**Nghiên cứu về cơ sở dữ liệu NoSQL**

Biswajeet Sethi , Samaresh Mishra, Prasant ku. Patnaik

School of Computer Engineering, KIIT University

Bhubaneswar, India

**Tóm tắt – Một số ứng dụng của dịch vụ mạng 2.0 cần xử lý dữ liệu lớn. Điều này đòi hỏi cơ sở dữ liệu quan hệ hiện có mở rộng quy mô theo chiều ngang để đạt được yêu cầu hiệu suất cao, đặc biệt đối với các ứng dụng yêu cầu cao quy mô dữ liệu của người dùng và có tính đồng thời cao. Những vấn đề này là sự cân nhắc quan trọng cho các nhà thiết kế đưa ra một nhóm cơ sở dữ liệu mới, thường được gọi là cơ sở dữ liệu NoSQL. Sự phát triển về nhu cầu điện toán đám mây và sự phát triển của Internet thúc đẩy phong trào NoSQL. Bài báo này đề cập đến các tính năng và mô hình dữ liệu của cơ sở dữ liệu NoSQL được sử dụng với độ mạnh và độ hạn chế của từng mô hình trong môi trường điện toán đám mây. Ngoài ra bài báo này nói về sự phân loại của cơ sở dữ liệu NoSQL dự trên định lý CAP .**

Từ khoá – NoSQL; CAP; Document Oriente; Column Family; Big Data.

I.GIỚI THIỆU

Lưu trữ và truy xuất dữ liệu hiệu quả với tính khả dụng và khả năng mở rộng là mục đích chính của cơ sở dữ liệu NoSQL. NoSQL không có nghĩa là không có SQL; nó có nghĩa là “không chỉ SQL”. Cơ sở dữ liệu NoSQL là chỉ một sự thay thế đến cơ sở dữ liệu truyền thống. Ngành cơ sở dữ liệu đã giới thiệu nhiều cơ sở dữ liệu phi quan hệ như MongoDB, Hbase, Neo4j trong vài năm gần đây. Tuỳ thuộc vào yêu cầu kinh doanh và chiến lược một nhà cung cấp đám mây có thể đi với bất kì loại cơ sở dữ liệu nào. Vẫn còn một số nhà thiết kế cơ sở dữ liệu phi quan hệ trước cho rằng cơ sở dữ liệu NoSQL không đủ hiệu quả trong việc xử lí tính toàn vẹn của dữ liệu. Bài báo này được tổ chức như sau: Phần 2 mô tả tầm quan trọng của cơ sở dữ liệu NoSQL. Phần 3 làm nổi bật lên mô hình dữ liệu NoSQL. Phần 4 làm nổi bật về giao dịch trong cơ sở dữ liệu NoSQL Phần 5 nêu rõ sự so sánh đối với cơ sở dữ liệu NoSQL. Cuối cùng, chúng tôi kết thúc bài báo trong phần 6.

II.TẦM QUAN TRỌNG CỦA CƠ SỞ DỮ LIỆU NOSQL

A.Lịch sử:

Trong vài năm qua, SQL so với NoSQL đã nổi lên một sự tranh cãi sôi nổi trên mạng xã hội. Sự tranh cãi “SQL so với NoSQL”, thực ra nói về cơ sở dữ liệu quan hệ so với cơ sở dữ liệu phi quan hệ. Bởi vì mô hình dữ liệu chuẩn hoá và thực thi các thuộc tính ACID nghiêm ngặt, cơ sở dữ liệu SQL truyền thống là được coi là cơ sở dữ liệu hướng giao dịch dựa trên lược đồ. Nó yêu cầu một lược đồ được xác định trước chính xác trước khi lưu trữ dữ liệu vào đó. Việc xác định lại lược đồ trong trường hợp có thay đổi trong tương lai, sau khi dữ liệu được đưa vào cơ sở dữ liệu sẽ bị gián đoạn. Trong khi đó, trong kỷ nguyên Dữ liệu lớn, nhu cầu bổ sung liên tục các loại dữ liệu mới để làm phong phú các ứng dụng. Một lần nữa, giải pháp lưu trữ của cơ sở dữ liệu SQL có thể tạo ra tác động lớn lên tốc độ và khả năng mở rộng. Các dịch vụ mạng như Amazon và Google có hàng terabyte và petabyte dữ liệu được lưu trữ trong trung tâm dữ liệu lớn của họ và phải đáp ứng các yêu cầu đọc-ghi lớn mà không có độ trễ đáng kể. Để mở rộng quy mô cơ sở dữ liệu SQL, dữ liệu cần được phân chia trên nhiều máy chủ. Trước khi cung cấp cho ứng dụng, thông tin mong muốn phải được thu thập từ nhiều bảng và kết hợp. Tương tự trong khi ghi dữ liệu cũng vậy; nó phải được thực hiện trên nhiều bảng một cách phối hợp. Đối với bất kỳ ứng dụng nào, nó có thể là kênh điều khiển để xử lý các bảng trên nhiều máy chủ. Trong cơ sở dữ liệu SQL, thao tác "liên kết" làm chậm hệ thống khi thu thập dữ liệu, đặc biệt khi hàng triệu người dùng đang thực hiện tra cứu đối với các bảng có hàng triệu hàng dữ liệu. Các dịch vụ web quy mô lớn như Google, Amazon, Yahoo, Facebook nhận thấy đây là những trường hợp cần phát triển cơ sở dữ liệu NoSQL của riêng họ để đáp ứng nhu cầu về khả năng mở rộng và hiệu suất.

