***Nghiên cứu so sánh giữa MongoDB và My SQl***

Tóm tắt- Trong bài báo này, chúng tôi sẽ cố gắng trình bày một nghiên cứu so sánh về cơ sở dữ liệu phi quan hệ và cơ sở dữ liệu quan hệ. Chúng tôi chủ yếu tập trung trình bày vào một triển khai của công nghệ cơ sở dữ liệu NoSQL, cụ thể là MogoDB, và so sánh với một triển khai khác của cơ sở dữ liệu quan hệ, cụ thể là MySQL, và do đó giải thích lý do tại sao MongoDB hiệu quả hơn MySQL, Chúng tôi cũng sẽ trình bày những ưu điểm của sử dụng cơ sở dữ liệu phi quan hệ so với cơ sở dữ liệu quan hệ, được tích hợp trong một diễn đàn trong lĩnh vực phát triển cá nhân và nghề nghiệp. Cơ sở dữ liệu NoSQL được sử dụng để phát triển diễn đàn là MongoDB 16, 7) và được chọn từ nhiều cơ sở dữ liệu không hợp lý, nhờ một số khía cạnh mà chúng tôi sẽ nhấn mạnh trong bài viết này. Việc tích hợp cơ sở dữ liệu trong khuôn khổ cũng sẽ được trình bày.

Thuật ngữ chỉ mục MySQL, MongoDB, NoSQL, RDBMS, cơ sở dữ liệu không quan hệ

***I. GIỚI THIỆU***

vài năm trước, một ứng dụng thường chỉ có hàng nghìn người dùng đến hàng chục nghìn người dùng có thể truy cập trong điều kiện tốt nhất, hiện tại có những ứng dụng có hàng triệu người dùng và được kết nối 24/7, 365 ngày mỗi năm. Điều quan trọng là sử dụng cơ sở dữ liệu phù hợp, hỗ trợ kết nối đồng thời hàng trăm nghìn người dùng.

Cơ sở dữ liệu quan hệ được sử dụng rộng rãi trong hầu hết các ứng dụng và chúng có hiệu suất tốt khi xử lý một lượng dữ liệu hạn chế. Để xử lý khối lượng dữ liệu khổng lồ như internet, đa phương tiện và mạng xã hội, việc sử dụng cơ sở dữ liệu quan hệ truyền thống là không hiệu quả. Để khắc phục vấn đề này, thuật ngữ "KHÔNG CÓ SQL" đã được giới thiệu. Thuật ngữ NoSQL được đặt ra bởi Carlo Strozzi vào năm 1998 và dùng để chỉ các cơ sở dữ liệu phi quan hệ, thuật ngữ này sau đó được giới thiệu lại vào năm 2009 bởi Eric Evans. Gần đây hơn, thuật ngữ này đã nhận được một nghĩa khác, đó là "Không chỉ SQL", đây là một biến thể nhẹ nhàng hơn của việc xác định thuật ngữ, so với trước đây của nó.

NOSQL, không phải là một công cụ, mà là một phương pháp bao gồm một số công cụ bổ sung và cạnh tranh [1]. Ưu điểm chính của cơ sở dữ liệu NoSQL là, không giống như cơ sở dữ liệu quan hệ, nó có thể xử lý dữ liệu phi cấu trúc như tài liệu, e-mail, đa phương tiện và mạng xã hội một cách hiệu quả [1, 10]. Cơ sở dữ liệu phi quan hệ không sử dụng các nguyên tắc RDBMS (Hệ thống quản lý cơ sở dữ liệu quan hệ) và không lưu trữ dữ liệu trong các bảng, lược đồ không cố định và có mô hình dữ liệu rất đơn giản. Thay vào đó, họ sử dụng các khóa nhận dạng và dữ liệu có thể được tìm thấy dựa trên các khóa được chỉ định. Có bốn chiến lược để lưu trữ dữ liệu trong cơ sở dữ liệu phi quan hệ, như được trình bày trong [2], và chúng như sau:

1. Khóa-Giá trị - Cơ sở dữ liệu Khóa-Giá trị [1, 2], chẳng hạn như Riak, là từ điển phân tán khái niệm và không có một lược đồ được xác định trước; chúng là lược đồ ít hơn. Chìa khóa có thể làtổng hợp hoặc tự tạo và giá trị có thể là bất kỳ thứ gì: chuỗi, JSON, BLOB và những thứ khác
2. Document Couchbase và MongoDB [2] là những cơ sở dữ liệu dựa trên tài liệu phổ biến nhất. Chúng linh hoạt trong loại nội dung vì chúng không có lược đồ được xác định trước. Về mặt khái niệm, chúng làm việc với các loại tài liệu khác nhau: JSON, BSON, XML và BLOB [2, 8]. Về cơ bản họ chỉ đại diện cho một chuyên môn hóa của cơ sở dữ liệu khóa-giá trị. Một tài liệu được viết/đọc bằng một phím. Ngoài chức năng Khóa-Giá trị, các tài liệu dựa trên cơ sở dữ liệu có thể sử dụng các các chức năng khác nhau để tìm tài liệu dựa trên nội dung của chúng.
3. Cột - Cơ sở dữ liệu từ danh mục BigTable, chẳng hạn như HBase và Hypertable là loại cột và phải có lược đồ được xác định trước [2, 4]. Dữ liệu được lưu trữ trong các ô được nhóm trong các cột và các cột được nhóm hợp lý thành các họ cột. Về mặt lý thuyết, chúng có thể chứa một số lượng không giới hạn (giới hạn tùy thuộc vào việc triển khai) các cột có thể được tạo trong thời gian chạy hoặc tại định nghĩa lược đồ
4. Định hướng đồ thị - Chiến lược này có thể hỗ trợ các truy vấn dữ liệu phức tạp cũng được thực hiện trong một khoảng thời gian tương đối nhỏ hơn so với các cơ sở dữ liệu khác sử dụng các chiến lược được đề cập ở trên.

