ ở đây là:

 DynamoDB (do Amazon phát triển)

 Berkeley DB (được phát triển bởi Oracle)

 REDIS: Kho lưu trữ khóa-giá trị nguồn mở nâng cao, còn được gọi là cấu trúc dữ liệu máy chủ vì các khóa có thể bao gồm chuỗi, giá trị băm, danh sách, bộ và bộ được sắp xếp. Đây

sản phẩm, được viết bằng C/C++, cực kỳ nhanh, khiến nó trở nên hoàn hảo cho dữ liệu

bộ sưu tập trong thời gian thực.

 Riak: Một nguồn mở mạnh mẽ, cơ sở dữ liệu phân tán có thể dự đoán quy mô

khả năng và đơn giản hóa việc tạo bằng cách tạo nguyên mẫu, phát triển và triển khai

các ứng dụng một cách nhanh chóng. Được viết bằng Erlang và C, công nghệ này cung cấp chức năng chịu lỗi/chuyển lỗi rõ ràng, một API toàn diện và linh hoạt, hoàn hảo cho hệ thống kiểm soát nhà máy và điểm bán hang

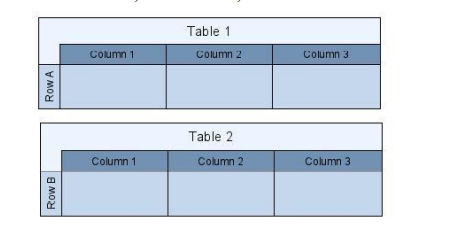
 VoltDB: cơ sở dữ liệu có thể mở rộng trong bộ nhớ cung cấp ACID giao dịch hoàn chỉnh tính nhất quán và thông lượng cực cao, được tự gọi là NewSQL. Đây công nghệ dựa vào phân đoạn và sao chép để đạt được dữ liệu có tính sẵn sàng cao ảnh chụp nhanh và ghi nhật ký lệnh lâu dài bằng cách sử dụng các quy trình được lưu trữ trong Java (đối với sự cố phục hồi), làm cho nó trở nên lý tưởng cho thị trường vốn, mạng kỹ thuật số, dịch vụ mạng và để chơi game trực tuyến.

B. Cơ sở dữ liệu lưu trữ cột

Mẫu này sử dụng lưu trữ dữ liệu trong các ô riêng lẻ được chia thành các cột, thay vì lưu trữ dữ liệu trong các bộ dữ liệu quan hệ. Cơ sở dữ liệu hướng cột chỉ hoạt động trên cột. Chúng cùng nhau lưu trữ một lượng lớn dữ liệu trong các cột. Định dạng cột và tiêu đề sẽ phân kỳ từ hàng này sang hàng khác. Mỗi cột được xử lý khác nhau, nhưng vẫn như vậy cơ sở dữ liệu thông thường, mỗi cột riêng lẻ sẽ chứa một số cột khác. (Niharika,2020)

Về cơ bản, các cột ở dạng chế độ lưu trữ này. Dữ liệu luôn sẵn có và có thể để thực hiện các truy vấn như Number, AVERAGE, COUNT trên các cột một cách dễ dàng.

---hình ảnh---



Hình 4. Một ví dụ về lưu trữ cột

Những trở ngại cho hệ thống này bao gồm: các giao dịch nên tránh hoặc không được hỗ trợ, các truy vấn có thể làm giảm hiệu suất cao với các phép nối bảng, ghi cập nhật và xóa giảm hiệu quả lưu trữ và có thể khó thiết kế các lược đồ phân vùng/lập chỉ mục hiệu quả.

Ví dụ ở đây là:

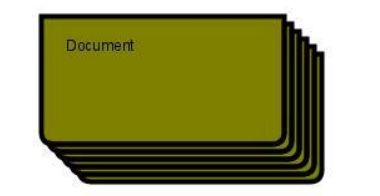
 HBase: HBase là Kho lưu trữ dữ liệu lớn, di động, phân tán theo mô hình của Google Công nghệ BigTable, cơ sở dữ liệu Hadoop.