B. Những tính năng của NoSQL

Cơ sở dữ liệu NoSQL có thể không yêu cầu lược đồ bảng được xác định trước, thường chia tỷ lệ theo chiều ngang và thường tránh các thao tác liên kết. Do lược đồ ít bản chất hơn và liên quan đến phân tích tập hợp con nhỏ hơn của hệ thống NoSQL, cơ sở dữ liệu này có thể được mô tả tốt hơn dưới dạng kho lưu trữ dữ liệu có cấu trúc. Ba tính năng cơ bản quan trọng của cơ sở dữ liệu NoSQL là mở rộng quy mô, cấu trúc dữ liệu linh hoạt và sao chép, được giải thích như sau.

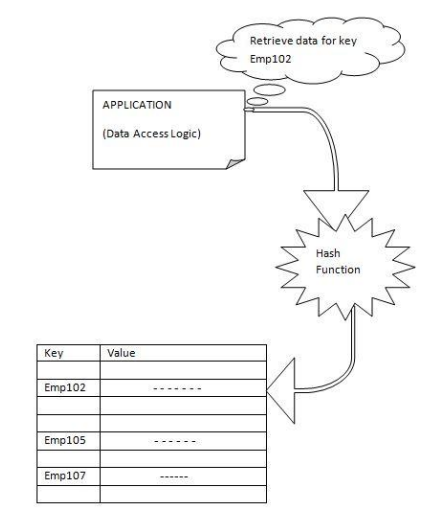
• Mở rộng quy mô: Mở rộng quy mô đề cập đến việc đạt được hiệu suất cao trong môi trường phân tán bằng cách sử dụng nhiều máy có mục đích chung. Cơ sở dữ liệu NoSQL cho phép phân phối dữ liệu trên một số lượng lớn máy có tải xử lý phân tán. Nhiều cơ sở dữ liệu NoSQL cho phép tự động phân bố dữ liệu tới các máy mới khi chúng được thêm vào nhóm. Mở rộng quy mô được đánh giá về khả năng mở rộng và tính đàn hồi.

• Tính linh hoạt: Tính linh hoạt về cấu trúc dữ liệu nói rằng không cần xác định lược đồ cho cơ sở dữ liệu. Cơ sở dữ liệu NoSQL không yêu cầu lược đồ được xác định trước. Điều này cho phép người dùng lưu trữ dữ liệu của các cấu trúc khác nhau trong cùng một bảng cơ sở dữ liệu. Tuy nhiên, hỗ trợ cho các ngôn ngữ truy vấn cấp cao như SQL không được hầu hết các cơ sở dữ liệu NoSQL hỗ trợ.

• Sao chép dữ liệu: Một trong những tính năng của cơ sở dữ liệu NoSQL là sao chép dữ liệu. Trong quá trình này, một bản sao của dữ liệu được phân phối cho các hệ thống khác nhau để đạt được sự dư thừa và phân phối vào. Tuy nhiên, có khả năng mất tính nhất quán dữ liệu giữa các bản sao. Nhưng người ta tin rằng đôi khi sự nhất quán này cuối cùng có thể đạt được. Tính nhất quán và tính sẵn sàng là yếu tố để đánh giá khả năng sao chép.

III. MÔ HÌNH DỮ LIỆU NOSQL

A. Cơ sở dữ liệu khóa-giá trị

Để xử lý truy cập đồng thời cao vào cơ sở dữ liệu, NoSQL được thiết kế là các nơi lưu trữ khóa-giá trị. Đây là cách đơn giản nhất nhưng vẫn là nơi lưu trữ dữ liệu mạnh mẽ nhất. Trong khóa-giá trị lưu trữ mỗi dữ liệu bao gồm một cặp khóa và giá trị duy nhất. Để lưu dữ liệu, một khóa được tạo bởi ứng dụng và giá trị được liên kết với khóa. Và cặp khóa-giá trị này được gửi đến kho lưu trữ dữ liệu. Các giá trị dữ liệu được lưu trữ trong các kho khóa giá trị có thể có các tập thuộc tính động được đính kèm với nó và không rõ ràng đối với hệ thống quản lý cơ sở dữ liệu. Do đó, khóa là công cụ duy nhất để truy cập các giá trị dữ liệu. Loại ràng buộc từ khóa đến giá trị phụ thuộc vào lập trình ngôn ngữ sử dụng trong ứng dụng. Một ứng dụng cần phải cung cấp chìa khóa cho các kho dữ liệu để truy xuất dữ liệu. Nhiều kho lưu trữ dữ liệu khóa-giá trị sử dụng hàm băm. Ứng dụng chia khóa và tìm ra vị trí của dữ liệu trong cơ sở dữ liệu. Các cơ sở dữ liệu khóa-giá trị được tập trung ở hàng cho phép ứng dụng truy xuất dữ liệu để hoàn thiện thực thể.

