Báo cáo MongoDB

Nhóm 21

Mục lục

Contents

Mục lụ	c	2
Chươ	ơng 1: Cơ bản về MongoDB	3
1.	Tổng quan	3
2.	Tính phổ biến của MongoDB	4
3.	Tính tương thích	4
4.	Khả năng hỗ trợ cho các ngôn ngữ	4
5.	Các công cụ của MongoDB	5
6.	Tính năng vượt trội của MongoDB	5
Chươ	ơng 2: Cài đặt	5
1.	Lựa chọn các phiên bản cài đặt	5
2.	Cài đặt MongoDB	6
2.3	Start MongoDB Server	8
3.	Chạy MongoDB	8
4.	Cài đặt thêm các Driver hỗ trợ phát triển	10
Chươ	ơng 3: Mô hình dữ liệu	12
1.	Database	13
2.	Collection	13
3.	Document (object)	13
4.	Field	13
Chươ	ơng 4: Thao tác với dữ liệu	14
1.	Shell	14
2.	Kết nối database	14
3.	Lệnh thao tác cơ bản với collection	15
4.	Thao tác trên dữ liệu	16
Chươ	ong 5: GridFS	20
1.	Yêu cầu đặt ra	20
2.	Tổng quát GridFS	20
3.	Khi nào dùng GridFS?	20

Chươ	ng 6: Replication	21
	Muc đích Replication của MongoDB	
	Khi nào thì sử dụng Replication	
	ng 7: Sharding	
	Sự cần thiết phải có cơ chế Sharding	
3.	Thực hiện sharding với MongoDB	25

Chương 1: Cơ bản về MongoDB

1. Tổng quan

MongoDB là hệ quản trị cơ sở dữ liệu NoSQL hướng văn bản. Mongodb được phát triển bởi công ty phần mềm **10gen** vào năm 2007. Năm 2009 MongDB được phát hành mã nguồn mở vào năm 2009.

2. Tính phổ biến của MongoDB

Hiện nay theo wikipedia có các công ty đang sử dụng mongoDB để quản lý cơ sở dữ liệu của mình là:

- MTV Networks
- craigslist
- Disney Interactive Media Group
- Wordnik
- diaspora
- Shutterfly
- foursquare
- bit.ly
- The New York Times
- SourceForge
- Business Insider
- Etsy
- CERN LHC
- Thumbtack
- AppScale
- Uber
- The Guardian
- Barclays

3. Tính tương thích

MongoDB có khả năng tương thích với Windows, Linux, OS X, Solaris

4. Khả năng hỗ trợ cho các ngôn ngữ

- C
- C++
- C# / .NET
- Erlang
- Haskell
- Java
- JavaScript
- Lisp
- node.JS
- Perl
- PHP
- Python
- Ruby
- Scala

5. Các công cụ của MongoDB

- o Mongo
- o Mongostat
- o Mongotop
- Mongosniff
- Mongoimport
- o mongodump

6. Tính năng vượt trội của MongoDB

Hướng văn bản

- o Sử dụng tốt cho các ngôn ngữ lập trình kiểu dữ liệu.
- Các văn bản(document) và mảng được nhúng vào nên không cần thực hiện phép kết, làm tăng tốc đô truy vấn.
 - o Lược đồ đông nên dễ dàng cho việc phát triển tiến, tiến hóa lược đồ.
 - o Không cần phép kết và transactions nên làm tăng tốc độ truy vấn.

> Tính thực thi cao

- o Không có phép kết, và dùng phương pháp nhúng nên làm cho đọc và viết dữ liệu nhanh hơn.
- o Chỉ mục chứa cả khóa chỉ mục của các document và mảng được nhúng.
- o Không có sư thông báo đáp lai đã nhân được thông tin(no acknowledgements)
- > Tính sẵn sàng cao:
 - o Các máy chủ bản sao cập nhật một cách tự động.

Dễ dàng tăng tính mở rộng:

- o Cơ chế sharding tự động (tự động phân vùng dữ liệu trên máy chủ).
- Động bộ hóa dữ liệu tốt.
- Ngôn ngữ truy vấn phong phú

Chương 2: Cài đặt

1. Lựa chọn các phiên bản cài đặt

MongoDB cho phép chúng ta Download miễn phí tại website Mongodb.org/dowload. Việc bạn cần cân nhắc là lựa chọn phiên bản phù hợp với Hệ điều hành và kiến trúc hệ thống của mình.

Hiện tại MongoDB có thể chạy trên các hệ điều hành Windown, Linux, MacOS và Solaris với hai kiểu kiến trúc là x86 và x64.

Chú ý:Khi biết được Hệ điều hành và kiến trúc hệ thống của mình, bạn phải chọn phiên bản

MongoDB hợp lý cho hệ thống mà bạn đang muốn vận hành và sử dụng.

- 2. Cài đặt MongoDB
- a. Trên Windown

Bước 1: Tải bản mongoDB từ web mongodb.org: http://www.mongodb.org/downloads

Bước 2: Lựa chọn phiên bản phù hợp, ở đây ta ví dụ cài trên windows 7 và theo kiếm trúc Hệ Điều Hạnh 64bit, nên ta tiến hành tải:

http://www.mongodb.org/dr/downloads.mongodb.org/win32/mongodb-win32-x86_64-2.0.3.zip/download

Bước 3: Sau khi tải về, tiến hành giải nén vào 1 thư mục.

