**Một nghiên cứu so sánh: MongoDB vs MySQL**Mr. Sushil Soni ,Mr. Mayuresh Ambavane ,Mr. Shamal Ambre , Mr. Shirshendu Maitra

**Tóm tắt**

Cơ sở dữ liệu quan hệ đã là nền tảng của các ứng dụng doanh nghiệp trong nhiều thập kỷ và kể từ khi MySQL được phát hành trong 1995 nó đã là một nguồn tài nguyên phổ biến và rẻ tiền. Tuy nhiên, với sự bùng nổ về khối lượng và sự đa dạng của dữ liệu, gần đây các công nghệ cơ sở dữ liệu phi quan hệ như MongoDB đã xuất hiện để giải quyết nhu cầu của các ứng dụng mới. MongoDB không chỉ được sử dụng cho các ứng dụng mới mà còn để tăng cường hoặc thay thế cơ sở hạ tầng quan hệ hiện có.

Trong bài báo này, chúng tôi sẽ cố gắng trình bày trường hợp một nghiên cứu so sánh về cơ sở dữ liệu phi quan hệ và cơ sở dữ liệu quan hệ. Chúng tôi chủ yếu nhấn mạnh bài thuyết trình của chúng tôi về một ứng dụng của công nghệ cơ sở dữ liệu NoSQL, được biết đến là MongoDB và so sánh với một ứng dụng khác của cơ sở dữ liệu quan hệ, được biết đến là MySQL, và do đó biện minh cho lý do tại sao MongoDB hiệu quả hơn MySQL.Chúng tôi cũng sẽ trình bày những lợi ích của việc sử dụng cơ sở dữ liệu phi quan hệ so với cơ sở dữ liệu quan hệ. Một tiêu chí so sánh bao gồm sự khác biệt về lý thuyết, đặc điểm, giới hạn, tính toàn vẹn, phân phối, yêu cầu hệ thống và kiến trúc, truy vấn và thời gian chèn.

**1.GIỚI THIỆU**

Một vài năm trở lại đây, một ứng dụng bình thường từng chỉ có hàng nghìn người dùng đến hàng chục hàng ngàn người dùng trong trường hợp cao nhất,  ngày nay có những ứng dụng có hàng triệu người dùng và những người đã kết nối cả ngày lẫn đêm, năm này qua năm khác.Việc quan trọng là sử dụng một cơ sở dữ liệu thích hợp, hỗ trợ kết nối đồng thời hàng trăm nghìn người dùng.

Cơ sở dữ liệu quan hệ được sử dụng trên toàn cầu trong hầu hết các ứng dụng và chúng có hiệu suất tốt khi họ xử lý một lượng dữ liệu hạn chế. Để xử lý một khối lượng lớn dữ liệu như internet, đa phương tiện và phương tiện truyền thông xã hội, việc sử dụng truyền thống cơ sở dữ liệu quan hệ không hiệu quả. Để khắc phục vấn đề này, thuật ngữ "NO SQL" đã được giới thiệu.Thuật ngữ NoSQL được Carlo Strozzi sử dụng vào năm 1998 và đề cập đến cơ sở dữ liệu phi quan hệ, thuật ngữ mà sau đó đã được giới thiệu lại vào năm 2009 bởi EricEvans.Ngày nay, thuật ngữ này đã nhận được một nghĩa khác, cụ thể là "Not Only SQL",đó là một biến thể khoan dung của việc xác định thuật ngữ, so với ý nghĩa trước đây của nó, phản quan hệ.