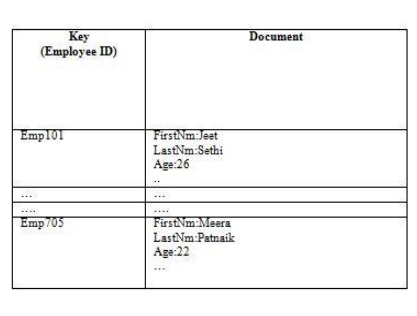
Hình.1 mô tả việc truy xuất dữ liệu từ cơ sở dữ liệu khóa-giá trị. Ứng dụng đã chỉ định khóa 'Emp102' cho kho lưu trữ dữ liệu để truy xuất dữ liệu. Sử dụng hàm băm, ứng dụng chia khóa để theo dõi vị trí của dữ liệu trong kho dữ liệu. Thiết kế của khóa nên hỗ trợ các truy vấn thường xuyên nhất được dùng trên kho lưu trữ dữ liệu. Hiệu quả của hàm băm, thiết kế khóa và kích thước của các giá trị được lưu trữ là các yếu tố ảnh hưởng đến hiệu suất của kho lưu trữ dữ liệu khóa giá trị. Các hoạt động được thực hiện trên các kho lưu trữ dữ liệu chủ yếu được giới hạn trong các hoạt động đọc và ghi. Bởi vì tính đơn giản của cơ sở dữ liệu khóa-giá trị nên nó cung cấp cho người dùng công cụ nhanh nhất để lưu trữ và tìm nạp dữ liệu. Tất cả các phần khác của NoSQL được xây dựng dựa trên sự đơn giản, khả năng mở rộng và hiệu suất của kho lưu trữ dữ liệu khóa-giá trị. Các hệ thống cơ sở dữ liệu Redis, Voldemort và Membase là những ví dụ về kho lưu trữ dữ liệu khóa-giá trị nổi bật nhất.

Hình.1. Ví dụ của cơ sở dữ liệu khóa - giá trị.

B. Cơ sở dữ liệu hướng tài liệu

Ở một mức độ trừu tượng cơ sở dữ liệu hướng tài liệu tương tự như cơ sở dữ liệu khóa-giá trị. Nó cũng giữ giá trị, ứng dụng có thể đọc hoặc tìm nạp bằng cách sử dụng khóa. Một số cơ sở dữ liệu tài liệu tự động tạo khóa duy nhất trong khi tạo một tài liệu mới. Tài liệu trong cơ sở dữ liệu tài liệu là một thực thể (tập hợp các trường được đặt tên). Một đặc điểm phân biệt cơ sở dữ liệu hướng tài liệu với cơ sở dữ liệu khóa-giá trị là tính minh bạch của dữ liệu được tổ chức bởi cơ sở dữ liệu. Do đó, khả năng truy vấn không bị hạn chế với khóa. Để hỗ trợ các tình huống mà ứng dụng yêu cầu truy vấn cơ sở dữ liệu không chỉ dựa trên khóa của nó mà còn với các giá trị thuộc tính, có thể chuyển đổi cho cơ sở dữ liệu tài liệu. Một tài liệu cần phải tự mô tả trong một tài liệu cơ sở dữ liệu định hướng. Thông tin được lưu trữ trong một thiết bị di động và định dạng tốt như XML, BSON hoặc JSON.

Như trong Hình 2, cơ sở dữ liệu tài liệu lưu trữ dữ liệu trong dạng cặp khóa-giá trị. Nhưng dữ liệu được lưu trữ trong cơ sở dữ liệu là minh bạch đối với hệ thống không giống như cơ sở dữ liệu khóa-giá trị. Ứng dụng có thể truy vấn cơ sở dữ liệu không chỉ bằng khóa tức là 'Employee ID' mà còn với các trường được xác định trong tài liệu tức là FirstNm, LastNm, tuổi, v.v. Cơ sở dữ liệu tài liệu là cách tiếp cận hiệu quả để mô hình hóa dữ liệu dựa trên các vấn đề phần mềm phổ biến. Nhưng cái giá phải trả là hiệu suất và khả năng mở rộng kém hơn so với cơ sở dữ liệu khóa-giá trị.Một vài cơ sở dữ liệu tài liệu nổi bật nhất như Riak, MongoDB [11], CouchDB.