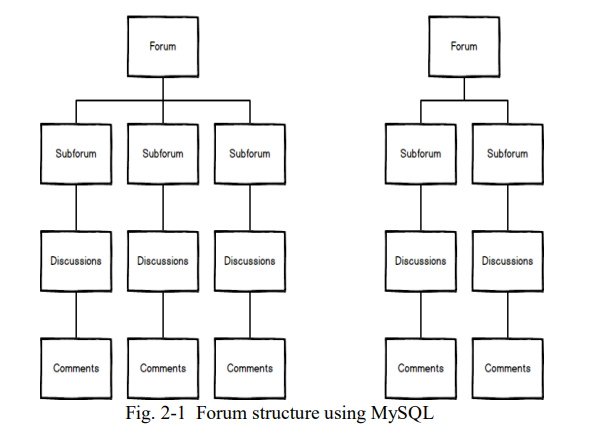
Ngoài ra, cơ sở dữ liệu phi quan hệ có thể cung cấp tính linh hoạt cao khi thêm hoặc xóa thuộc tính khỏi cơ sở dữ liệu vì chúng không có lược đồ cơ sở dữ liệu cố định. Tùy thuộc vào yêu cầu của ứng dụng, chúng ta có thể sử dụng các loại cơ sở dữ liệu NoSQL khác nhau và mỗi cơ sở dữ liệu NoSQL có các tính năng, mô hình dữ liệu và lựa chọn kiến ​​trúc của cơ sở dữ liệu tùy thuộc vào ứng dụng.

Trong bài báo này, chúng tôi tập trung vào một trong những công nghệ NOSQL, cụ thể là MongoDB, và so sánh với MySQL để làm nổi bật lý do MongoDB hiệu quả hơn MySQL. Ngoài ra, chúng tôi sẽ trình bày những ưu điểm của việc sử dụng cơ sở dữ liệu phi quan hệ trong ứng dụng diễn đàn, sử dụng MongoDB làm cơ sở dữ liệu NoSQL.

***II. PHÁT TRIỂN ỨNG DỤNG SỬ DỤNG Mongodb VS. MYSQL***

Chúng tôi đã thực hiện một nghiên cứu so sánh giữa MongoDB và MySQL dựa trên các khái niệm và lệnh của chúng được sử dụng cho các hoạt động khác nhau.

Nghiên cứu so sánh này dựa trên sự phát triển của một diễn đàn có cấu trúc động, tùy thuộc vào sở thích của người dùng. Sử dụng một cơ sở dữ liệu quan hệ như MySQL [9], cấu trúc sẽ là tĩnh và mỗi người dùng sẽ phải tuân theo một cấu trúc được thiết lập hoàn toàn trong cơ sở dữ liệu (ví dụ: diễn đàn - diễn đàn con - thảo luận - nhận xét). Do đó, mỗi người dùng buộc phải tuân theo cấu trúc, được minh họa trong Hình



Cấu trúc diễn đàn sử dụng MySQL

Forum: diễn đàn

Subforum: diễn đàn con

Discussions: thảo luận

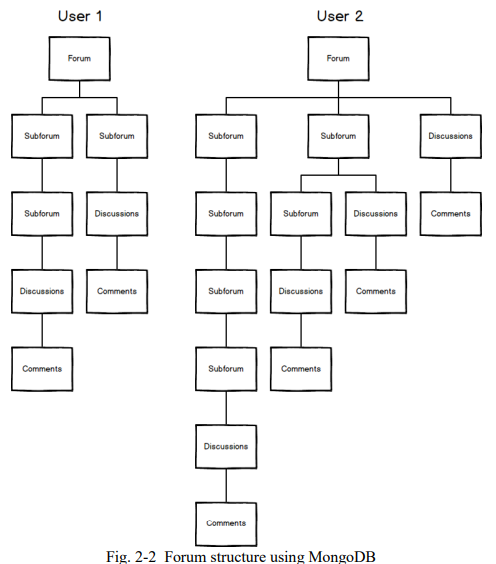
Comments: bình luận

Trong Hình 2-1, mỗi người dùng có số lượng diễn đàn con khác nhau trong một diễn đàn, nhưng phương pháp thực hiện mà mỗi người dùng phải đáp ứng là giống nhau: đầu tiên, họ cần tạo một diễn đàn, sau đó là các diễn đàn con, tiếp theo là tạo của các cuộc thảo luận và cuối cùng là bằng cách bổ sung các ý kiến ​​trong cuộc thảo luận.