 BigTable của Google

 Cassandra: Hệ quản trị cơ sở dữ liệu phân tán mã nguồn mở được xây dựng để quản lý khối lượng dữ liệu rất lớn nằm rải rác trên một số máy chủ mà không có một điểm duy nhất thất bại trong khi cung cấp một dịch vụ có khả năng tiếp cận cao. Được viết bằng Java, sản phẩm này là tốt nhất để phân tích dữ liệu thời gian thực phi giao dịch với khả năng mở rộng tuyến tính và đã được chứng minh khả năng chịu lỗi kết hợp với các chỉ số cột.

C. Cơ sở dữ liệu tài liệu

Ở dạng cặp khóa-giá trị, cơ sở dữ liệu bản ghi tìm nạp và tích lũy thông tin, nhưng ở đây các giá trị được gọi là tài liệu. Một cấu trúc dữ liệu phức tạp có thể được biểu diễn dưới dạng chữ. Tài liệu có thể ở dạng văn bản, mảng, chuỗi, JSON (Ký hiệu đối tượng JavaScript), XML (Ngôn ngữ đánh dấu mở rộng) hoặc bất kỳ định dạng nào khác. Việc sử dụng các tài liệu lồng nhau là vô cùng phổ biến. Nó có hiệu quả cao vì hầu hết các thông tin được tạo ra thường là ở dạng JSON và không có cấu trúc



Hình 5. Một ví dụ về tài liệu

Định dạng này cực kỳ hữu ích và phù hợp với dữ liệu bán cấu trúc và rất đơn giản để truy xuất và xử lý tài liệu từ kho lưu trữ. Những hạn chế liên quan đến hệ thống này bao gồm yếu tố thách thức của việc xử lý nhiều tài liệu và hoạt động không chính xác của các thao tác tổng hợp.

Ví dụ về các cơ sở dữ liệu như vậy là:

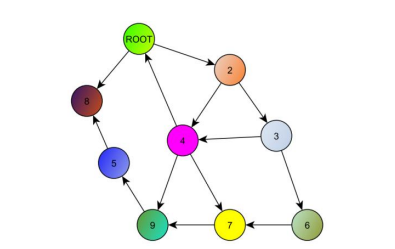
 MongoDB: Tính năng cơ sở dữ liệu NoSQL nguồn mở, hiệu suất cao, có thể mở rộng này lưu trữ hướng tài liệu (giống JSON), hỗ trợ chỉ mục đầy đủ, sao chép và cập nhật nhanh trên trang web từ "humongous". Sản phẩm này phù hợp cho các truy vấn động, cấu trúc dữ liệu động, được viết bằng C/C++ và nếu bạn ưu tiên lập chỉ mục hơn Bản đồ/Thu nhỏ.

 CouchDB: Cũng là một cơ sở dữ liệu nguồn mở tập trung vào việc dễ dàng lưu trữ dữ liệu trong một loạt tài liệu JSON, mỗi tài liệu có định nghĩa lược đồ riêng. cuối cùng tính nhất quán được thực thi bởi ngữ nghĩa ACID để ngăn khóa các tệp cơ sở dữ liệu trong khi viết. Sản phẩm này, được viết bằng Java, phù hợp cho các ứng dụng dựa trên web quản lý số lượng lớn dữ liệu được tổ chức lỏng lẻo.

D. Cơ sở dữ liệu đồ thị

Mẫu kiến ​​trúc này giải quyết rõ ràng việc lưu trữ và quản lý thông tin trong biểu đồ. Đồ thị về cơ bản là các cấu trúc biểu diễn quan hệ giữa hai hay nhiều đối tượng trong một số dữ liệu. Các đối tượng hoặc thực thể được gọi là các nút và được kết nối với các mối quan hệ đã biết như các cạnh. Có một mã định danh duy nhất trên mỗi cạnh. Đối với đồ thị, mỗi nút đóng vai trò là một điểm của cảm ứng. Trong các mạng xã hội nơi có nhiều và số lượng lớn các thực thể, mô hình này được sử dụng rất rộng rãi và mỗi thực thể có một hoặc nhiều đặc điểm được liên kết bởi các cạnh.