NoSQL, không phải là một công cụ, mà là một hệ phương pháp bao gồm một số công cụ phụ thuộc và cạnh tranh nhau. Lợi ích chính của NoSQL cơ sở dữ liệu là, không giống như một cơ sở dữ liệu quan hệ, nó có thể xử lý dữ liệu phi cấu trúc như tài liệu, email, đa phương tiện và phương tiện truyền thông xã hội hiệu quả. Cơ sở dữ liệu phi quan hệ không sử dụng Nguyên tắc RDBMS (Cơ sở dữ liệu quan hệ Hệ thống quản lý) và không lưu trữ dữ liệu trong bảng, lược đồ không cố định và rất đơn giản mô hình dữ liệu. Thay vào đó, họ sử dụng khóa nhận dạng và dữ liệu có thể được tìm thấy từ các khóa được chỉ định. Có bốn chiến lược để lưu trữ dữ liệu trong một cơ sở dữ liệu phi quan hệ, như đã được hiển thị, và những chiến lược đó như sau:

1. Key-Value - Cơ sở dữ liệu khóa-giá trị là từ điển phân phối khái niệm và không có một lược đồ được xác định trước; chúng là lược đồ ít hơn, khóa có thể là tổng hợp hoặc tự tạo và giá trị có thể là bất cứ thứ gì: chuỗi, JSON, BLOB và các loại khác.

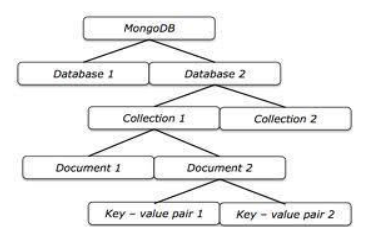
2. Document - MongoDB là cơ sở dữ liệu dựa trên tài liệu phổ biến nhất .Họ linh hoạt trong loại nội dung vì chúng không có lược đồ được xác định trước. Về mặt khái niệm, chúng hoạt động tốt với các tài liệu thuộc nhiều loại khác nhau như JSON, BSON, XML và BLOBs. Về cơ bản họ chỉ đại diện cho một chuyên môn hóa khóa-giá trị cơ sở dữ liệu. Một tài liệu được viết hoặc đọc bằng cách sử dụng chìa khóa, Bên cạnh phạm vi khả năng Key-Value, cơ sở dữ liệu dựa trên tài liệu thêm khác nhau cơ hội tìm tài liệu dựa trên nội dung

3. Column/ Field – Cơ sở dữ liệu từ BigTable danh mục, như HBase và Hypertable là cột nhập và phải có một giản đồ được xác định trước. Dữ liệu được lưu trữ trong các ô được nhóm trong các cột và các cột được nhóm hợp lý thành các nhóm cột. Theo giả thuyết, chúng có thể chứa số lượng không giới hạn (giới hạn tùy thuộc vào ứng dụng) của các cột có thể được tạo trong thời gian chạy hoặc theo định nghĩa giản đồ.

4. Graph-Oriented -Chiến lược này có thể giúp các truy vấn dữ liệu phức tạp cũng được thực hiện trong một khoảng thời gian xấp xỉ nhỏ hơn so với các cơ sở dữ liệu khác bằng cách sử dụng các chiến lược được đề xuất ở trên

Ngoài ra, cơ sở dữ liệu phi quan hệ cung cấp sự linh hoạt cao khi chèn hoặc xóa một thuộc tính từ cơ sở dữ liệu vì chúng không có giản đồ cơ sở dữ liệu cố định. Dựa trên yêu cầu của ứng dụng, chúng ta có thể sử dụng các loại cơ sở dữ liệu NoSQL khác nhau và mỗi cơ sở dữ liệu NoSQL có các chức năng riêng, mô hình dữ liệu và các tùy chọn cấu trúc của cơ sở dữ liệu phụ thuộc vào ứng dụng.

Trong bài báo này, chúng tôi tập trung vào một trong những Các công nghệ NoSQL, cụ thể là MongoDB và thực hiện so sánh với MySQL để làm nổi bật lý do tại sao MongoDB có nhiều khả năng hơn MySQL. Ngoài ra, chúng tôi sẽ trình bày những lợi ích của việc sử dụng cơ sở dữ liệu phi quan hệ trong một ứng dụng khác nhau, sử dụng MongoDB làm cơ sở dữ liệu NoSQL.