Trong ứng dụng này, một cơ sở dữ liệu phi quan hệ, MongoDB, đã được chọn - một cơ sở dữ liệu dựa trên các tài liệu JSON với các lược đồ động, được viết bằng C++. MongoDB ban đầu được ra mắt vào năm 2009, hiện đang được phát triển và đang mở rộng [6, 7]. MongoDB là một cơ sở dữ liệu có thể được sử dụng cả trong các dự án nhỏ có vài nghìn người dùng, nhưng chủ yếu dành cho các sản phẩm và ứng dụng chứa hàng trăm nghìn người dùng, bao gồm Craigslist, eBay, Foursquare hoặc New York Times [3, 4].

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 như bảng và lưu trữ dữ liệu dưới dạng tài liệu BSON (đối tượng giống như JSON được mã hóa nhị phân) [1, 10]. Một tài liệu là một tập hợp các trường và có thể được coi là một hàng trong một bộ sưu tập. Nó có thể chứa các cấu trúc phức tạp như danh sách hoặc thậm chí toàn bộ tài liệu. Mỗi tài liệu có một trường ID, được sử dụng làm khóa chính và mỗi bộ sưu tập có thể chứa bất kỳ loại tài liệu nào, nhưng các truy vấn và chỉ mục chỉ có thể được áp dụng trên các bộ sưu tập [1]

Sử dụng cơ sở dữ liệu không quan hệ, chẳng hạn như MongoDB, để phát triển một diễn đàn, cho phép cấu trúc dành riêng cho từng người dùng, do đó mỗi người dùng có khả năng tổ chức diễn đàn của mình theo một cách riêng, như trong Hình 2-2.



Cấu trúc diễn đàn sử dụng MongoDB

Hình 2-2 cho thấy rằng việc sử dụng cơ sở dữ liệu phi quan hệ, cấu trúc tĩnh trong đó một diễn đàn chứa các diễn đàn con, các diễn đàn con chứa các cuộc thảo luận và các cuộc thảo luận chứa các bình luận, không bắt buộc phải đáp ứng. Như vậy, một diễn đàn con có thể chứa diễn đàn con khác, diễn đàn này có thể chứa diễn đàn con khác, diễn đàn khi nữa cũng có thể chứa diễn đàn con khác, v.v. Ngoài ra còn có khả năng đính kèm các cuộc thảo luận trực tiếp vào một diễn đàn mà không cần phải tạo một diễn đàn con để mở một cuộc thảo luận mới.

Ứng dụng được phát triển bằng Symfomy2 [5], một khung công tác PHP mã nguồn mở cho phép dễ dàng tích hợp cơ sở dữ liệu MongoDB. Việc tích hợp cơ sở dữ liệu MongoDB trong khung được thực hiện thông qua một gói, một gói phụ trợ phải được đăng ký trong danh sách các gói được sử dụng trong ứng dụng.

Sau khi gói đã được tích hợp vào Symfony2 [5], cấu hình cơ sở dữ liệu đã được thực hiện (lựa chọn máy chủ - trong trường hợp này là máy chủ cục bộ và lựa chọn tên cơ sở dữ liệu - chọn cơ sở dữ liệu không tồn tại không dẫn đến lỗi, nhưng việc tạo của cơ sở dữ liệu), như sau:

# app/config/config.yml

doctrine\_mongodb:

connections:

default:

server:

mongodb://localhost:27017

options: {}

default\_database: my\_database

document\_managers:

default:

auto\_mapping: true

***III. NGHIÊN CỨU SO SÁNH.***

Để hiểu thêm về cách sử dụng MongoDB, Bảng 3.1 cho thấy sự so sánh giữa các thuật ngữ được sử dụng trong MySQL và MongoDB tương ứng.

Như được hiển thị trong Bảng 3.1, trong MongoDB, một số thuật ngữ MySQL, chẳng hạn như bảng hoặc hàng, có tên khác, cụ thể là bộ sưu tập, tài liệu BSON tương ứng. Nói cách khác, chúng ta có thể nói rằng MongoDB chứa các bộ sưu tập, bộ sưu tập chứa các tài liệu và một tài liệu chứa nhiều trường

|  |  |
| --- | --- |
| MySQL | MongoDB |
| Database | Database |
| Table | Collection |
| Index | Index |
| Row | BSON Document |
| Column | BSON Field |
| Join | Embedded documents and linking |
| Primary key | Primary key |
| Group by | Aggregation |

Điều khoản giữa MySQL và MongoDB

Không giống như MySQL, nơi cơ sở dữ liệu được trình bày dưới dạng đồ họa dưới dạng bảng, trong MongoDB, cơ sở dữ liệu có cấu trúc đồ họa sau:

{

"\_id": "d4acaf3a76e4378b853eb15fde216722",

"username": "andra",

"email": "andra@gmail.com",

}

{

"\_id": "d4rvgf3a76e4378b853tr15fde216722",

"username": "ioana",

"email": "ioana@gmail.com",

}

Ví dụ trên hiển thị cơ sở dữ liệu cho người dùng, mỗi người dùng có một id duy nhất và được tạo tự động, tên người dùng và địa chỉ email. I Ứng dụng sẽ có 3 lớp người dùng là quản trị viên, người điều hành và người dùng thông thường. Mỗi người dùng có quyền tạo một diễn đàn/diễn đàn con riêng và có toàn quyền quản trị trên diễn đàn/diễn đàn con cụ thể đó. Trong diễn đàn con, người điều hành có quyền chỉnh sửa/xóa diễn đàn con và họ cũng có thể kiểm duyệt các cuộc thảo luận của người dùng khác, trong khi người dùng thông thường chỉ được phép đăng thảo luận và để lại nhận xét. Nếu cơ sở dữ liệu quan hệ đã được sử dụng, các cột dành cho diễn đàn và diễn đàn con sẽ xuất hiện ở tất cả người dùng diễn đàn, mặc dù người dùng bình thường sẽ không bao giờ có quyền tạo, sửa đổi hoặc xóa chúng, trừ khi tất nhiên, họ là quản trị viên của diễn đàn cụ thể đó. Sử dụng MongoDB, các trường liên quan đến diễn đàn và diễn đàn con này sẽ chỉ xuất hiện với những người dùng có quyền đó (người điều hành và quản trị viên), do đó giảm đáng kể dung lượng lưu trữ cao hơn nhiều khi sử dụng MySQL. Trên thực tế, những vai trò này không được gán cho người dùng mà cho chính các diễn đàn. Điều này là do người dùng có thể không bao giờ là quản trị viên trên diễn đàn mà chỉ là người dùng thông thường và không có lý do gì để chỉ định trường quản trị viên cho người dùng đó (trường này sẽ luôn là null), trong khi một diễn đàn sẽ luôn có trường này, bởi vì quản trị viên của nó thực tế là tác giả của nó. Như trong cơ sở dữ liệu quan hệ, MongoDB cũng có mối quan hệ một-nhiều, nhưng trong trường hợp này, khái niệm khóa ngoại không được sử dụng; thay vào đó, khái niệm chú thích được sử dụng. Do đó, trong trường hợp này, liên quan đến một diễn đàn, kết nối giữa diễn đàn và các diễn đàn con của nó như sau: trong tài liệu diễn đàn, các diễn đàn con được tham chiếu bằng cách sử dụng chú thích

@MongoDB\EmbedMany (targetDocument = "Subforum"),

and in the subforum document, the forum is referenced by the

following annotation that connects the two documents:

@MongoDB\ReferenceOne (targetDocument = "Forum",

inversedBy="Subforums").

class Forum

{

/\*\*

\* @MongoDB\EmbedMany(targetDocument="Subforum")

\*/

public $subforums = array();

}

Class Subforum

{

/\*\*

\* @MongoDB\ReferenceOne(targetDocument="Forum",

inversedBy="subforums")

\*/

protected $forum;

}

Để tạo cấu trúc linh hoạt cho diễn đàn, kết nối thực tế giữa diễn đàn và các diễn đàn con của nó, cũng như giữa diễn đàn - thảo luận, diễn đàn con - diễn đàn con, diễn đàn con - thảo luận được thực hiện bằng cách sử dụng ID duy nhất và cho phép liên kết động. giữa các thành phần của diễn đàn. Vì vậy, một cuộc thảo luận có thể dễ dàng được gắn vào cả diễn đàn và diễn đàn con, nhờ vào kết nối động giữa các bộ sưu tập.

Ngoài ra, nếu trong tương lai muốn chèn một trường mới vào cơ sở dữ liệu, điều này có thể dễ dàng đạt được bằng cách chèn tên và loại vào tài liệu tương ứng mà không ảnh hưởng đến dữ liệu thực được lưu trữ trong cơ sở dữ liệu và không cần xác định lại toàn bộ cấu trúc cơ sở dữ liệu.

Trong chương sau, chúng tôi sẽ trình bày sự khác biệt giữa hai cơ sở dữ liệu, MongoDB và MySQL, về mặt hiệu suất, sau khi thực hiện các hoạt động khác nhau trên cả hai cơ sở dữ liệu.

***IV. KIỂM TRA HIỆU SUẤT***

Để làm nổi bật những lợi thế của việc sử dụng phi quan hệ cơ sở dữ liệu MongoDB so với cơ sở dữ liệu quan hệ MySQL, các hoạt động khác nhau đã được thực hiện trên hai cơ sở dữ liệu. Các thao tác này là bốn thao tác cơ bản có thể được thực hiện trên bất kỳ cơ sở dữ liệu nào, cụ thể là:

1. Insert

2. Select (query)

3. Update

4. Delete.

Do kết quả kiểm tra phụ thuộc vào máy tính thực hiện các kiểm tra này, điều quan trọng cần lưu ý là tất cả các kết quả được trình bày bên dưới đều thu được từ các nghiên cứu được thực hiện trên máy tính có các đặc điểm sau: Windows 7 Ultimate 64-bit, bộ xử lý Intel Core i3 (2.4 GHZ), bộ nhớ RAM 4 GB.

