III. TÍNH NĂNG CỦA RELATIONAL DATABASES VÀ MÔ HÌNH CƠ SỞ DỮ LIỆU NOSQL

1. Mã nguồn đóng và mã nguồn mở

Relational Databases bao gồm các mã nguồn mở và độc quyền[5]. Những loại Relational Databases độc quyền như Oracle có khả năng mở rộng quy mô hơn so với các đối tác mã nguồn mở như MySQL. Tuy nhiên, rất nhiều mô hình cơ sở dữ liệu NoSQL là mã nguồn mở ví dụ như MongoDB, CouchDB và Cassandra [9]. Bản chất mã nguồn mở của NoSQL mang lại nhiều cơ hội hơn cho các nhà nghiên cứu trong việc tìm hiểu các tính năng của cơ sở dữ liệu và cung cấp dung lượng lưu trữ rẻ hơn cho những người dùng không có khả năng mua các mô hình cơ sở dữ liệu độc quyền.

1. Khả năng mở rộng

Relational Databases, thường mở rộng quy mô, phải nâng cấp phần cứng máy tinh cho máy chủ để cho nó chạy hiệu quả hơn . Điều này khiến cho các quản trị viên cần phải nổ lực hơn trong việc nâng cấp Relational Databases [1]. Phương pháp nâng cấp này đối mặt với khó khăn về giới hạn phần cứng được thiết kế cố định và không thể thay đổi. Ví dụ: dung lượng RAM tối đa hoặc dung lượng lưu trữ của phần cứng có giá trị không dổi do nhà sản xuất quy định. Nghĩa là Relational Databases có khả năng mở rộng nhưng sẽ có giới hạn về mức độ mở rộng do nó được xác định bởi phần cứng. Để cung cấp khả năng mở rộng, NoSQL yêu cầu sử dụng máy chủ hàng hóa, mở rộng theo chiều ngang [8] [10]. Việc mở rộng quy mô theo chiều ngang không bị ảnh hưởng bởi các giới hạn phần cứng vì các máy chủ nhỏ hơn, rẻ hơn và yếu hơn có thể được kết hợp để mang lại khả năng mở rộng quy mô to hơn thay vì có một máy chủ đắt tiền. Điều đó giúp việc thực hiện trở nên dễ dàng vì các máy ảo có thể được sử dụng làm máy chủ hàng hóa trong tình huống không có phần cứng thực tế. Máy ảo có thể được thêm và xóa mà không gây ảnh hưởng đến cơ sở dữ liệu. Các ứng dụng internet hiện đại như phương tiện truyền thông yêu cầu mở rộng quy mô to mà Relational Databases không đáp ứng được nhưng được cung cấp trong NoSQL [16].

C. Chi phí

Relational Databases tốt nhất là độc quyền và do đó, đòi hỏi số tiền đầu tư đáng kể từ các tổ chức và cá nhân muốn hưởng lợi từ các tính năng nâng cao của chúng. Phần cứng bổ sung để nâng cấp cũng thêm các chi phí khác. Điều này làm cho Relational Databases trở thành một phương pháp lưu trữ dữ liệu đắt tiền [7]. NoSQL chủ yếu là mã nguồn mở, làm cho nó trở thành một giải pháp thay thế rẻ hơn cho Relational Databases [9]. Khả năng sử dụng máy ảo làm máy chủ hàng hóa giúp giảm hơn nữa chi phí duy trì cơ sở dữ liệu NoSQL, khiến NoSQL trở thành kho lưu trữ dữ liệu chi phí thấp hấp dẫn cho các tổ chức.

D. Khối lượng và sự đa dạng của dữ liệu

Các ứng dụng Internet làm tăng lượng dữ liệu mà cơ sở dữ liệu phải xử lý [11]. Internet đã chứng kiến sự xuất hiện của web 2.0 và 3.0 đã làm tăng số lượng và sự đa dạng của dữ liệu phải được lưu trữ. Sự xuất hiện của dữ liệu lớn cũng đã làm tăng số lượng và sự đa dạng của dữ liệu. Relational Databases không thể xử lý khối lượng lớn dữ liệu từ các nguồn này. NoSQL vượt trội trong việc xử lý lượng dữ liệu lớn làm cho nó phù hợp với các ứng dụng internet sử dụng nhiều dữ liệu [12]. Điều này có thể thấy ở các công ty như Google, Facebook và Yahoo đã chuyển sang NoSQL [3].