Ví dụ: D:\xampp\mongodb (đặt đâu cũng được). Sau đó vào thư mục trên và tạo thư mục là data (ví dụ đường dẫn thư mục data sau khi tạo là: D:\xampp\mongodb\data)

Bước 4: Chạy CMD với quyền quản trị để tiến hành cài đặt mongodb như service. Để chạy cmd với quyền administrator bạn vào thư mục C:\Windows\System32, nhấp chuột phải vào cmd.exe chon Run as administrator

Bước 5: Từ CMD gõ lệnh:

```
D:\xampp\mongodb\bin\mongod.exe --install --logpath
D:\xampp\mongodb\log.txt --dbpath D:\xampp\mongodb\data
```

Ta sẽ nhận được thông báo sau:

```
all output going to: D:xamppmongodblog.txt
```

Bước 6: Tiến hành start mongodb

```
net start mongodb
```

Ta sẽ nhận được thông báo sau

```
The Mongo DB service is starting.
The Mongo DB service was started successfully.
```

Mở tập tin D\:xampp\mongodblog.txt, sẽ nhân được thông tin của mongodb đang chay

Creating service MongoDB.

```
Service creation successful.

Service can be started from the command line via 'net start "MongoDB"'.

Tue Mar 06 20:27:18 dbexit:

Tue Mar 06 20:27:18 shutdown: going to close listening sockets...

Tue Mar 06 20:27:18 shutdown: going to flush diaglog...

Tue Mar 06 20:27:18 shutdown: going to close sockets...

Tue Mar 06 20:27:18 shutdown: waiting for fs preallocator...

Tue Mar 06 20:27:18 shutdown: lock for final commit...

Tue Mar 06 20:27:18 shutdown: final commit...

Tue Mar 06 20:27:18 shutdown: closing all files...

Tue Mar 06 20:27:18 closeAllFiles() finished

Tue Mar 06 20:27:18 dbexit: really exiting now
```

Bước 7: Để chạy mongodb, trong cmd ta gõ tiếp:

```
net start mongodb
```

Kết quả là:

```
The Mongo DB service is starting.
The Mongo DB service was started successfully.
```

Để kiểm tra lần nửa việc mongodb đã cài đặt thành công, bạn mở trang web:

http://localhost:27017

Nội dung trang web sẽ như sau:

You are trying to access MongoDB on the native driver port. For http diagnostic access, add 1000 to the port number

b. Trên Linux

Bước 1. Install MongoDB Database Server

Bước 1.1 Download & Unzip package

Download bản Linux 64 bit về rồi giải nén thành /opt/mongodb-linux-x86_64-1.6.1

MongoDB mặc dù viết bằng C++ nhưng chỉ cần download về giải nén là chạy được!

```
#cd /opt
#wget http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-
1.6.5.tgz
#tar -xvzf mongodb-linux-x86_64-1.6.5.tgz
#ln -s mongodb-linux-x86_64-1.6.1 mongodb
```

Buróc 1.2 Add and enable 10gen MongoDB repository

```
#vi /etc/yum.repos.d/10gen-mongodb.repo
```

Add:

```
[10gen] name=10gen Repository baseurl=http://downloads-
distro.mongodb.org/repo/redhat/os/x86_64 gpgcheck=0
#yum install mongo-10gen mongo-10gen-server
```

2. Configure MongoDB Database Server

Buróc 2.1 Edit /etc/mongod.conf file:

```
#vi /etc/mongod.conf
  logpath=/var/log/mongod.log port=27017
dbpath=/var/lib/mongo
2.2 Fix /var/lib/mongo directory permissions
#chown-R mongod:mongod /var/lib/mongo/
```

2.3 Start MongoDB Server

#/etc/init.d/mongod start

2.4 Start MongoDB on boot

```
#chkconfig --levels235 mongod on
```

3. Test MongoDB Server

Buróc 3.1 Open MongoDB Command Line Client

```
# mongo
> use test
switched to db test
```

3. Chay MongoDB

a. Yêu cầu trước khi khởi động

Trước khi khởi động các Services của MongoDB, bạn cần tạo một data directory cho MongoDB lưu trữ các file mà trong quá trình chạy nó tạo ra. Mặc định thì MongoDB lưu trữ Data trong thư mục C:/data/db đối với HĐH Windown và /data/db đối với HĐH Linux

Chú ý: MongoDB không tư tạo thư mục data cho mình được. Vì vậy chúng tạ phải tạo thư mục

đó bằng tay hoặc là MongoDB sẽ báo lỗi khi chạy và hiển thị một vào error message cho bạn.

Một điều quan trọng nữa là khi bạn tạo thư mục data phải chắc rằng thư mục đó có quyền read, write và create thư mục con.

Khi bạn không muốn MongoDB lưu trữ vào thư mục mặc định của nó thì bạn phải chỉ ra đường dẫn lưu trữ bằng việc chỉ ra –dbpath khi thực thi các services.

Sau khi tạo các thư mục cần thiết, ta có thể start MongoDB core Database bằng cách thực thi mongod appilcation.

b. Tổng quan về các Layout trong MongoDB

Sau khi cài đặt thành công MongoDB thì chúng ta có cấu trúc các ứng dụng như sau

```
|-- bin
| -- mongo (the database shell)
| -- mongod (the core database server)
| -- mongos (auto-sharding process)
| -- mongodump (dump/export utility)
| -- mongorestore (restore/import utility)
```

Figure 2-1. Drilling down on the included MongoDB applications

Chúng ta có tổng cộng là 5 ứng dụng được lưu trữ trong thư mục /bin để có thể sử dụng các chức năng của MongoDB database bằng cách liên kết chúng với nhau. Mặc định MongoDB đã thiết lập các liên kết đó, việc chúng ta cần làm là start các ứng dụng.