*Hình 1: Cấu trúc của MongoDB*

**2 .TỔNG QUAN VỀ MongoDB**

MongoDB là một lược đồ ít định hướng tài liệu hơn cơ sở dữ liệu. Cái tên MongoDB xuất phát từ "khổng lồ". Cơ sở dữ liệu được viết bằng C++ và được thiết kế để có thể mở rộngLý do chính cho sự di chuyển ra khỏi mô hình quan hệ là thực hiện mở rộng quy mô dễ dàng hơn. Ý tưởng cơ bản là thay thế khái niệm “hàng” với mô hình linh hoạt hơn; tài liệu". Bằng cách sử dụng nhúng tài liệu và mảng, quan điểm này làm cho nó có thể đại diện cho thứ bậc phức tạp mối quan hệ với một bản ghi duy nhất. MongoDB cung cấp hiệu suất cao, khả năng hoạt động cao, tính sẵn sàng cao và khả năng mở rộng dễ dàng. MongoDB hoạt động dựa trên ý tưởng cơ bản về bộ sưu tập và tài liệu. Cơ sở dữ liệu: - Cơ sở dữ liệu là một vật lý, thời gian thực container cho các bộ sưu tập. Mỗi và mọi cơ sở dữ liệu có tập hợp tệp duy nhất của riêng mình trên hệ thống tệp. Một máy chủ MongoDB cụ thể thường có nhiều cơ sở dữ liệu.

Bộ sưu tập: - Bộ sưu tập là một tập hợp tài liệu MongoDB. Nó tương tự như một bảng RDBMS.Một bộ sưu tập hoạt động trong một cơ sở dữ liệu duy nhất. Bộ sưu tập không thực thi lược đồ. Các tài liệu trong một bộ sưu tập có thể có nhiều trường khác nhau.Nói chung, mỗi và mọi tài liệu trong một bộ sưu tập có động cơ tương tự hoặc liên quan. Tài liệu: - Tài liệu là một tập hợp khóa-giá trị cặp. Tài liệu có lược đồ động. Lược đồ động có nghĩa là các tài liệu trong cùng một bộ sưu tập không cần phải có cùng một bộ các trường hoặc cột hoặc cấu trúc và các trường chung trong các tài liệu của bộ sưu tập có thể chứa nhiều các loại dữ liệu khác nhau.

Thiết kế dữ liệu trong cơ sở dữ liệu MongoDB chứa một tập hợp các bộ sưu tập. Một bộ sưu tập không có lược đồ được xác định trước, chẳng hạn như bảng và lưu trữ dữ liệu dưới dạng tài liệu. BSON (các đối tượng như JSON được mã hóa nhị phân) được sử dụng để lưu trữ tài liệu. Tài liệu là một tập hợp các trường có thể được coi là một hàng hoặc bộ trong một.

MongoDB có ngôn ngữ truy vấn riêng tên là Mongo Query Language. Để lấy một số tài liệu nhất định từ bộ sưu tập db, một tài liệu truy vấn được tạo có chứa các trường mà các tài liệu mong muốn phải khớp. Ví dụ,

**Insert Command**

db.users.insert ({ user id:”xyz123”, age: 34, status:”X”})

**Select Command**

db.users.find ({ status:”X”, age: 34})

**Delete Command**

db.users.remove ({ status:”X”})

**Drop Command**

db.users.drop ()

**3 SO SÁNH MongoDB & MySQL**

Theo đánh giá chi tiết của một số bài báo, một nghiên cứu so sánh được thực hiện giữa MongoDB và MySQL dựa trên khái niệm cơ bản của chúng và các lệnh được sử dụng cho các hoạt động khác nhau

**A.Dựa trên Điều khoản/Khái niệm**

|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Row/Tuple | Document |
| Column | Field |
| Table Join | Embedded  Documents |
| Primary Key (explicitly) | Primary Key (Default key \_id provided by MongoDB  implicitly) |
| Group by | Aggregation |
| Fixed schema | Schema less |