***A. Thao tác chèn***

Chúng tôi bắt đầu thử nghiệm với việc tạo cơ sở dữ liệu mà không có bất kỳ nội dung nào, cả trong MongoDB và MySQL. Cấu trúc của hai cơ sở dữ liệu tương tự nhau và chúng có cùng số lượng cột/trường và bảng/tài liệu. Các yếu tố chung của hai cơ sở dữ liệu là:

- Table/document User with the columns: id, username, password, email.

- Table/document Forum with the columns: id, title, author, info (short description).

- Table/document Subforum with the columns: id, title, author, info, created, updated.

- Table/document Discussion with the columns: id, title, author, created, updated, content.

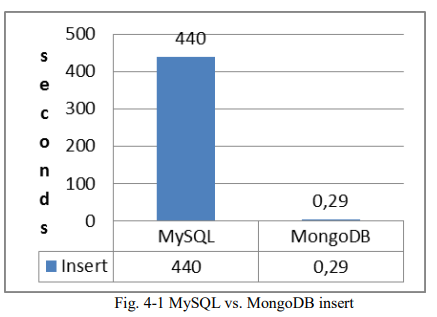
- Table/document Comments with the columns: id, author, created, content, approved.

Ngoài các trường commen này, còn tồn tại các trường khác liên kết các bảng trong MySQL và các tài liệu trong MongoDB và chúng thường là các khóa ngoại xác định ví dụ: một diễn đàn con thuộc về diễn đàn mẹ của nó hoặc tác giả của một bình luận nào đó.

Việc chèn dữ liệu đã bắt đầu trong cả hai cơ sở dữ liệu với việc chèn của người dùng. Đối với mỗi cơ sở dữ liệu, 10.000 người dùng đã được chèn vào. ID của họ được tạo tự động bởi cả hai cơ sở dữ liệu và cho tên người dùng, mật khẩu và địa chỉ email, chúng tôi đã sử dụng các hàm PHP như md5, rand, substr và str\_shuffle.

Để ghi lại thời gian cần thiết để chèn các phần tử vào cơ sở dữ liệu, chúng tôi đã sử dụng hàm microtime của PHP, ghi lại thời gian từ khi bắt đầu thời gian chạy tập lệnh và cho đến khi hoàn thành.

Như vậy, sau khi chạy hai tập lệnh, một cho MySQL và một cho Mongo, Hình 4.1 cho thấy 10.000 người dùng đã được đưa vào MySQL trong 440 giây, trong khi ở MongoDB, thời gian chỉ là 0,29 giây!



Khi người dùng đã được chèn vào cơ sở dữ liệu, việc chèn dữ liệu sẽ được chuyển đến các diễn đàn, diễn đàn con, thảo luận và nhận xét. Tất cả ID của tất cả các bảng được tạo tự động trong cả MySQL và MongoDB và để tạo các trường khác, chúng tôi đã sử dụng các chức năng được mô tả ở trên. Vì một diễn đàn con không thể tồn tại nếu một diễn đàn hoặc một diễn đàn con khác không được tạo trước và vì không có bình luận nào có thể tồn tại trên chính nó vì nó phải được sở hữu bởi một cuộc thảo luận, chúng tôi đã bắt đầu bằng cách chèn các diễn đàn, sau đó là các diễn đàn con, tiếp theo là chèn các thảo luận và cuối cùng là nhận xét. Để thử nghiệm, người ta đã quyết định chèn 5000 hàng cho mỗi bảng một phần, như sau: 5000 diễn đàn, 5000 diễn đàn con, tương ứng là 5000 thảo luận và 5000 bình luận. Vì vậy, sau khi chèn dữ liệu vào hai cơ sở dữ liệu, sau đây thời gian chèn kết quả: MySQL: 1010 giây, MongoDB 3,3331 giây. Chúng tôi nhận thấy rằng MongoDB dành ít thời gian hơn MySQL để chèn một lượng lớn thông tin vào cơ sở dữ liệu.

***B. Các thao tác truy vấn***

Để kiểm tra hiệu suất của các hoạt động truy vấn, chúng tôi đã thực hiện hai

chọn truy vấn, cụ thể là:

-Chọn tất cả các cuộc thảo luận mà người dùng đã tham dự và có ngày khác với ngày nhất định,

-Lựa chọn tất cả người dùng từ cơ sở dữ liệu và số lượng của các cuộc thảo luận bắt đầu bởi mỗi người dùng.

Lựa chọn đầu tiên có cú pháp MySQL sau (Chọn 1-MySQL)

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "SELECT d.username, d.dtitle, d.dcontent,

d.created, s.id, f.id

FROM discussion d

inner join subforum s on s.id = d.subforumid

inner join forum f on f.id = s.forumid

where d.username = `andra`

and d.created <> '2014-11-27'";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Select 1- MongoDB):

$m = new MongoClient();

$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$date = '1417091683';

$q = array(

'subcategories.topics.discussions.dauthor' =>

'andra',

'subcategories.topics.discussions.created' =>

array( '$ne' => new MongoInt32($date))

);

$cursor = $collection->find($q);

Lựa chọn thứ hai có cú pháp MySQL như sau (Chọn 2- Mysql):

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "select u.id, u.username, count(d.dtitle)

from users u

left outer join discussion d on d.userid

= u.id

group by u.id, u.username";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Select 2- MongoDB):