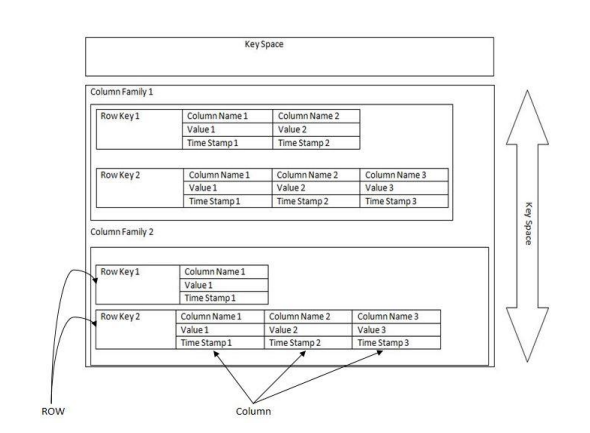
Hình.2. Ví dụ về cở sở dữ liệu hướng tài liệu.

C. Cơ sở dữ liệu họ cột

Đôi khi một ứng dụng có thể muốn đọc hoặc tìm nạp một tập hợp con của các trường, tương tự như phép chiếu của SQL. Cơ sở dữ liệu họ cột cho phép lưu trữ dữ liệu trong cột theo cách tiếp cận lấy cột trung tâm. Cở sở dữ liệu họ cột phân vùng các không gian khóa. Trong NoQuery một không gian khóa được coi là một đối tượng giữ tất cả các họ cột đã thiết kế lại với nhau. Đây là nhóm bên ngoài nhất của dữ liệu trong kho lưu trữ dữ liệu. Mỗi phân vùng của không gian khóa được hiểu là bảng. Nhóm các cột được khai báo bởi các bảng này. Với mỗi họ cột bao gồm số cột. Một hàng trong một họ cột là được cấu trúc như các tập hợp có số lượng cột bất kỳ. Mỗi cột là ánh xạ của cặp khóa-giá trị. Trong ánh xạ này, các khóa là tên của các cột và bản thân các cột là các giá trị . Mỗi ánh xạ này được gọi là một ô. Mỗi hàng trong cơ sở dữ liệu họ cột được xác định bằng một khóa hàng duy nhất, được xác định bởi ứng dụng. Việc sử dụng các khóa hàng này giúp cho việc truy xuất dữ liệu nhanh hơn. Để tránh ghi đè lên các giá trị ô, một số cơ sở dữ liệu họ cột phổ biến thêm dấu thời gian thông tin tự động đến các cột riêng lẻ. Mỗi lần có một bản cập nhật, nó tạo ra một phiên bản mới của các ô đã bị ảnh hưởng bởi hoạt động cập nhật. Người đọc luôn luôn đọc giá trị được ghi hoặc cam kết cuối cùng. Một khóa hàng, họ cột, cột và dấu thời gian tạo thành một khóa. Kể từ đây ánh xạ chính xác có thể được biểu diễn dưới dạng khóa. Do đó, ánh xạ có thể đại diện chính xác như

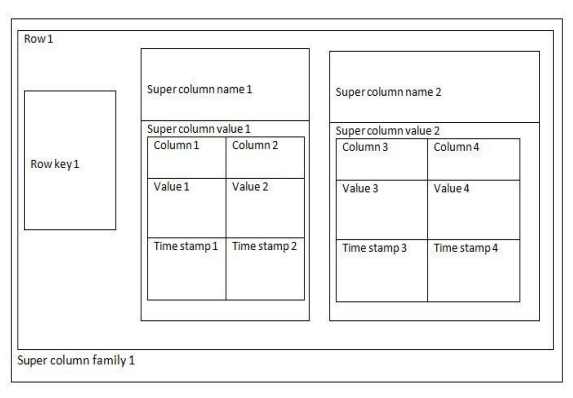
(khóa hàng, họ cột, cột, dấu thời gian) -> giá trị.

Cấu trúc tổng quát của cơ sở dữ liệu họ cột có được quan sát trong hình 3.



Hình 3. Lưu trữ dữ liệu họ cột.