Trong 5 ứng dụng trên, có 2 ứng dụng quan trọng nhất là mongo và mongod

Mongo là ứng dụng cho phép chúng ta sử dụng database shell

Mongod là ứng dụng khởi động services

c. Sử dụng MongoDB Shell

Nếu chúng ta khởi động MongoDB services với các tham số mặc định và khổi động shell với các setup mặc định thì MongoDB sẽ mặc định connect vào database "test" chạy ở chế độ Local Host. Database "Test" được tạo ra mặc định ngay khi chúng ta connect đến nó. Đó là một trong những hữu ích trong MongoDB mà chúng ta cần biết đến: tức là nếu bạn truy cập đến một đối tượng mà nó chưa tồn tại thì MongoDB sẽ tạo đối tượng đó ra. MongoDB không đảm bảo các tham số mặc định của đối tượng đó là tốt. Nếu chưa phù hợp với mục đích sử dụng thì ta phải tùy chỉnh lại các tham số của đối tượng đó.

```
MongoDB shell version: 1.5.5
connecting to: test
```

Figure 2-2. The MongoDB shell

Trước khi đi sâu vào việc tìm hiểu MongoDB, chúng ta có thể sơ lược một số thao tác cơ bản với MongoDB shell

Command	Function
show dbs	Hiển thị tên tất cả các Database có trong MongoDB
show collections	Hiển thị tất cả các Collection có trong Database hiện tại
show users	Hiển thị User hiện tại trong transaction
use <db name=""></db>	Set database ta muốn thao tác

4. Cài đặt thêm các Driver hỗ trợ phát triển

Hiện tại MongoDB có hỗ trợ các ngôn ngữ phát triển như:

- C
- C++
- Erlang
- Haskell
- Java
- Javascript
- .NET (C# F#, PowerShell, etc)
- Node.js
- Perl
- PHP
- Python
- Ruby
- Scala

Chúng ta sẽ tìm hiểu cách thêm các Driver mà MongoDB hỗ trợ để phát triển. Trong đó 2 ngôn ngữ mà được chú trọng nhất là PHP và Python

a. Cài đặt PHP Driver

Trước tiên bạn phải Download PHP driver cho hệ điều hành hiện tại. Driver cho các ngôn ngữ được MongoDB cung cấp tại địa chỉ http://www.mongodb.org/display/DOCS/Drivers

Ta chọn mục PHP driver để download phiên bản mới nhất.



Figure 2-3. A list of currently available language drivers for MongoDB

Chọn đúng gói dành cho HDH và làm theo hướng dẫn của nó:

- copy file php_mongo.dll vào thư mục "ext" trong thư mục cài đặt PHP (VD: "xampp/php/ext/".
- thêm dòng code "extension=php mongo.dll" vào file php.ini.
- test bằng lenh phpinfo(): nếu thành công driver thì sẽ hiện lên phần mongo

b. Thao tác kết nối Database với PHP

```
<?php
//mongod --dbpath "c:\data"
// connect
phpinfo();die;
$m = new Mongo();
//chú ý: sau khi setup driver cho php (trong apache) thì mới sử dụng
được class Mongo()
// select a database
$db = $m->nguyen;
// select a collection (analogous to a relational database's table)
```

```
$collection = $db->user;
// add a record
$obj = array( "username" => "admin", "pass" => "123456" );
$collection->insert($obj);
// find everything in the collection
$cursor = $collection->find();
// iterate through the results
foreach ($cursor as $obj) {
   echo $obj["username"] . "\n";
}
?>
```

Một vài tham số cấu hình ta phải để ý giữa MongoDB và PHP

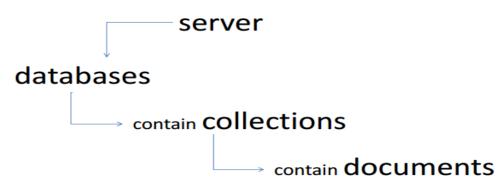
mongo

MongoDB Support	enabled
Version	1.0.6

Directive	Local Value	Master Value
mongo.allow_persistent	On	On
mongo.auto_reconnect	On	On
mongo.chunk_size	262144	262144
mongo.cmd	\$	\$
mongo.default_host	localhost	localhost
mongo.default_port	27017	27017
mongo.utf8	1	1

Figure 2-4. Displaying your MongoDB information in PHP

Chương 3: Mô hình dữ liệu



1. Database

- Một hệ thống Mongo chứa tập các database. Các database trong server độc lập với nhau và dữ liệu của mỗi database được lưu trữ một cách riêng biệt, an toàn và thuận tiên cho việc quản lý dữ liệu.
- Một databse chứa một hoặc 1 tập các collection, và được thiết lập một tùy chọn các thông tin bảo mật để kiểm soát truy cập.
 - Metadata: là nơi chứa toàn bô thông tin tổng quát của database:
 - o namespaces: danh sách tất cả các namespace.
 - o index: danh sách các chỉ mục
 - o profile: thông tin profile.
 - O User: danh sách các user được truy cập vào database.

2. Collection

- <u>Khái niệm:</u>Collection trong MongoDB bản chất được gọi là nhóm các document. Một collection chứa tập các document, Ta có thể xem nó tương tự như các table(bảng) trong cơ sở dữ liệu quan hệ.
- <u>Đặt tên:</u> của collection nên được bắt đầu bởi các chữ cái hoặc dấu gạch dưới và có thể chứa các kiểu số. các collection có thể được cấu tạo trong các namespace (là tên của một nhóm các collection). Ví dụ: ta có thể định nghĩa các collections Posts, Authors trong namespace Blog: blog.post, blog.Authors.
- <u>Bản chất của Collection:</u> tập hợp các văn bản BSON (BSON documents). Nhưng trong MongDB khác biệt là các document không tuân theo cấu trúc tức là các document trong 1 collection không có cấu trúc cố định như nhau (không cần lưu các cột giống nhau), như vậy ta không cần phải định nghĩa thành phần các cột trong 1 collection như trong cơ sở dữ liệu quan hệ.