---hình ảnh---



Hình 6. Một ví dụ về đồ thị

Có các bảng được kết nối lỏng lẻo trong mẫu cơ sở dữ liệu quan hệ, trong khi các biểu đồ thường bản chất mạnh mẽ và cứng nhắc, có tốc độ truyền nhanh hơn do kết nối và cho phép dữ liệu không gian có thể dễ dàng xử lý, nhưng các kết nối không chính xác có thể dẫn đến các vòng lặp vô hạn. (Ian, 2016)

Ví dụ về các cơ sở dữ liệu như vậy là:

 Neo4J: Cơ sở dữ liệu đồ thị gốc và nền tảng đồ thị hàng đầu là Neo4J: Neo4j. Vì

mức độ bảo mật doanh nghiệp và hiệu suất cao và độ tin cậy bằng cách phân cụm, đó là có sẵn cả dưới dạng nguồn mở và thông qua giấy phép thương mại. Cypher, biểu đồ ngôn ngữ truy vấn của Neo4j, rất đơn giản để học và có thể sử dụng ứng dụng mở mới được phát hành bộ công cụ nguồn, "Cypher trên Apache Spark (CApS) và Cypher cho Gremlin để hoạt động trên các sản phẩm dựa trên Neo4j, Apache Spark và Gremlin."

 FlockDB (Được sử dụng bởi Twitter): FlockDB dễ dàng hơn các cơ sở dữ liệu đồ thị khác vì nó cố gắng giải quyết ít vấn đề hơn. Nó phù hợp theo chiều ngang và được tối ưu hóa cho trực tuyến, độ trễ thấp, môi trường thông lượng cao như các trang web.

 ArangoDB: loại cơ sở dữ liệu đồ thị này yêu cầu một cơ sở dữ liệu, Một ngôn ngữ truy vấn, Ba mô hình cho dữ liệu. Khả năng vô hạn. ArangoDB đang phát triển nhanh cơ sở dữ liệu NoSQL đa mô hình gốc, với hơn một triệu lượt tải xuống.

 OrientDB: OrientDB là đa mô hình DBMS phân tán đầu tiên có Biểu đồ thực

Người lái xe. Multi-Model có nghĩa là NoSQL thế hệ thứ 2 có khả năng quản lý

miền phức tạp với hiệu quả đáng kinh ngạc.

 Titan: Titan là cơ sở dữ liệu đồ thị có thể mở rộng được thiết kế để lưu trữ và truy vấn đồ thị trải rộng trên một cụm nhiều máy bao gồm hàng trăm tỷ đỉnh và cạnh. Titan là một cơ sở dữ liệu giao dịch có thể tạo điều kiện thuận lợi cho việc thực hiện theo thời gian thực của duyệt đồ thị phức tạp bởi hàng nghìn người dùng đồng thời.

 DataStax: Trong một môi trường thay đổi nhanh chóng, nơi có nhiều khát vọng, DataStax giúp các doanh nghiệp phát triển mạnh và các công nghệ mới xuất hiện hàng ngày.

 Amazon Neptune: Amazon Neptune với bộ dữ liệu được kết nối cao để tạo và chạy các ứng dụng an toàn, nhanh chóng và có dịch vụ cơ sở dữ liệu đồ thị được quản lý đầy đủ dễ. (Niharika, 2020).

VII. Điểm mạnh và điểm yếu của NOSQL

|  |  |  |
| --- | --- | --- |
| S/N | Điểm mạnh | Điểm yếu |
| 1 | Thích hợp, có sức mạnh để  lưu trữ và tra cứu dữ liệu lớn. | Yêu cầu một cách có thể hình dung được cơ sở hạ tầng đắt tiền |
| 2 | Ứng dụng có tập trung | Quá phức tạp |
| 3 | Hỗ trợ dữ liệu KHỔNG LỒ  dung tích | Tài năng kỹ thuật vẫn còn khó khăn  để tìm |
| 4 | Nhập dữ liệu nhanh | Nói chung, không có SQL  giao diện |
| 5 | Tốc độ tra cứu nhanh  (trên các cụm) | có lập trình hạn chế  giao diện |
| 6 | Cho phép truyền dữ liệu  và dữ liệu hiệu suất cao  máy chủ không cần thiết | Không phù hợp để phân tích  Truy vấn (tổng hợp,chỉ số, BI) |

VIII. Sự kết luận

Công trình này đã xem xét và nghiên cứu dữ liệu lớn trong thời gian gần đây đã và đang có cách xử lý ngày càng tăng đặc biệt trong dữ liệu khối lượng. Cơ sở dữ liệu có cấu trúc gặp khó khăn khi xử lý dữ liệu phi cấu trúc do kích thước của nó. Để hiểu được Dữ liệu lớn, kiến ​​trúc và phương pháp mới là cần thiết, công việc này cũng đã kiểm tra các loại Kiến trúc cơ sở dữ liệu NoQuery dữ liệu lớn khác nhau liên quan đến chúng, tầm quan trọng và cách sử dụng.