Mô hình dữ liệu này đã được chấp nhận phổ biến là "thưa thớt, phân tán, bản đồ được sắp xếp đa chiều nhất quán " [4]. Lợi thế của việc sử dụng cơ sở dữ liệu họ cột trên một cơ sở dữ liệu truyền thống đang xử lý các giá trị NULL. Trong một cơ sở dữ liệu quan hệ, khi một giá trị cho một thuộc tính không áp dụng cho một hàng cụ thể, NULL sẽ được lưu trữ. Trong khi ở một cơ sở dữ liệu họ cột có thể được gỡ bỏ một cách đơn giản cho hàng tương ứng trong trường hợp không có dữ liệu. Đó là lý do tại sao Google gọi nó là cơ sở dữ liệu thưa thớt. Một trong những tính năng chính của cơ sở dữ liệu này là nó có thể được phân phối trong hàng tỷ ô trên hàng nghìn máy. Các ô được sắp xếp dựa trên khóa hàng. Sắp xếp các khóa cho phép tìm kiếm dữ liệu cho một loạt các khóa. Dữ liệu trong loại mô hình này được tổ chức dưới dạng tập hợp các hàng và cột, đại diện cho cơ sở dữ liệu này là gần giống nhất với cơ sở dữ liệu quan hệ. Nhưng giống như cơ sở dữ liệu quan hệ , nó không cần bất kỳ lược đồ được xác định trước nào. Trong thời gian chạy, các hàng và cột có thể được thêm vào một cách linh hoạt nhưng đôi khi họ cột phải được xác định trước, dẫn đến việc cơ sở dữ liệu kém linh hoạt hơn cơ sở dữ liệu tài liệu hoặc cơ sở dữ liệu khóa-giá trị. Các nhà phát triển nên hiểu dữ liệu được thu thập bởi ứng dụng và khả năng truy vấn có thể trước khi quyết định họ cột. Cơ sở dữ liệu họ cột được thiết kế tốt cho phép một ứng dụng đáp ứng phần lớn các truy vấn bằng cách duyệt qua ít họ cột nhất có thể. So sánh với cơ sở dữ liệu quan hệ chứa lượng dữ liệu tương đương, một cơ sở dữ liệu họ cột có khả năng mở rộng và nhanh hơn. Nhưng hiệu suất đi kèm với giá của cơ sở dữ liệu là ít tổng quát hơn một cơ sở dữ liệu quan hệ vì nó được thiết kế trong hỗ trợ cho một tập hợp các truy vấn cụ thể. Hệ thống cơ sở dữ liệu Hbase [9] và Hypertable dựa trên mô hình dữ liệu được mô tả ở trên. Trong khi một hệ thống cơ sở dữ liệu khác Cassandra khác từ mô hình dữ liệu, vì nó có một không gian mới được thêm vào được gọi là siêu cột [1]. Như thể hiện trong Hình 4, một siêu cột gồm nhiều cột. Một tập các siêu cột cùng với một khóa hàng tạo thành một hàng của nhóm siêu cột. Như trong các cột, tên siêu cột và tên cột phụ được sắp xếp. Siêu cột cũng là một thực thể tên-giá trị nhưng không có dấu thời gian.

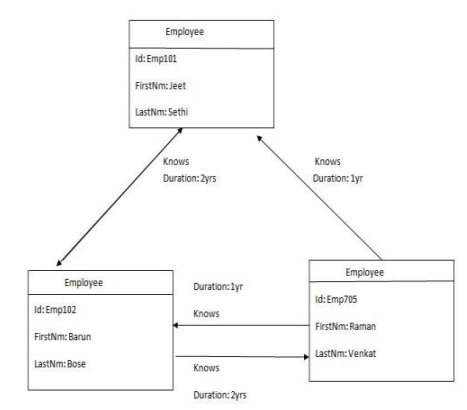


Hình 4. Lưu trữ dữ liệu họ cột (Cassandra).

D. Cơ sở dữ liệu đồ thị

Cơ sở dữ liệu đồ thị được coi là chuyên gia của dữ liệu có tính liên kết cao. Do đó, nó xử lý dữ liệu liên quan đến một số lượng lớn mối quan hệ [8]. Về cơ bản có ba lõi trừu tượng của cơ sở dữ liệu đồ thị. Có các nút, các cạnh mà kết nối hai nút khác nhau và thuộc tính. Mỗi nút giữ thông tin về một thực thể. Các cạnh đại diện cho sự tồn tại của mối quan hệ giữa các thực thể. Mỗi mối quan hệ có một loại mối quan hệ và được định hướng với một điểm bắt đầu (nút) và một điểm cuối. Điểm cuối có thể khác nút so với nút bắt đầu hoặc có thể là cùng một nút. Các thuộc tính khóa-giá trị không chỉ được liên kết với các nút mà còn với các mối quan hệ. Các thuộc tính của các mối quan hệ cung cấp thông tin bổ sung về các mối quan hệ. Hướng của mối quan hệ xác định đường đi từ nút này sang nút khác trong cơ sở dữ liệu đồ thị.