Lưu ý:

- Các collection được tạo ra khi 1 document đầu tiên được tạo ra.
- Kích thước tối đa của tên 1 collection là 128 ký tư.

3. Document (object)

- Nhắc đến document là ta nghĩ đến 1 văn bản có cấu trúc. Nhưng đối với MongoDB thì không bó buộc 1 document phải theo một cấu trúc dữ liệu nào đó. Chính xác là 1 document như là 1 đối tượng dữ liêu.
 - ♣ Các document trong MongDB là JSON. Đặc biệt có định dạng BSON(Binary JSON).
- ♣ kích thước: Hiện tại thời điểm này (version 1.7/1.8), MongDB giới hạn kích thước của 1 document là 16MB.

a. BSON

BSON là dãy tuần tự nhị phân được mã hóa như tài liệu JSON. BSON giống như JSON, hỗ trợ nhúng các đối tượng và mảng trong các đối tượng và mảng khác.

MongoDB dùng BSON khi lưu trữ dữ liệu và là định dạng các document khi chúng được di chuyển trên network.

4. Field

- **Khái niệm:** Field là một thuật ngữ trong MongDB, nó chứa cặp thông tin key, và value, nó gần như thuộc tính trong lập trình hướng đối tượng có tên và giá trị.
 - Một document chứa tập các trường (field).
 - Key là tên của trường (field), có kiểu chuỗi.

- Value là giá trị có thể có kiểu sau:
- Kiểu cơ bản: string(kiểu chuỗi), integer (kiểu số nguyên), float (kiểu số thực), binary(kiểu nhị phân)...
 - Kiểu văn bản.
 - Kiểu mảng các phần tử.
 - ♣ _id Field
- o _id Field là 1 trường đặc biệt luôn có trong 1 document, dùng để xác định duy nhất của một document đó. Giá trị của trường _id có kiểu bất kỳ.
- Khi người dụng tạo một document mà không cung cấp trường _id thì , cơ sở dữ liệu tự động phát sinh _id cho document đó và lưu vào cơ sở dữ liệu

Chương 4: Thao tác với dữ liệu

1. Shell

Shell là phương tiện cho phép người dùng thực hiện các lệnh trong mongodb bằng command line. Shell được dùng để:

- Kiểm tra nội dung database.
- Kiểm tra các truy vấn.
- Tạo bảng mục.
- Thực hiện các chức năng của admin.

2. Kết nối database

♣ Server trong mongodb: là tiến trình chạy trên server, suốt quá trình hoạt động của nó là chờ các kết nối từ các người dùng. Để start server gõ lệnh chạy file mongod.exe bằng commandline:

```
-/$ ./mongod

#

# some logging output

#

Tue Mar 9 11:15:43 waiting for connections on port 27017

Tue Mar 9 11:15:43 web admin interface listening on port 28017
```

Câu lênh kết nối đến database

```
mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database]
[?options]]
```

Chú giải:

- Username, password: tên usernam, mật khẩu người dùng mà kết nối vào database cần kết nối.
- Host, port: địa chỉ và port của server mà chứa database cần kết nối. nếu câu lệnh không chỉ rõ port thì port mặc định sẽ là 27017.
- Database: tên cơ sở dữ liệu mà cần kết nối. nếu không cung cấp database thì hệ thống hiểu ngầm là admin database.
- Options: các tùy chỉnh khi kết nối. các tùy chỉnh có thể là:
- O Replica set: tên tập các bản sao
- o Single server
- Cấu hình: chế độ an toàn, thời gian kết nối mở trước thời gian timeout, ...
- ♣ Ví du:
- Kết nối database admin ở local host, port mặc định:

```
mongodb://localhost
```

 Kết nối database basz, với username: fred, password: foobar, kết nối trên localhost với port mặc định:

```
mongodb://fred:foobar@localhost/baz
```

3. Lệnh thao tác cơ bản với collection

H Ghi chú: câu lệnh tạo mới collection không khai báo các thuộc tính vì bản chất của mongoDB là cấu trúc các document không cố định, các document có thể có các field khác nhau.

Stt	Tên câu lệnh	Cú pháp	Diễn giải
1	Tạo	>db.createCollection("collectionName")	Db: database đang dùng
	collection		collectionName: tên
			collection muốn tạo.

2	Đổi tên	>db.oldname.renameCollection("newName")	db : database đang dùng
	collection		oldname: tên collection
			cần đổi tên
			newName: tên mới muốn
			đổi.

4. Thao tác trên dữ liệu.

a. Inserting

Stt	Tên câu lệnh	Cú pháp	Diễn giải	Ví dụ
1	Thêm	>db.[collectionName].i	[collectionName]: tên	Db.user.insert({a:3,
	mới 1	nsert({key1:value1,	collection muốn thêm	b:5})
	docume	key2:value2,})	document.	
	nt		Key:value: tên key và giá trị	
2	Thêm	>[arrayName] =	[arrayName]: mång chứa các	>batch = [{:name =>
	mới	[{document1,	phần tử document cần insert.	"mongodb"}, {:name
	nhiều	document2,}]	Cơ chế: tạo mảng các phần tử	=> "mongoid"}]
	docume	>db.collection.insert([ar	document, sau đó thự hiện	>db.collection.insert(
	nt một	rayName]	lệnh insert tham số đầu vào là	batch)
	lần.		mảng chứa các document.	