**B. Dựa trên các câu lệnh lược đồ**

|  |  |  |  |
| --- | --- | --- | --- |
| SQL schema | | MongoDB schema | |
| CREATE Command | | | |
| CREATE TABILE  teachers(  t\_id Varchar(30), age Number, status char(1),  PRIMARY KEY(id)) | | db.teachers.inSsert(  {t\_id :"abc123",age:55,  status : "A"}) | |
| DROP Command | | | |
| DROP TABLE teachers | | db.teachers.drop() | |
| INSERT Command | | | |
| INSERT  teachers( t\_id status) VALUES ("a123",45,"A") | INTO  , age, | db.teachers.insert(  {t\_id :"a123",age:45,stat  us:“A”} ) | |
| SELECT Command | | | |
| SELECT t\_id, status, age  FROM teachers | | db.teachers.find(  },{t\_id :1,  status:"B",age:45 }) | { |
| DELETE Command | | | |
| DELETE FROM  teachers  WHERE status ="D" | | db.teachers.remove( status:"D" }) | { |

**C. Dựa trên hiệu suất**

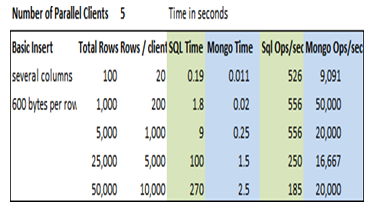
Một số tác giả bài báo nghiên cứu khác đã thực hiện thử nghiệm và do đó đã so sánh MongoDB với cơ sở dữ liệu MySQL. Họ đã thực hiện thử nghiệm bằng cách sử dụng hệ thống quản lý sách giáo khoa. Biểu đồ đã cho cho thấy kết quả thử nghiệm. Trong kiểm tra hiệu suất, các tác giả đã chèn 100 đến 50.000 thông tin sách giáo khoa vào cơ sở dữ liệu.

Chi phí thời gian cho MongoDB và MySQL được ghi lại như trong hình. Hai yếu tố quan trọng khiến MongoDB được ưu tiên hơn MySQL là:

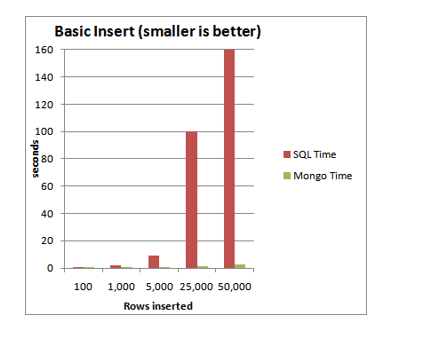
Chi phí thời gian cho MongoDB và MySQL được ghi lại như trong hình. Hai yếu tố quan trọng khiến MongoDB được ưu tiên hơn MySQL là:

**Tốc độ chèn**

Từ biểu đồ, chúng ta có thể thấy rằng MongoDB dành ít thời gian hơn MySQL, cho một lượng lớn thông tin như trong hình 2. Nó giúp MongoDB nhanh hơn từ 30 đến 50 lần so với MySQL như thể hiện trong hình 3.



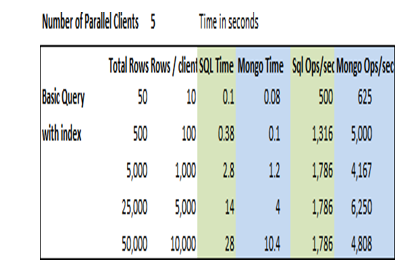
*Hình 2. So sánh tốc độ chèn*



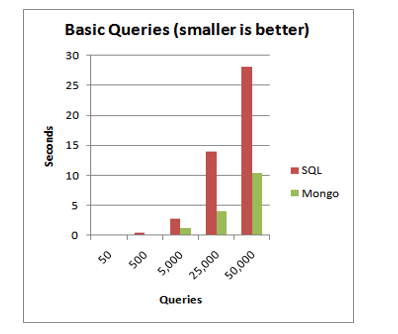
Hình 3: Thời gian chèn cho MySQL và MongoDB