Hình 5 đại diện cho một phần của cơ sở dữ liệu 'Employee' có cấu trúc như cơ sở dữ liệu đồ thị. Mỗi nút trong cơ sở dữ liệu đồ thị này đại diện cho một thực thể nhân viên. Các thực thể này có quan hệ với nhau thông qua mối quan hệ của kiểu quan hệ “knows”. Thuộc tính gắn với quan hệ là "Duration". Sự khác biệt chính giữa biểu đồ và quan hệ cơ sở dữ liệu là truy vấn dữ liệu. Thay vì sử dụng nhiều chi phí quá trình như tham gia đệ quy như trong cơ sở dữ liệu quan hệ, biểu đồ cơ sở dữ liệu sử dụng phương pháp truyền tải. Trong khi truy vấn thông qua biểu đồ cơ sở dữ liệu, nút bắt đầu phải được chỉ định bởi ứng dụng. Quá trình truyền tải bắt đầu từ nút bắt đầu và tiến triển qua mối quan hệ với các nút được kết nối với nút bắt đầu, dựa trên một số quy tắc được xác định bởi logic ứng dụng. Phương pháp truyền tải chỉ liên quan đến các nút có liên quan đến ứng dụng không phải toàn bộ tập dữ liệu. Do đó, một sự gia tăng lớn trong số lượng nút không ảnh hưởng nhiều đến tốc độ truyền tải. Mạng xã hội, khai thác dữ liệu, quản lý mạng và tính toán các tuyến đường là một vài trong số các trường mà cơ sở dữ liệu đồ thị đã được sử dụng rộng rãi. Neo4j[8], GraphDB là những cơ sở dữ liệu đồ thị phổ biến được sử dụng hiện nay.



Hình 5. Ví dụ về cơ sở dữ liệu đồ thị

IV. GIAO DỊCH TRONG CƠ SỞ DỮ LIỆU NOSQL

Khi chúng ta nói về SQL và NoSQL, sự cạnh tranh thực sự không phải giữa cơ sở dữ liệu. Sự so sánh là giữa các mô hình của cơ sở dữ liệu. Giao dịch được định nghĩa là đơn vị logic của quá trình xử lý cơ sở dữ liệu được hình thành bởi một chương trình đang thực thi. Giao dịch của cơ sở dữ liệu SQL dựa trên các thuộc tính ACID nghiêm ngặt. ACID là từ viết tắt của Atomicity (tính nguyên tử), Consistence (tính nhất quán), Isolation (tính cô lập) và Durability (tính bền vững). Nhưng các nhà thiết kế cơ sở dữ liệu NoSQL đã đưa ra quyết định rằng các thuộc tính ACID quá hạn chế để đạt được yêu cầu của dữ liệu lớn. Do đó, giáo sư Eric Brewer vào năm 2000 đã đưa ra một định lý mới gọi là định lý CAP. CAP là chữ viết tắt của Consistency (tính nhất quán), Availability (tính khả dụng) và Partition (dung sai phân vùng). Định lý nói rằng các nhà thiết kế có thể đạt được bất kỳ hai thuộc tính trong số này cùng một lúc trong môi trường phân tán. Các nhà thiết kế có thể đảm bảo tính nhất quán và tính khả dụng với chi phí dung sai phân vùng, tức là cơ sở dữ liệu dựa trên CA. Nếu nhà thiết kế sử dụng tính khả dụng và dung sai phân vùng với chi phí của tính nhất quán, thì đó là cơ sở dữ liệu dựa trên AP. Và nếu đảm bảo tính nhất quán và dung sai phân vùng với chi phí khả dụng thì cơ sở dữ liệu dựa trên CP. Giao dịch của NoSQL có thể được phân loại như sau.

• Quan hệ về tính nhất quán và tính khả dụng (CA): Loại hệ thống cơ sở dữ liệu này đảm bảo mức độ ưu tiên của nó đối với tính khả dụng và tính nhất quán của dữ liệu bằng các sử dụng phương pháp sao chép. Một phần của cơ sở dữ liệu không bận tâm về sự phân vùng. Trong trường hợp xảy ra phân vùng giữa các nút, dữ liệu sẽ không đồng bộ. Cơ sở dữ liệu quan hệ, hệ thống cơ sở dữ liệu Vertica và Greenplum thuộc loại cơ sở dữ liệu này.

• Quan hệ về tính nhất quán và dung sai phân vùng (CP): Ưu tiên của hệ thống cơ sở dữ liệu đó là đảm bảo tính nhất quán của dữ liệu. Nhưng nó không hỗ trợ cho tính khả dụng. Dữ liệu được lưu trữ trong các nút phân tán. Khi một nút ngưng hoạt động, dữ liệu sẽ không khả dụng để duy trì tính nhất quán giữa các nút. Nó duy trì dung sai phân vùng bằng các ngăn đồng bộ hóa dữ liệu. HyberTable, BigTable, Hbase là một vài hệ thống cơ sở dữ liệu quan tâm đến CP.

• Quan hệ về tính khả dụng và dung sai phân vùng (AP): Ưu tiên của hệ thống cơ sở dữ liệu đó là đảm bảo tính khả dụng của dữ liệu và dung sai phân vùng là chủ yếu. Ngay khi có lỗi giao tiếp giữa các nút, các nút vẫn trực tuyến. Một lần sau khi phân vùng được giải quyết, quá trình đồng bộ hóa lại dữ liệu diễn ra nhưng không đảm bảo tính nhất quán. Riak, CouchDB, KAI là một vài cơ sở dữ liệu tuân theo nguyên tắt này.