$m = new MongoClient();

$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$user =

'subcategories.topics.discussions.dauthor';

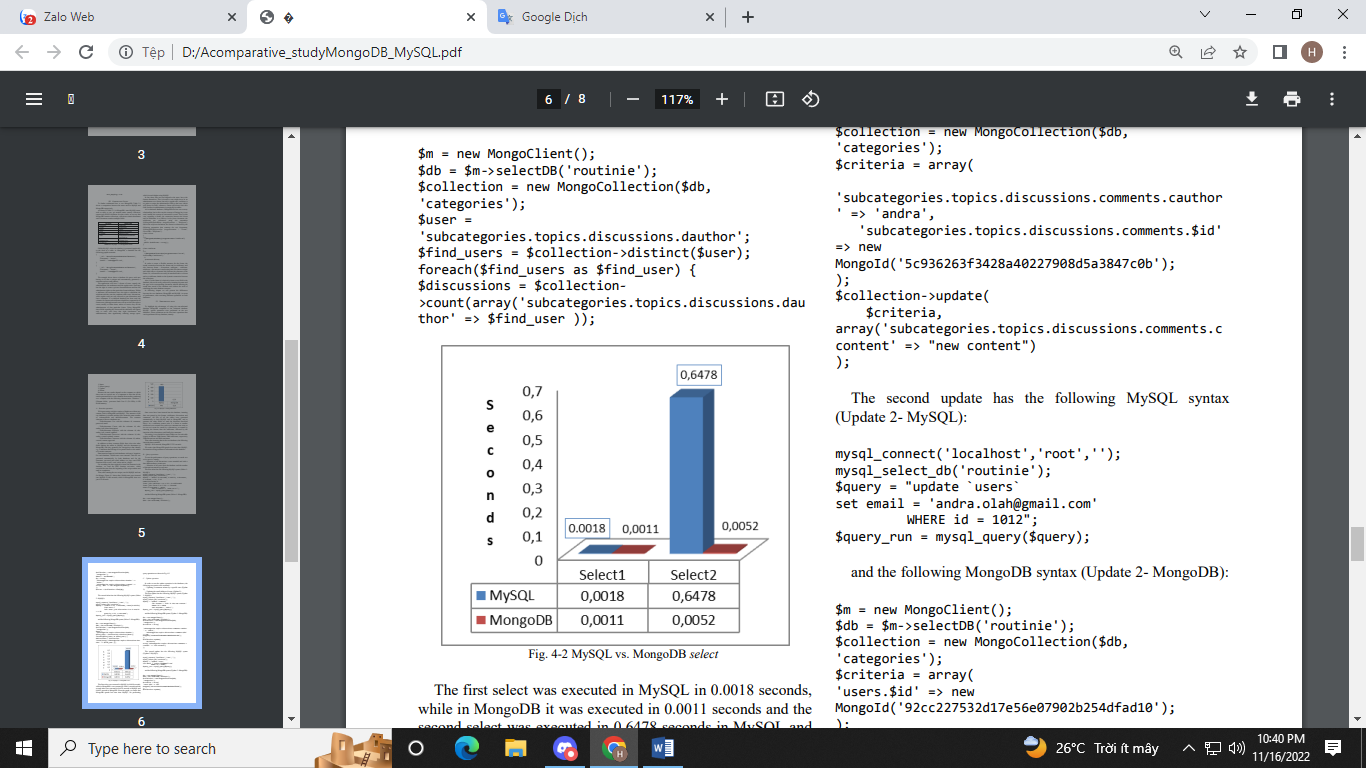
$find\_users = $collection->distinct($user);

foreach($find\_users as $find\_user) {

$discussions = $collection-

>count(array('subcategories.topics.discussions.dau

thor' => $find\_user ));



Lựa chọn đầu tiên được thực hiện trong MySQL trong 0,0018 giây, trong khi ở MongoDB, nó được thực hiện trong 0,0011 giây và khi ở lựa chọn thứ hai được thực hiện trong 0,6478 giây trong MySOL và 0,0052 giây trong MongoDB. Từ biểu đồ, chúng ta nhận thấy rằng MongoDB dành ít thời gian hơn MySQL để thực hiện hoạt động truy vấn như trong hình 4.2

***C. Hoạt động cập nhật***

Để kiểm tra hoạt động cập nhật trong cơ sở dữ liệu, hai câu hỏi tiếp theo đã được thực hiện: 1). Cập nhật nhận xét được viết bởi một người dùng cụ thể (Cập nhật Cập nhật địa chỉ email của người dùng. {Cập nhật 2). Bản cập nhật đầu tiên có cú pháp MySQL theo sau (Cập nhật1-MySQL);

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "update `comments`

set content = 'This is the new content.'