E. Khả dụng

Số lượng người dùng và thời gian truy cập dữ liệu đã tăng lên, với các ví dụ như phương tiện truyền thông xã hội, thương mại điện tử và lưu trữ đám mây đang dẫn đầu. Theo thiết kế, Relational Databases thường bị lỗi ngay cả đối với các máy chủ rất mạnh [5]. Tính khả dụng bị hạn chế hơn nữa do mở rộng Relational Databases. Một điểm thất bại duy nhất là không phù hợp với các ứng dụng internet hiện đại ngày nay mà người dùng đặt niềm tin lớn vào để hỗ trợ họ trong cuộc sống hàng ngày. Do đó, bản chất phân tán của NoSQL làm cho nó trở thành lựa chọn tốt hơn để luôn cung cấp cho người dùng tính khả dụng ngay cả trong trường hợp lỗi phần cứng [10]. Các nguyên tắc cơ bản tích hợp sẵn của NoSQL giúp truy cập các phần của cơ sở dữ liệu trong trường hợp có lỗi. Bất kể lỗi hệ thống, người dùng được đảm bảo tiếp tục truy cập vào cơ sở dữ liệu.

F. Hiệu suất

Relational Databases cần nhiều thời gian hơn để xử lý thông tin, khiến chúng chậm hơn so với NoSQL xử lý nhanh [13]. Hiệu suất của NoSQL được cải thiện hơn nữa vì nó truy xuất dữ liệu từ bộ nhớ khả biến, không giống như Relational Databases truy xuất dữ liệu từ bộ nhớ không khả biến. Theo thiết kế, bộ nhớ khả biến nhanh hơn bộ nhớ không khả biến. Trong các ứng dụng tìm kiếm trên Internet, NoSQL đã vượt trội so với cơ sở dữ liệu quan hệ trong việc truy xuất thông tin [14]. Các thử nghiệm đã được tiến hành để kiểm tra hiệu suất của NoSQL và Relational Databases. So sánh cơ sở dữ liệu quan hệ với MongoDB cho thấy MongoDB có hiệu suất tốt hơn khi đọc, cập nhật và truy vấn cơ bản, trong khi SQL chỉ hoạt động tốt khi cập nhật các thuộc tính không phải khóa [15].

G. Tính phức tạp

Relational Databases tạo ra dữ liệu phức tạp trong đó dữ liệu do người dùng lưu trữ khó chuyển đổi thành bảng [1]. Việc nhấn mạnh vào việc lưu trữ dữ liệu có cấu trúc trong Relational Databases mang lại sự phức tạp này. Khi một số thao tác đọc hoặc ghi đơn giản có thể đủ, chẳng hạn như trong phương tiện truyền thông xã hội, các truy vấn và giao dịch phức tạp của Relational Databases có thể không cần thiết. NoSQL có thể lưu trữ dữ liệu bán cấu trúc và phi cấu trúc [16]. Khả năng lưu trữ dữ liệu bán cấu trúc và phi cấu trúc của NoSQL mang lại sự linh hoạt cần thiết để hỗ trợ các loại dữ liệu khác nhau ở trạng thái ban đầu mà không làm mất thông tin. Ví dụ: chuyển đổi bản ghi âm khiếu nại của khách hàng thành văn bản để lưu trữ trong Relational Databases dẫn đến mất thông tin về cảm nhận của khách hàng. Những thông tin như vậy có thể được lưu giữ trong NoSQL vì các bản ghi có thể được lưu trữ ở trạng thái của chúng mà không cần chuyển đổi.

H. Ngôn ngữ truy vấn

Relational Databases có nền tảng vững chắc và được ghi chép đầy đủ trên SQL. SQL là ngôn ngữ thao tác dữ liệu duy nhất được sử dụng bởi tất cả các Relational Databases [5]. Tuy nhiên, việc triển khai SQL hơi khác nhau đối với các cơ sở dữ liệu quan hệ khác nhau được sử dụng. Nền tảng vững chắc do SQL cung cấp đã làm cho Relational Databases trở nên phổ biến đối với các nhà phát triển vì quá trình học tập để triển khai bất kỳ Relational Databases là tương đối ngắn. NoSQL vẫn thiếu nền tảng này vì nó dựa vào API hướng đối tượng để thao tác dữ liệu [1]. Mỗi triển khai của NoSQL có ngôn ngữ thao tác dữ liệu riêng, đòi hỏi các nhà phát triển phải dành thời gian học hỏi khi phát triển một loại mô hình NoSQL khác với những gì họ quen thuộc. Có nhiều cách để truy vấn NoSQL, giới hạn số lượng truy vấn được hỗ trợ do mỗi lần triển khai phải cung cấp các truy vấn duy nhất của riêng nó [17]. Các yêu cầu của web 2.0 và 3.0 yêu cầu các phương pháp phát triển nhanh, mà NoSQL có thể không đáp ứng được do thời gian phát triển tăng lên vì các nhà phát triển cần học ngôn ngữ triển khai.