b. Removing

Stt	Tên câu lệnh	Cú pháp	Diễn giải	Ví dụ
1	Xóa toàn bộ document	>db.[collectionName].r emove({})	[collectionName]: tên collection muốn xóa document.	>db.things.remove({})
2	Xóa document thỏa điểu kiện	>db.[collectionName].r emove({điều kiện})	Điều kiện chọn ra các document cần xóa.	db.things.remove({n:1}); // xóa document nào có thuộc tính n == 1

c. Updating

Lệnh update có cú pháp chung như sau:

Db.collection.update(criteria, objNew, upsert, multi)

- Criteria: câu truy vấn lấy ra các record cần update.
- objNew: đối tượng update mới, hoặc câu lệnh chỉnh sửa đối tượng.
- upsert: mang giá trị true/false, nếu lệnh cần upsert thì có nghĩa, nếu không có record nào thỏa lệnh criteria để update thì sẽ thực hiện lệnh insert 1 document. Ngược lại thì không cần insert.
- Multi: nếu giá trị này được bật lên true thì có nghĩa chỉ update 1 doucment duy nhất trong số các document thỏa điểu kiện.
 - ➤ Ví du:

>Var member = db.users.findOne({"name": "TOM"})

>member.age++

>Db.user.update({"name":"TOM"}, member, true, false)

Member là document mới được chỉnh sửa, câu lệnh update muốn tìm ra các document nào mà trường(field) có key: "name", và giá trị là "TOM" thì sẽ chỉnh sửa thành giống như document member. Upsert mang giá trị true có nghĩa nếu không tìm thấy document nào thỏa "name": "TOM" thì sẽ thêm mới một document giống như member vào cơ sở dữ liệu. multi mang giá trị false tức là sẽ chỉnh sửa hết toàn bộ các document nào thỏa điểu kiên.

♣ Các thao tác trong update:

Stt	Tên	Cú pháp	Diễn giải	Ví dụ
	thao tác			
1	Tăng	>\$inc:{key: n}	Tăng giá trị của key lên n đơn	>db.people.update({
	giá trị		vį	name:"Joe" }, { \$inc:
	của một			{ n:1 } });
	trường			
2	Gán giá	>\$set : { field : value}	Gán giá trị cho field giá trị	>db.people.update({
	trị cho	>\$set. { field. value}	value.	name:"Joe" }, { \$set:
	một			{ n:1 } });
	trường			
3	Xóa	>\$unset: {field: 1}	Xóa trường field trong	>db.users.update({"na
	trường	>\$uiiset. {field. 1}	document	me" : "joe"},
				{"\$unset" :
				{"favorite book":
				1}})

4	Thêm giá trị vào 1 trường, trường này là một mảng	>\$push:{field:value}	Thêm một giá trị vào cuối mảng Điều kiện : Field phải có kiểu mảng.	>db.papers.update({"a uthors cited" : {"\$ne" : "Richie"}}, {\$push : {"authors cited" : "Tom"}})
5	Xóa một phần tử trong mảng	>{\$pop:{field: 1}} >{\$pop:{field:-1}}	Giá trị 1: xóa cuối mảng Giá trị -1: xóa đầu mảng	>db.lists.update({}, {"\$pop" : {"todo" :1}})
6	Xóa toàn bộ các giá trị của field nếu có trong tập	>\$pull:{field:_value}	_ value : tập hợp giá trị mà muốn xóa, _value này có thể là điểu kiện.	>db.lists.update({}, {"\$pull" : {"todo" : "laundry"}})

d. Queries

Câu truy vấn có cú pháp chung như sau:

Db.collection.find({*criteria*}, {field1: 0, field2: 1, ...})

- Criteria: điều kiện của truy vấn.
- {field1: 0, field2: 1, ...}: các trường muốn lấy ra, hoặc không muốn lấy ra, field được gán 0: loại trường field đó ra khỏi danh sách các trường muốn lấy ra cho người dùng. Nếu 1 thì có ý nghĩa ngược lại.
 - ♣ Ví dụ:

 - ✓ db.users.find() // lấy toàn bộ danh sách các document
 ✓ db.users.find({last_name: 'Smith'}, {'ssn': 1});// lấy các document thỏa last_name == "Smith" và truy xuất trường 'ssn' ra màn hình.

Stt	Tên	Cú pháp	Diễn giải	Ví dụ
	thao tác			

1	Lệnh so sánh	{\$gt: value} //> {\$lt: value} // < {\$gte: value} // >= {\$lte: value} //<=	Từ khóa \$gt, \$lt, \$gte, \$lte thể hiện so sánh theo thứ thự: >, <, >=, <=	>db.things.find({j: {\$lt: 3}}); >db.things.find({j: {\$gte: 4}});
2	Thao kiểm tra mảng chứa mảng cho sẵn	field: { \$all: [valueSet] } });	Kiểm tra mảng field có chứa các giá trị trong tập value set	>db.things.find({ a: { \$all: [2, 3] });
3	xem có tồn tại trường đó hay không	field : { \$exists : true }	hay không	>db.things.find({ a : { \$exists : true } });
4	Lấy số dư của phép chia	field: { \$mod:[divisor, Remainder] }	kiểm tra field chia cho divisor có số dư là remainder không	>db.things.find({ a : { \$mod : [10 , 1] } })
5	lệnh khác (#)	field : { \$ne : value }	kiểm tra xem field có khác value không	>db.things.find({ x : { \$ne : 3 } });
6	lệnh or	\$or : [{ field1 : value } , { field2 : value, }]	điểu kiện or	>db.foo.find({ \$or : [{ a : 1 } , { b : 2 }] })
7	lệnh and	\$and:[{ field1: value}, { field2: value,}]	điều kiện and	>db.foo.find({ \$and: [{ a: 1 }, { a: { \$gt: 5 } }] })

e. Currsor methods

Stt	Tên câu	Ý nghĩa	Ví dụ
	lệnh		

1	Count()	Đếm số lượng record	>nstudents = db.students.find({'address.state' : 'CA'}).count();
2	Limit(n)	Lấy n record	>db.students.find().limit(10).forEach(function(student) { print(student.name + ""); });
3	Sort()	Sắp xếp các record theo tăng dần hoặc giảm dần	db.myCollection.find().sort({ ts:-1 }); // sắp xếp theo ts, giảm dần
4	batchSize()	Chỉ ra số lượng document mỗi lần trả về cho client	>db.myCollection.find().batchSize(10);