WHERE id = 10594

and username = `andra`";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Update 1- MongoDB):

$m = new MongoClient();

$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$criteria = array(

'subcategories.topics.discussions.comments.cauthor

' => 'andra',

'subcategories.topics.discussions.comments.$id'

=> new

MongoId('5c936263f3428a40227908d5a3847c0b');

);

$collection->update(

$criteria,

array('subcategories.topics.discussions.comments.c

content' => "new content")

);

The second update has the following MySQL syntax

(Update 2- MySQL):

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "update `users`

set email = 'andra.olah@gmail.com'

WHERE id = 1012";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Update 2- MongoDB):

$m = new MongoClient();

$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$criteria = array(

'users.$id' => new

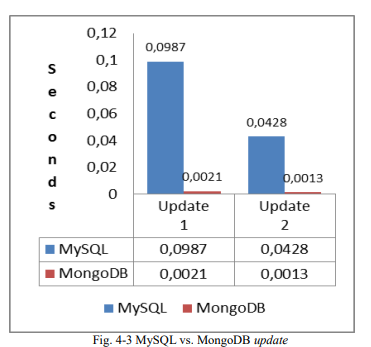
MongoId('92cc227532d17e56e07902b254dfad10');

);

$collection->update($criteria, array('users.email' =>

"andra.olah@gmail.com")

);



Bản cập nhật đầu tiên được thực thi trong MySQL trong 0,0987 giây. trong khi ở MongoDB, nó được thực thi trong 0,0021 giây và bản cập nhật thứ hai wvas được thực thi trong 0,0428 giây trong MySQL và

0,0013 giây trong MongoDB. Từ Hình 4-3, chúng tôi nhận thấy rằng MongoDB dành ít thời gian hơn MyŠQL để thực hiện thao tác cập nhật như trong Hình 4-3.

***D. Thao tác xóa***

Như các hoạt động khác được mô tả ở trên, chúng tôi đã thực hiện hai truy vấn xóa cho mỗi cơ sở dữ liệu.

Truy vấn đầu tiên đã xóa tất cả các nhận xét được đăng bởi người dùng (Xóa 1). Truy vấn thứ hai đã xóa tất cả các diễn đàn được tạo bởi một người dùng cụ thể (Xóa 2). Điều quan trọng cần đề cập là với việc xóa một diễn đàn. tất cả các diễn đàn con, nhận xét và thảo luận liên quan đến diễn đàn được chỉ định cũng sẽ tự động bị xóa. trong beth MySQL và MongoDB.

Yêu cầu xóa đầu tiên có cú pháp MySQL như sau (Delete I-MySQL):

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "delete from comments where username =

`andra`";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Delete 1- MongoDB):

$m = new MongoClient();

$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$criteria =

array('subcategories.topics.discussions.comments.c

author' => 'andra');

$collection->update(

$criteria,

array('$unset' =>

array('subcategories.topics.discussions.comments'

=> true)),

array('multiple' => true)

);

Lần xóa thứ hai có cú pháp MySQL sau (Xóa 2-MySOQL):

mysql\_connect('localhost','root','');

mysql\_select\_db('routinie');

$query = "delete FROM `forum` WHERE username

=`andra`";

$query\_run = mysql\_query($query);

and the following MongoDB syntax (Delete 2- MongoDB):

$m = new MongoClient();

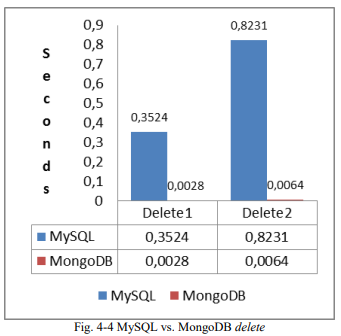
$db = $m->selectDB('routinie');

$collection = new MongoCollection($db,

'categories');

$q = array('admins.username' => 'andra');

$cursor = $collection->remove($q);



Từ hình 4-4, chúng tôi nhận thấy rằng lần xóa đầu tiên được thực hiện trong MySQL trong 0,3524 giây, trong khi ở MongoDB, nó được thực hiện trong 00028 giây và lần xóa thứ hai được thực hiện trong 0,8231 giây trong MySQL và 0,0064 giây trong MongoDB.

MongoDB cung cấp thời gian thực thi thấp hơn MySQL trong cả bốn thao tác cơ bản, điều này rất cần thiết khi một ứng dụng cần cung cấp hỗ trợ cho hàng nghìn người dùng đồng thời. Như vậy, so sánh trên chứng minh rằng đối với lượng dữ liệu lớn, MongoDB có hiệu suất tốt và nó được ưu tiên hơn MySQL.

***V. Kết luận***

Chuyển đổi từ cơ sở dữ liệu quan hệ sang cơ sở dữ liệu phi quan hệ có thể là một thách thức về nhiều mặt, trong đó cần nghiên cứu kỹ tất cả các loại cơ sở dữ liệu phi quan hệ có thể có và tìm ra giải pháp tối ưu cho ứng dụng cụ thể và tạo ra một cơ sở dữ liệu phi quan hệ. -cơ sở dữ liệu quan hệ có thể cung cấp chính xác các tính năng và hoạt động truy vấn giống như cơ sở dữ liệu sẽ thay thế. Nhưng việc sử dụng cơ sở dữ liệu quan hệ sẽ không bao giờ kết thúc, bởi vì nó cung cấp cho chúng ta một bộ tính năng vô song, bằng cách duy trì tính toàn vẹn và khả năng mở rộng của dữ liệu. Chủ yếu, công việc của nhà phát triển là quyết định cơ sở dữ liệu nào sẽ được sử dụng để đáp ứng các yêu cầu ứng dụng .Đối với ứng dụng này. L.e. diễn đàn. cơ sở dữ liệu phi quan hệ phù hợp nhất là MongoDB. bởi vì diễn đàn sẽ có hàng ngàn người dùng, hoặc thậm chí hàng chục ngàn và MongoDB đã kích hoạt khả năng tùy biến của nó để phù hợp với từng người dùng bằng cách tạo các diễn đàn riêng. mỗi cái có cấu trúc linh hoạt và năng động riêng.