**• Tốc độ truy vấn**

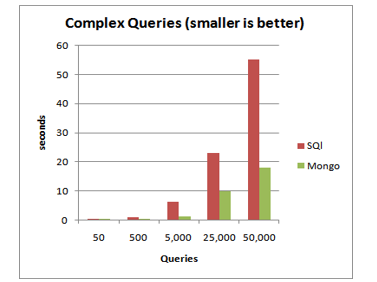
Trong hình 4, nó tính toán thời gian để lấy dữ liệu ra khỏi cơ sở dữ liệu. MongoDB dẫn đầu MySQL với hiệu năng gần như gấp 3 lần như trong hình 5, 6. Nhưng MongoDB dành nhiều thời gian hơn cho việc giải quyết vấn đề cũng như các vấn đề bảo trì sau và không dễ dàng hơn MySQL. Như vậy từ so sánh trên; nó chứng minh rằng đối với lượng dữ liệu lớn, MongoDB được ưu tiên hơn MySQL.



*Hình 4: SO SÁNH TỐC ĐỘ TRUY CẬP*



*Hình 5: Thời gian query cơ bản cho MySQL VÀ MongoDB*



*Hình 6: Thời gian truy vấn phức tạp cho MySQL VÀ MongoDB*

**4. Kết Luận**

RDBMS sẽ không biến mất, chúng chắc chắn cần thiết. Nhưng các yêu cầu lưu trữ cho thế hệ ứng dụng mới phần lớn khác với các ứng dụng cũ. Chúng ta có thể chọn MongoDB thay vì MySQL vì 2 yếu tố dễ sử dụng và hiệu năng. Chúng tôi kết luận rằng nếu ứng dụng của bạn sử dụng nhiều dữ liệu hoặc lưu trữ nhiều dữ liệu, truy vấn nhiều dữ liệu và thường dựa vào dữ liệu của nó, thì bạn nên làm điều đó một cách hiệu quả hoặc đốt tài nguyên (tức là tiền). Cuối cùng, báo cáo kết luận bằng cách đề xuất phương pháp tích hợp cơ sở dữ liệu bằng cách sử dụng phần mềm trung gian giữa hai lớp. Trong phương pháp này, ứng dụng không phải xem xét về độ phức tạp của lớp cơ sở dữ liệu cơ bản có phân phối và lưu trữ dữ liệu. Họ phải sử dụng ngôn ngữ truy vấn SQL cơ bản để nhận kết quả từ cơ sở dữ liệu và tất cả các quy tắc chuyển đổi định dạng sẽ được thực hiện bởi Siêu dữ liệu. Hệ thống được đề xuất vì MongoDB mới ra đời, trong khi ngôn ngữ SQL tiêu chuẩn đã qua nhiều năm và do đó, nếu chúng ta hợp nhất cả hai, chúng ta có thể sử dụng các tính năng của cả hai cơ sở dữ liệu. Mặc dù, NoSQL có lợi thế là mở rộng theo chiều ngang, nhưng đối với các yêu cầu SQL phức tạp, nó không thể hỗ trợ chúng rất tốt. Đối với Truy vấn dựa trên KEY/ VALUE và yêu cầu lưu trữ dữ liệu lớn, NOSQL là một lựa chọn tốt.

**5.THAM KHẢO**

SQL Support over MongoDB using Metadata Sanobar Khan\*, Prof.Vanita Mane\*\*

Bogdan George Tudorica, Cristian Bucur, “A Comparison between several NoSQL Databases with comments and notes”

Jing Han, Haihong E, Guan Le, “Survey on NoSQL Database”, IEEE 978-1-4577-0208-2,2011

Neal Levitt, “Will NoSQL Databases Live Up to Their Promise?” IEEE Computer Society, vol.43, no.2, pp.12-14,

Feb.2010

A Comparative Study: MongoDB vs. MySQL Conference Paper · June 2015 DOI: 10.13140/ RG.2.1.1226.7685

A comprehensive comparison of SQL and MongoDB databases by Rajat Aghi, Sumeet Mehta, Rahul Chauhan, Siddhant Chaudhary and Navdeep Bohra.