Sau đó, định lý CAP được mở rộng thành PACELC. PACELC là viết tắt của Partition (phân vùng), Availability (tính khả dụng), Consistency (tính nhất quán), Else, Latency (độ trễ) và Consistency. Theo mô hình này, sự đánh đổi giữa tính khả dụng và tính nhất quán không chỉ dựa trên dung sai phân vùng mà còn phụ thuộc vào sự tồn tại của phân vùng mạng. Nó cho thấy độ trễ là một trong những yếu tố quan trọng, vì hầu hết các hệ thống cơ sở dữ liệu phân tán đều được sử dụng công nghệ sao chép để đảm bảo tính khả dụng.eBay sau đó đã giới thiệu một định lý mới được gọi là BASE. Nó nhằm mục đích đạt được tính khả dụng thay vì tính nhất quán của cơ sở dữ liệu. BASE là chữ viết tắt của Basically (cơ bản), Available (khả dụng), Soft (trạng thái mềm) và cuối cùng là Consistent (nhất quán).

• Khả dụng cơ bản: Khi một phần của cơ sở dữ liệu không khả dụng, các phần khác của cơ sở dữ liệu vẫn tiếp tục hoạt động. Trong trường hợp nút bị lỗi, hoạt động sẽ tiếp tục trên bản sao của dữ liệu được lưu trữ trong một số nút khác.

• Trạng thái mềm: Trên cơ sở tương tác của người dùng, dữ liệu có thể phụ thuộc vào thời gian. Những dữ liệu này cũng có thể hết hạn sau một khoảng thời gian nhất định. Do đó, để giữ cho dữ liệu có liên quan trong một hệ thống, nó phải được cập nhật hoặc truy cập.

• Nhất quán sau cùng: Sau bất kỳ cập nhật dữ liệu nào, nó có thể không trở nên nhất quán trên toàn bộ hệ thống nhưng sau cùng nó sẽ trở nên nhất quán theo thời gian. Do đó, dữ liệu được cho là nhất quán trong tương lai.

V. SO SÁNH CƠ SỞ DỮ LIỆU NoSQL

Không có quy tắc nghiêm khắc và chắc chắn nào để quyết định cơ sở dữ liệu NoSQL nào là tốt nhất cho doanh nghiệp. Mô hình kinh doanh, chiến luộc, chi phí và nhu cầu về mô hình giao dịch là một số yếu tố quan trọng mà doanh nghiệp nên cân nhắc khi lựa chọn cơ sở dữ liệu. Sau đây là một vài tiêu chí có thể giúp ích trong việc lựa chọn cơ sở dữ liệu cho doanh nghiệp.

• Nếu các ứng dụng chỉ lưu trữ và truy xuất các mục dữ liệu không rõ ràng đối với hệ thống quản lý cơ sở dữ liệu và các đối tượng lớn dưới dạng nhị phân bằng cách sử dụng khóa làm mã định danh, thì lưu trữ khóa-giá trị là sự lựa chọn tốt nhất. Nhưng nếu ứng dụng thích truy vấn cơ sở dữ liệu với một số giá trị thuộc tính khác với khóa, nó sẽ không thành công. Ngoài ra, trong khi cập nhật hoặc đọc một trường riêng lẻ trong lưu trữ khóa-giá trị bản ghi là một lỗi.

• Khi các ứng dụng có tính chọn lọc cao hơn và cần lọc các bản ghi dựa trên các trường không phải khóa, hoặc truy xuất hoặc cập nhật các trường riêng lẻ trong một bản ghi vì nó, thì lưu trữ tài liệu là một giải pháp hiệu quả. Kho lưu trữ cung cấp khả năng truy vấn tốt hơn so với kho lưu trữ dữ liệu khóa-giá trị.

• Khi các ứng dụng cần lưu trữ các bản ghi với hàng trăm hoặc hàng nghìn trường, nhưng truy xuất một tập hơn con của các trường đó trong hầu hết các truy vấn mà nó thực hiện, trong trường hợp đó lưu trữ dữ liệu lưu trên đĩa là một lựa chọn hiệu quả. Các kho lưu trữ dữ liệu như vậy phù hợp với các tập dữ liệu có quy mô lớn.

• Nếu các ứng dụng cần lưu trữ và xử lý thông tin trên dữ liệu được liên kết chặt chẽ với mối quan hệ phức tạp cao giữa các thực thể thì cơ sở dữ liệu đồ thị là lựa chọn tốt nhất. Trong cơ sở dữ liệu đồ thị, các thực thể và mối quan hệ giữa các thực thể được coi như nhau.