Chuong 5: GridFS

1. Yêu cầu đặt ra

Bạn cần lưu trữ các file có kích thước lớn như: video, images, nhưng kích thước của một document quá nhỏ (v1.7/1.8 là 16MB). Để lưu trữ các dữ liệu này, MongoDB có giải pháp đó là cơ chế GridFS.

2. Tổng quát GridFS

Là một kỹ thuật đặc biệt dùng cho việc lưu trữ các file lớn trong MongoDB. Nó phân chia các đối tượng lớn thành các khối nhỏ, thường kích thước 1 khối là 256k. các khối nhỏ được lưu dưới dạng các document trong 1 chunks collection. Các thông tin metadata về file bao gồm thông tin filename, kiểu dữ liệu, và thông tin tùy chỉnh được lưu dưới dạng document trong collection file. Vì thế bất kỳ thông tin của file nào được lưu trữ dùng kỹ thuật GridFS thì đều có collection files, và các chunk (khối dữ liệu được phân chia) nằm trong chunk collection.

3. Khi nào dùng GridFS?

Ta dùng GridFS trong các trường hợp sau:

- o Có nhiều file. GridFS thường quản lý các file lớn tốt hơn là file systems.
- O Các file được người dùng tải lên: khi người dùng tải file lên thường có xu hướng là tải nhiều file, và muốn chúng được sao chép backup. GridFS là lựa chọn tốt để lưu trữ các file này vì chúng có thể quản lý

tốt dữ liệu cho người dùng.

O Các file thường bị thay đổi. nếu ta có các file hay bị thay đổi. ta nên dùng GridFS vì, khi thay đổi file thì GridFS sẽ đồng bộ các client lưu trữ file mà ta không cần phải làm việc này.

Chuong 6: Replication

Giống như hệ quản trị CSDL quan hệ, MongoDB hỗ trợ replication của database theo thời gian thực hoặc gần với thời gian thực. Replication trong MongoDB rất đơn giản để cài đặt và sử dụng. Đó cũng là một trong những điểm đặc biệt của hệ quản trị cơ sở dữ liệu thế hệ mới, thế hệ cho nền tảng web 2.0 và công nghệ lưu trữ điện toán đám mây.

Có nhiều kịch bản đặt ra cho việc sử dụng Replication. MongoDB hỗ trợ đầy đủ các trường hợp Replication mà thực tế đặt ra.

Hiện nay MongoDB hỗ trợ các kịch bản Replication sau:

- Single master/single slave replication
- Single master/multiple slave replication
- Multiple master/single slave replication
- Cascade replication
- Master/master replication

- Interleaved replication
- Replica pairs (legacy MongoDB clustering)
- Replica sets (advanced MongoDB clustering)

1. Mục đích Replication của MongoDB

Replication đem lại nhiều lợi ích tùy thuộc và loại Replication mà bạn chọn và những tùy chọn mà chúng ta chọn. Nhưng một trong những lợi ích thông thường của Replication là dữ liệu luôn sẵn sàng mọi lúc mọi nơi khi chúng ta cần.

2. Khi nào thì sử dụng Replication

Replication là giải pháp được ứng dụng cho môi trường phân phối dữ liệu trên nhiều Server, chính vì vậy mà sử dụng chúng khi:

- Sao chép và phân phối dữ liệu trên nhiều Server khác nhau
- Phân phối bản sao dữ liệu theo lịch trình nhất định.
- Phân phối dữ liệu vừa thay đổi trên nhiều Server khác nhau.
- Cho phép nhiều người dùng và nhiều Server kết hợp dữ liệukhacs nhau một cách thống nhất mà không sợ mất dữ liệu.
- Xây dựng CSDL sử dụng cho những ứng dụng trực tuyến hay ngoại tuyến
- Xây dụng ứng dụng Web khi người dùng cần trình bày một số lượng lớn dữ liệu

Chuong 7: Sharding

1. Sự cần thiết phải có cơ chế Sharding

Khi Word Wide Web bùng nổ vào những năm 90 thế kỷ XX và phát triển đến nay với nền tảng thế hệ web 2.0. Trong quá trình phát triển đó, sự tăng lên của người dùng, các website và số lượng thông tin lưu trữ khổng lồ đặt ra một bài toán phân tán dữ liệu. Tức là sự lưu trữ dữ liệu, thiết đặt các server phục vụ không còn nằm một chỗ nữa, thực tế cho thấy sự phát triển thương mại của các tổ chức liên doanh, liên kết tầm vùng miền, quốc gia, khu vực.... cộng với nền tảng thế hệ web 2.0 càng làm người ta thấy sự cần thiết phải lưu trữ dữ liệu ở nhiều nơi để phục vụ người dùng một cách tốt nhất. Như vậy bài toán phân tán ra đời.

MongoDB được phát triển dựa trên thực tế yêu cầu một Hệ quản trị Cơ sở dữ liệu mang những đặc trưng mới như NoSQL, tính tái tạo, tính phân tán, tính đơn giản và hiệu suất cao. Vì vậy, cơ chế Sharding trong MongoDB khá đơn giản và linh hoạt.