References

Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and

NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of

Research in Computer Science, 4(2), 1–10.

https://doi.org/10.9734/ajrcos/2019/v4i230108

Andreas Meier, & Michael Kaufmann. (2019). SQL & NoSQL Databases Models, Languages,

Consistency Options and Architectures for Big Data Management. Springer Vieweg.

https://doi.org/10.1007/978-3-658-24549-8

Best Graph Databases in 2020. (2020, May 27). G2. https://www.g2.com/categories/graphdatabases

Big Data and NoSQL Technologies | DB Best Chronicles. (2012, June 15). DB BEST.

https://www.dbbest.com/blog/big-data-nosql-technologies/

Column-Oriented Database Technologies | DB Best Chronicles. (2012, July 24). DB BEST.

https://www.dbbest.com/blog/column-oriented-database-technologies/

Francis, K. K. (2019). NoSQL Databases for Big Data Management: Review and Application

in Mobile Commerce. ResearchGate, 9. https://doi.org/10.13140/RG.2.2.10239.87204

Garrett, Alley. (2019, March 4). What Is Big Data Architecture? - DZone Big Data.

Dzone.Com. https://dzone.com/articles/what-is-big-data-architecture

Guy Harrison. (2015). Next Generation Databases NoSQL, NewSQL, and Big Data.

APRESS.

Ian. (2016, June 16). What is a Graph Database? | Database.Guide.

https://database.guide/what-is-a-graph-database/

Kalid, S., Syed, A., Mohammad, A., & Halgamuge, M. N. (2017). Big-data NoSQL

databases: A comparison and analysis of ―Big-Table‖, ―DynamoDB‖, and ―Cassandra‖.

2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(, 89–93.

https://doi.org/10.1109/ICBDA.2017.8078782

Khan, M. A., Uddin, M. F., & Gupta, N. (2014). Seven V’s of Big Data understanding Big

Data to extract value. Proceedings of the 2014 Zone 1 Conference of the American

Society for Engineering Education, 1–5.

https://doi.org/10.1109/ASEEZone1.2014.6820689

Le, J. (2019, November 20). An Introduction to Big Data: NoSQL. Medium.

https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-datanosql-96b882f35e50

Liyakathunisa Syed, Saima Jabeen, S. Manimala, & Hoda Ahmed Galal Elsayed. (2019,

January). (5) Data Science Algorithms and Techniques for Smart Healthcare Using IoT

and Big Data Analytics: Towards Smarter Algorithms. ResearchGate.

https://www.researchgate.net/publication/330723399\_Data\_Science\_Algorithms\_and\_Te

chniques\_for\_Smart\_Healthcare\_Using\_IoT\_and\_Big\_Data\_Analytics\_Towards\_Smarte

r\_Algorithms#fullTextFileContent

Md. Razu, A., Mst. Arifa, K., Md. Asraf, A., & Kenneth, S. (2018). A literature review on

NoSQL database for big data processing. International Journal of Engineering &

Technology, 7(2), 902–906. https://doi.org/10.14419/ijet.v7i2.12113

Niharika, P. (2020, January 2). NoSQL Data Architecture Patterns. GeeksforGeeks.

https://www.geeksforgeeks.org/nosql-data-architecture-patterns/

Raouf, D & Ali, N. (2019). Improving the performance of big data databases. Kurdistan

Journal of Applied Research. 4. 206-220. 10.24017/science.2019.2.20

Schaefer, L. (2015). What is NoSQL? Available at https://www.mongodb.com/nosqlexplained

The basics of NoSQL databases—And why we need them. (2019, January 31).

FreeCodeCamp.Org.https://www.freecodecamp.org/news/nosql-databases5f6639ed9574/

Vahid Rahmati. (2016). Big Data: Now and Then. International Journal of Emerging

Computing Methods in Engineering, 1(2), 1–6.

https://www.researchgate.net/publication/309458088\_Big\_Data\_Now\_and\_Then