Bảng 1 trình bày danh sách các cơ sở dữ liệu, các mô hình dữ liệu tương ứng của chúng, cùng với mô hình giao dịch và ngôn ngữ truy vấn được sử dụng bởi các cơ sở dữ liệu này. Cassandra cho Facebook, Hbase cho Google, DynamoDB cho Amazon là một vài cơ sở dữ liệu được phát triển bởi các công ty khác nhau nhằm đáp ứng nhu cầu lưu trữ dữ liệu cao của họ. Mặt khác, các hệ thống cơ ở dữ liệu như Neo4j, Riak và MongoDB được phát triển để phục vụ các tổ chức khác. Về mô hình giao dịch, hầu hết các cơ sở dữ liệu như DynamoDB, Riak, Cassandra và Voldermort ưu tiên tính khả dụng hơn tính nhất quán. Trong khi Tokyo Cabinet, Hbase thích tính nhất quán hơn tính khả dụng. Cơ sở dữ liệu NoSQL được thiết kế để xử lý dữ liệu lớn, ngoại trừ một số hệ thống hỗ trợ của RDNMS như truy vấn. Mặc dù nhiều cơ sở dữ liệu NoSQL được đề cập trong Bảng 1, hỗ trợ cho việc truy vấn nhưng trình độ chuyên môn lập trình viết truy vấn cần cao hơn nhiều so với cơ sở dữ liệu quan hệ.

Bảng 1. So sánh khác nhau của cơ sở dữ liệu NoSQL

|  |  |  |  |
| --- | --- | --- | --- |
| Công cụ CSDL | Mô hình CSDL | Mô hình giao dịch | Truy vấn đặc biệt |
| DynamoDB | Khoá - Giá Trị | AP | Xây dựng trong API |
| Riak | Khoá - Giá Trị | AP | Corrugatedlron |
| Vodermort | Khoá - Giá Trị | AP | Không |
| Tokyo Cabinet | Khoá - Giá Trị | PC | Không |
| CouchDB | Tài liệu | AP | Cloudant, Lucene |
| MongoDB | Tài liệu | AP | Định dạng cơ sở BSON |
| RavenDB | Tài liệu | ACID | Xây dựng trong hạn chế |
| Cassandra | Cột – Họ | AP | HIVE, PIG |
| Hbase | Cột – Họ | PC | HIVE, PIG |
| Neo4J | Đồ thị | CA | Chyper |

V. KẾT LUẬN

Trong lĩnh vực cơ sở dữ liệu, cơ sở dữ liệu NoSQL được coi là khá mới. Tuy nhiên, những điều này đang được phát triển dựa trên lý thuyết đã biết và hiện có. Các hệ thống cơ sở dữ liệu NoSQL vẫn còn nhiều hạn chế. Không có tiêu chuẩn chung cũng như không có bất kỳ ngôn ngữ truy vấn phổ biển và quen thuộc nào để truy vấn cơ sở dữ liệu NoSQL. Mỗi cơ sở dữ liệu hoạt động theo một cách khác nhau và làm những việc khác nhau. Tương đối những cơ sở dữ liệu này là chưa trưởng thành và không ngừng phát triển. Cơ sở dữ liệu NoSQL không hỗ trợ các thuộc tính ACID nghiêm ngặt, do đó không có gì đảm bảo rẳng tất cả dữ liệu sẽ được ghi thành công vào kho lưu trữ dữ liệu. Bài báo này mô tả giới hạn của cơ sở dữ liệu quan hệ cùng với các loại mô hình dữ liệu NoSQL khác nhau. Vì không có sẵn đánh giá để tìm ra công cụ phù hợp, bài viết này so sánh sức mạnh và hạn chế của từng mô hình dữ liệu. Hạn chế của cơ sở dữ liệu NoSQL và việc sử dụng nó trong điện toán đám mây là những lĩnh vực cần được nghiên cứu chi tiết trong tương lai.

REFERENCES

[1] Maria Indrawan, “Database Research: Are We At A Crossroad?,” 15th International Conference on Network-Based Information Systems, pp. 45-48, 2012.

[2] Jing Han, Haihong E, Guan Le,Jian Du, “Survey on NoSQL Database,” IEEE, pp. 363- 366, 2011.

[3] Shalini R., Savita G., Subramanian A., “Comparison of Cloud Database: Amazon‟s SimpleDB and Google‟s Bigtable,” International Conference on Recent Trends in Information Systems, IEEE, pp. 165- 168, 2011.

[4] R Hecht, S Jablonski, “NoSQL Evaluation,” International Conference on Cloud and Service Computing, IEEE, pp. 336-338, 2011.

[5] Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin, “MongoDB vs Oracle - database comparison,” Third International Conference on Emerging Intelligent Data and Web Technologies, 2012.

[6] Jing Han, Meina Song and Junde Song, “A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing,” 10th IEEE/ACIS International Conference on Computer and Information Science, 2011.

[7] Guoxi Wang,Jianfeng Tang, “The NoSQL Principles and Basic Application of Cassandra Model,” IEEE, pp.1332-1333, 2012.

[8] Neo4j, <http://neo4j.org>.

[9] Hbase, <http://hbase.apache.org>.

[10] Mahdi Negahi Shirazi,Ho Chin Kuan,Hossein Dolatabadi, “Design Patterns to Enable Data Portability between Clouds‟ Databases,” 12th International Conference on Computational Science and Its Applications, pp. 117-118, 2012.

[11] Mongodb, <http://www.mongodb.org>.