2. Phân mảnh ngang và phân mảnh dọc

Sự phân chia dữ liệu là cơ chế chia cắt dữ liệu thành nhiều datasore độc lập. Mỗi datastore đó có thể được lưu trữ trên một máy hoặc trên các máy khác nhau đặt tại nhiều nơi khác nhau.

a. Phân mảnh dọc

Theo cách nhìn của cơ sở dữ liệu truyền thống, data được lưu trẽ thành các dòng và các cột. Sự phân mảnh dọc là việc chia các record thuộc các column khác nhau và lưu trữ riêng biệt những phần đó trong

những table hoặc collection.. Một ví dụ điển hình cho sự phân mảnh dọc là việc thiết kê cơ sở dữ liệu quan hệ sử dụng việc join các bảng với mối quan hệ 1-1 để cho ra kết quả thỏa mãn truy vấn. Thay vì chia thành các bảng như vậy thì ta cũng có thể để tất cả trong 1 bảng. Tuy nhiên các tiếp cận của CSDL quan hệ và CSDL NoSQL là khác nhau. Vì vậy ảnh hưởng đến việc thiết kế CSLD.

b. Phân mảnh ngang

Phân mảnh ngang là cơ chế chủ yếu trong Sharding của MongoDB. Sharding cho phép ta cắt một collection ngang trên nhiều server. Một ví dụ cho Sharding là khi một collection USER được chia thành các shard theo lastname của USER. Giả sử như shard1 bao gồm những USER có lastname từ A-G được đặt trên một server, shard bao gồm những USER có lastname từ H-M trên một server khác và phần còn lại nằm trên một server khác nữa. Như vậy dữ liệu đã bị phân tán thành nhiều nơi.

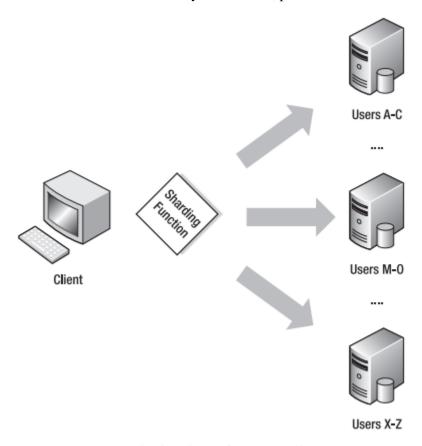


Figure 12-1. Simple sharding of a User collection

3. Thực hiện sharding với MongoDB

Có 3 thành phần quan trọng trong việc sharding với MongoDB mà chúng ta cần biết.

> Shard

Shard là thành phần nắm dữ các dữ liệu bị chia cắt của một collection. Một Shard là một

mongod server đơn (dùng cho việc phát triển/testing hoặc replica set (cho production))

Mongos

Mongos như một router nhận yêu cầu từ client và hồi đáp kết quả. Nó không lưu trữ bất kì data nào cả. Mongos định tuyến yêu cầu truy cập vào shard nào trong n shard của mô hình phân tán và trả kết quả cho client.

Config server

Config server lưu trữ các cấu hình cluster. Ví dụ như dữ liệu được lưu trữ ở trong những shard nào ? Nó giúp cho mongos quyết định được việc định tuyến router

MongoDB sử dụng cơ chế proxy để hỗ trợ việc sharding, khi đó mongos như một controller cho nhiều mongod (based shard server). Khi đó các yêu cầu từ người dùng (select, insert, update, delete) sẽ được gọi đến mongos và mongos có trách nhiệm đưa những yêu cầu đó đến những shard nào đó liên quan đến tác vụ yêu cầu

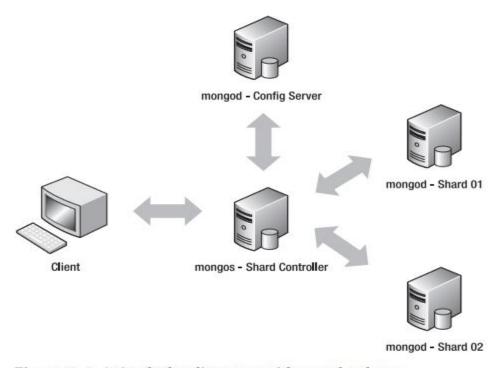


Figure 12-2. A simple sharding setup without redundancy

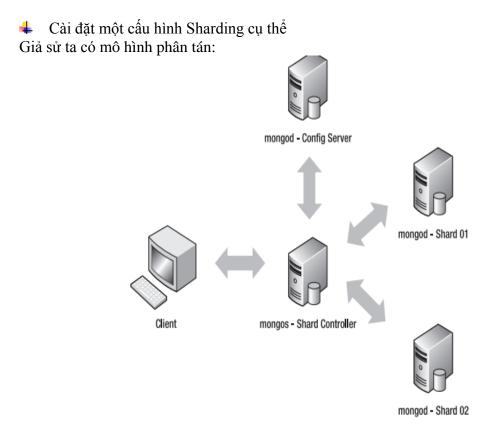
MongoDB cung cấp cơ chế sharding mức collection, không cung cấp cơ chế sharding mức database. Trong nhiều hệ thống, chỉ có một hoặc hai collection mà dữ liệu phát triển đến một mức nào đó mới đảm bảo sharding mang lại hiệu quả. Tức là không phải cứ sharding thì luôn luôn tốt, luôn luôn đảm bảo hiệu quả. Điều đó là không chắc.

Việc sharding trong MongoDB chỉ mang lại hiệu quả thiết thực khi dữ liệu chúng ta phát triển đến mức đủ lớn và có kịch bản sharding hợp lý.