Mặc dù được biết rằng MongoDB là một cơ sở dữ liệu không cho phép thao tác nối, nhưng vẫn có những giải pháp thay thế cho vấn đề này, phổ biến nhất là việc sử dụng các tham chiếu. đây là một quy ước để tạo tài liệu. Ví dụ. Khi chèn một bình luận vào cơ sở dữ liệu, bình luận nhất thiết phải chứa một tham chiếu đến tài liệu mà nó được liên kết tới, cụ thể là cuộc thảo luận và nó cũng sẽ chứa địa chỉ id của cuộc thảo luận mà nó thuộc về. Như vậy. kết nối giữa hai tài liệu được thực hiện thông qua tài liệu tham khảo.

Ưu điểm của việc sử dụng MongoDB được làm nổi bật hơn nữa bằng cách tiến hành các thử nghiệm và diễn giải kết quả của chúng, điều này đã được trình bày trong chương trước. MongoDB cung cấp thời gian thực thi thấp hơn MySQL trong cả bốn thao tác cơ bản. Điều đó là cần thiết khi một ứng dụng cung cấp hỗ trợ cho hàng nghìn người dùng đồng thời.

Chúng ta có thể chọn MongoDB thay vì MySQL nếu ứng dụng có nhiều dữ liệu , cần lưu trữ nhiều dữ liệu và nhiều dữ liệu cần phải truy vấn

Cũng cần lưu ý rằng trong cùng một ứng dụng,

Đôi khi mỗi người dùng có thể cần cài đặt tùy chỉnh của riêng mình và cơ sở dữ liệu quan hệ không cho phép tùy chỉnh toàn bộ

Chỉ dựa trên nhu cầu của người dùng. Do đó, ngày càng có nhiều ứng dụng bắt đầu sử dụng cơ sở dữ liệu phi quan hệ vì chúng cung cấp cấu trúc linh hoạt hơn, có thể định hình sau mỗi nhu cầu của người dùng. Chúng được thiết kế để lưu trữ lượng lớn dữ liệu, là cơ sở dữ liệu không chuẩn hóa và chúng được tăng hiệu suất sử dụng

***Tài liệu tham khảo***

[1] K. Sanobar, M. Vanita, “SQL Support over MongoDB using Metadata”,

International Journal of Scientific and Research Publications, Volume

3, Issue 10, October 2013

[2] T. Frătean, Bazele de date NoSQL – o analiză comparativă, To Day

Software Magazine, number 10 [Online]. Available:

http://www.todaysoftmag.ro/article/304/bazele-de-date-nosql-o-analizacomparativa, acceseed oct. 2014.

[3] S. Hoberman, “Data Modeling for MongoDB”, Publisher by Technics

Publications, LLC 2 Lindsley Road Basking Ridge, NJ 07920, USA,

ISBN 978-1-935504-70-2, 2014.

[4] S. Hall (2014, Jul) MySQL vs MongoDB, jul, 2014, Available:

http://www.scriptrock.com/articles/mysql-vs-mongodb, acceseed nov.

2014

[5] F. Potencier, “The symfony 1.3 & 1.4 Reference Guide”, Publisher:

Sensio SA (November 4, 2009), ISBN: 978-2918390145, pp 278.

[6] MongoDB, Inc. (2015, Aprilie), MongoDB Ops Manager Manual

Release 1.6, [Online]. Available:

https://docs.opsmanager.mongodb.com/current/opsmanager-manual.pdf

[7] R. P Padhy, M. R. Patra, S. C. Satapathy, “RDBMS to NoSQL:

Reviewing Some Next-Generation Non-Relational Database’s”,

International Journal of Advance Engineering Sciences and

Technologies, Vol. 11, Issue No. 1, 015-030, 2011.

[8] J. Clarence, M. Tauro, S. Aravindh, A. B. Shreeharsha, “Comparative

Study of the New Generation, Agile, Scalable, High Performance

NOSQL Database”, International Journal of Computer Applications,

ISSN 0975 – 888, Volume 48– No.20, June 2012.

[9] M. Irfan (March 14, 2014) Tools and tips for analysis of MySQL’s Slow

Query Log Oracle Corporation. [Online]. Available:

http://www.percona.com/blog/2014/03/14/tools-and-tips-for-analysis-ofmysqls-slow-query-log/

[10] Z. Wei-Ping, LI Ming-Xin, H. Chen, “Using MongoDB to Implement

Textbook Management System instead of MySQL”, IEEE 3rd

International Conference on Communication Software and Networks

(ICCSN), ISSN 978-1-61284-486, 2011.