I. Tính nhất quán

Relational Databases cung cấp tính nhất quán mạnh mẽ hơn thông qua các lược đồ nghiêm ngặt [8]. Tính năng này làm cho Relational Databases hy sinh khả năng sử dụng vì không miễn phí. Tính nhất quán cao rất tốt để cung cấp chế độ xem dữ liệu nhất quán ngay sau khi một hành động được thực hiện. Tuy nhiên, trong các ứng dụng như phương tiện truyền thông xã hội, tính linh hoạt quan trọng hơn tính nhất quán [16]. NoSQL cung cấp tính khả dụng cao hơn nhưng tính nhất quán kém hơn [7]. Vì vậy, đối với phương tiện truyền thông xã hội, NoSQL là tùy chọn lưu trữ tốt hơn so với Relational Databases.

J. Bảo mật

Relational Databases phải đối mặt với nhiều thách thức về bảo mật, chẳng hạn như SQL injection( tác động đến cơ chế có sẵn và lợi dụng nó) và cross-site scripting(tấn công bằng mã độc rồi mạo danh người dùng và ăn cắp dữ liệu). Bất chấp những thách thức này, SQL có các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu, bao gồm xác thực, ủy quyền, mã hóa, toàn vẹn và kiểm tra [1]. Cơ chế bảo mật là một phần của cơ sở dữ liệu. Trong NoSQL, bảo mật không phải là một phần của cơ sở dữ liệu mà được xử lý bởi phần mềm trung gian [7]. Điều này khiến cơ sở dữ liệu dễ bị tấn công. Ngoài ra, các cơ chế bảo mật được triển khai trong phần mềm trung gian phải được triển khai theo cách không ảnh hưởng đến khả năng mở rộng và hiệu suất.

Bảng 1

SO SÁNH RELATIONAL DATABASES VÀ NoSQL

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Relational Database | NoSQL |
| Sự đa dạng | Nền tảng mã nguồn đóng và mã nguồn mở[5] | NoSQL chủ yếu là mã nguồn mở [9] |
| Khả năng mở rộng quy mô | Mở rộng quy mô bằng cách nâng cấp phần cứng của máy chủ [1]. | Mở rộng quy mô với máy chủ hàng hóa [8] |
| Chi phí | Cách tốn kém để lưu trữ dữ liệu [7] | Rẻ hơn vì nó là mã nguồn mở và rẻ để nâng cấp [9] |
| Khối lượng dữ liệu | Xử lý dữ liệu hạn chế [11]. | Xử lý dữ liệu khối lượng lớn, đặc biệt là dữ liệu lớn [12]. |
| Khả dụng | Chịu một điểm thất bại [5] | Bản chất phân tán luôn cung cấp cho người dùng tính khả dụng trong trường hợp lỗi phần cứng [10] |
| Hiệu suất | Cần thêm thời gian để xử lý thông tin làm chúng chậm lại [13] | Có xu hướng có hiệu suất truy vấn tốt hơn [16] |
| Tính phức tạp | Tạo dữ liệu phức tạp trong trường hợp người dùng lưu trữ dữ liệu khó chuyển thành bảng [1] | Lưu trữ dữ liệu bán cấu trúc và phi cấu trúc ít phức tạp hơn [16] |
| Ngôn ngữ truy vấn | SQL là ngôn ngữ thao tác dữ liệu duy nhất được sử dụng bởi tất cả cơ sở dữ liệu quan hệ, với cách triển khai hơi khác nhau [5] | Mỗi lần thực hiện NoSQL có ngôn ngữ thao tác dữ liệu riêng [19] |
| Tính nhất quán | Có tính nhất quán cao với lược đồ nghiêm ngặt [8] | Ít nhất quán hơn với các phương pháp không có lược đồ [7] |
| Bảo mật | Có các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu [1] | Để bảo mật được xử lý bởi phần mềm trung gian và không phải là một phần của cơ sở dữ liệu [7] |