Hệ thống sharding sử dụng hàm sharding key để sắp xếp dữ liệu vào các chunk, chunk là các block lưu trữ document. Mỗi chunk lưu trữ document với các phần được cắt ra theo tiêu chí shard key. Cấu hình server sẽ thiết lập lưu trữ thông tin xem chunk nào được lưu ở server nào. Điều này là một đặc điểm quan trọng vì nó cho phép ta thên hoặc bớt các shard mà không cần backup và restore database.

Khi add thêm một shard vào cluster, hệ thống sẽ phân phối lại các chunk, chia dữ liệu ra để cải thiện hiệu suất, cũng như thế, khi remove một chunk nào đó thì sharding controller(mongos) sẽ ghép các chunk lại và phân phối lại dữ liệu.

Chú ý: Nếu chúng ta không chỉ ra sharding key thì MongoDB sẽ tự động lấy _id của collection để phân tán document vào các shard.



Chuẩn bị các thư mục và thông tin cấu hình như sau:

Service	Daemon	Port	Dbpath
Shard Controller	mongos	27021	N/A
Config Server	mongod	27022	/db/config/data
Shard0	mongod	27023	/db/shard1/data
Shard1	mongod	27024	/db/shard2/data

Lưu ý: Cứ mỗi bước sau phải mở mới 1 terminal chứ không được tắt services của nó.

Buóc 1: Config server

Set biến môi trường PATH=C:\mongodb\bin

Trong teminal gõ:

C:\Users\PhamDanhVu> mongod.exe --port 27022 --dbpath /db/config/data --configsvr

Buốc 2: Setup the shard controller (Mongos)

Trong teminal gõ:

C:\Users\PhamDanhVu>mongos.exe --configdb localhost:27022 --port
27021 --chunkSize 1

Bước 3: Bring up the two servers

-- Tạo thêm thư mục db/shard0/data

Mở terminal thứ 1: gõ

C:\Users\PhamDanhVu>mongod.exe --port 27023 --dbpath /db/shard0/data --shardsvr

Mở termianl thứ 2 : gõ

C:\Users\PhamDanhVu>mongod.exe --port 27024 --dbpath /db/shard1/data -shardsvr

Bước 4: Config location two shard with sharding system.

C:\Users\PhamDanhVu> mongo.exe localhost:27021

> use admin

```
switched to db admin
> db.runCommand( { addshard : "localhost:27023", allowLocal :
true } )
    { "added" : "localhost:27023", "ok" : 1 } // ket qua thong bao
> db.runCommand( { addshard : "localhost:27024", allowLocal :
true } )
    { "added" : "localhost:27024", "ok" : 1 } //ket qua thong bao
```

Buóc 5: View infor config the shards

```
> db.runCommand({listshards:1})
{
    "shards" : [
    {
        "_id" : "shard0",
        "host" : "localhost:27023"
    },
    {
        "_id" : "shard1",
        "host" : "localhost:27024"
    }
    ],
    "ok" : 1
}
```

Bước 6: Tạo mới 1 Database tên là testdb, enable sharding testdb, trong testdb tạo 1 collection có tên là testcollection

- Tao Database testdb

```
> testdb = db.getSisterDB("testdb")
```

testdb

- Enable sharding testdb

```
> db.runCommand({ enablesharding: "testdb"})
{ "ok" : 1 }
```

- Tạo collection testcollection và shardcollection với shard key là testkey

```
> db.runCommand({ shardcollection : "testdb.testcollection", key
: {testkey : 1}})
    { "collectionsharded" : "testdb.testcollection", "ok" : 1 }
```

Ở đây ta sử dụng PHP để random dữ liệu khoảng 200 000 document.

Bước 6.1: Cấu hình PHP - Mongo

- → Phải add driver PHP và Mongo:
- Lên trang Mongodb.org kiếm driver
- \rightarrow Down driver
- → Giải nén →
- Add 2 cái file .dll đó vào thư viện của PHP ở đường dẫn C:\wamp\bin\php\php5.3.8\ext
- → bật WARM và sửa trong file php.ini: thêm 2 dòng này vào

```
extension=php_mongo-1.2.10-5.3-vc9-nts-x86_64.dll
extension=php_mongo-1.2.10-5.3-vc9-x86_64.dll
```

Restart services

Bước 6.2: Tạo file Php random dữ liệu

```
<?php

// Open a database connection to the mongos daemon

$mongo = new Mongo("localhost:27021");

// Select the test database

$db = $mongo->selectDB('testdb');

// Select the TestIndex collection
```

```
$collection = $db->testcollection;
  for (\$i=0; \$i < 200000; \$i++) {
  $data=array();
  data['testkey'] = rand(1,100000);
  $data['testtext'] = "Because of the nature of MongoDB, many of
the more "
  . "traditional functions that a DB Administrator "
  . "would perform are not required. Creating new databases, "
  . "collections and new fields on the server are no longer
 necessary, "
  . "as MongoDB will create these elements on-the-fly as you
access
 them."
  . "Therefore, for the vast majority of cases managing databases
 and "
  . "schemas is not required.";
  $collection->insert($data);
  ?>
```

Lưu file Php và chạy file đó lên để insert dữ liệu vào testdb

Bước 7: Kiểm tra tình trạng dữ liệu insert và dữ liệu phân tán.

```
C:\Users\PhamDanhVu> mongo.exe localhost:27021

>use testdb

>db.testcollection.count()

200000 // kết quả
```

```
C:\Users\PhamDanhVu> mongo.exe localhost:27023

>use testdb

>db.testcollection.count()

149875

$mongo localhost:27024

>use testdb

>db.testcollection.count()

25125
```