

Introducció a NoSQL

Capacitació Tecnològica per a Professionals i
Empreses



Index

- 1. MongoDB**
- 2. Exercicis sobre MongoDB**



1. MongoDB

- 1.1** Introducció
- 1.2** Taula comparativa
- 1.3** Model
- 1.4** Propietats
- 1.5** Consultes i modificacions
- 1.6** Referència de la shell i operacions
- 1.7** Comparativa amb SQL
- 1.8** Instal·lació
- 1.9** Driver
- 1.10** Utilitat per l'aprenentatge
- 1.11** Desplegament



1.1 Introducció

MongoDB es una base de dades basada en documents (document database) que promet un alt rendiment, alta disponibilitat i una escalabilitat senzilla.

- L'esquema és flexible i dinàmic, i es basa en documents. Aquest mètode d'emmagatzematge cuadra molt bé amb els tipus de dades dels llenguatges de programació (principalment dels orientats a objectes).

- Les relacions son substituïdes per estructures del tipus array i documents, que poden ser contingudes dins alters documents.







1.1 Introducció

- *Gràcies als continguts recursius, i a l'existència d'índexs per totes les estructures disponibles, s'augmenta el rendiment de lectures i escriptures.*
- *El desplegament es fa en forma de servidors replicats, implementat un master failover.*
- *L'escalabilitat s'aconsegueix a través d'una estructura de sharding.*
- *El format intern de desat de dades es BSON: Es tracta d'una codificació binària sobre el format JSON*



1.2 Taula comparativa

Engine	MongoDB	Cassandra	Redis	HBase
Imatge				
Desenvolupador	10gen	Apache Software Foundation	Salvatore Sanfilippo	Apache Software Foundation
Data de llançament	2009	2008	2009	2010
Usuaris destacats	Foursquare, Craigslist	Twitter, Digg	StackOverflow	Facebook
Tipologia	Document	Columna	Key-value	Columna



1.2 Taula comparativa

Llicència	AGPL	Apache	BSD	Apache
Llenguatge	C++	Java	C/C++	Java
Exemple d'ús	CMS, sistemes de vots, desat de comentaris	Banca, logging	Stock prices, Analítics, dades en temps real	Missatgeria en temps real
Millor utilitzar en...	Consultes dinàmiques, escriptures freqüents, poques consultes estadístiques	Grans volums d'escriptures en front a les lectures	Canvis freqüents de dades, escriptura freqüent, poques consultes estadístiques	Lectures aleatòries en grans bases de dades



1.2 Taula comparativa

Punts clau	Manté algunes propietats d'SQL, com els índexs i les consultes	Alta disponibilitat: es tracta d'un encreuament entre les idees de BigTable i Dynamo	Molt ràpid	Modelat sota la direcció de BigTable. Dissenyat per desar ingents quantitats de dades.
Control de concurrència	Locks	Multiversion concurrency control (MVCC)	Locks	Locks
URL	mongodb.org	cassandra.apache.org	redis.io	hbase.apache.org
Procés de replicació	Asíncron	Asíncron	Asíncron	Asíncron



1.2 Taula comparativa

Sistemes operatius	Linux, Mac, Windows	Linux, Mac, Windows	Linux, Mac, Windows	Linux, Mac, Windows
Desat de dades	Disc	Disc	RAM	Hadoop
Característiques	Consistència, tolerància a particions, persistència	Alta disponibilitat, tolerància a particions, persistència	Consistència, tolerància a particions, persistència	Consistència, tolerància a particions, persistència



1.3 Model

L'estructura d'una base de dades es conforma en base a la següent imatge:

Cluster -> Database -> Collection -> Document -> Key-value pairs

- **Col·lecció:** *Agrupacions de documents BSON. No imposen cap esquema, i son els homòlegs de les taules en sistemes relacionals.*

- **Document:** *Unitat bàsica d'informació, sense esquema determinat, ni a nivell de camps, ni a nivell de tipus de dades. Poden interpretar-se com files d'una taula en el model relacional.*

- **Clau-valor:** *Son cadascuna de les entrades dins el document, i componen el seu contingut*



1.4 Propietats

L'atomicitat en MongoDB, queda garantida a nivell d'operacions sobre un document.

Igual que altres solucions, MongoDB introdueix el concepte de TTL.

Els índexs en MongoDB son molt similars als que trobem en SQL.

- *Automaticament es crea un índex sobre el camp `_id` pregenerat*
- *Cada índex requereix 8KB*
- *Afegir un índex te un impacte negatiu en les escriptures*

Quan un document creix (updates), automàticament es re-localitza en disc. Aquest operació pot ser costosa.



1.5 Consultes i modificacions

Lectures

<http://docs.mongodb.org/manual/crud/>

Les consultes es fan a nivell de col·lecció

Tenim disponibles les operacions `find()` i `findOne()`

Els paràmetres que es passen en aquestes funcions delimiten la cerca.

La signatura és:

```
db.mycollection.find( <query>, <projection>)
```

Retorna un cursor

`findOne` retorna un objecte directament

* Projection ens diu quins camps volem en el resultat



1.5 Consultes i modificacions

Exemple:

```
db.inventory.find( { type: 'food', price: { $lt: 9.95 } }, { item: 1, qty: 1 } )
```

Exemple d'ús d'un cursor

```
var myCursor = db.inventory.find( { type: 'food' } );  
var myDocument = myCursor.hasNext() ? myCursor.next() : null;
```

Les consultes son riques en semàntica

Aquesta semàntica es tradueix amb el símbol \$ i la paraula clau que defineix la funció requerida.



1.5 Consultes i modificacions

Obviament, podem consultar subdocuments, amb notació aniuada:

```
db.inventory.find( {  
    producer: {  
        company: "ABC"  
    }  
})
```

En la mateixa línia, quan es contenen arrays, també poden ser consultades

Per crear índexs

```
db.collection.ensureIndex( { <field1>: <order>, <field2>: <order>, ... } )
```

Cal destacar que també existeix suport per agregació



1.5 Consultes i modificacions

Esriptures

Tenim disponibles els següents mètodes sobre collection:

- insert
- update
- save
- findAndModify
- remove

Tenim disponible la opció de bulk inserts, tot i que no es garanteix l'atomicitat.

Podem ampliar informació en
<http://docs.mongodb.org/manual/core/write-operations/>



1.6 Referències de la Shell i operacions

Operacions sobre col·leccions:

- *count*
- *createIndex*
- *drop*
- *find*

Operacions sobre cursors:

- *hasNext*
- *next*
- *size*



1.6 Referències de la Shell i operacions

Operacions sobre base de dades:

- *addUser*
- *changeUserPassword*
- *getLastError*
- *help*

Replicació

- conf
- help
- add
- remove



1.6 Referències de la Shell i operacions

Sharding

- addShard
- enableSharding
- status

Connexió

- *getDB*
- *connect*

Per aprofundir en tots els mètodes disponibles, teniu el següent recurs.
<http://docs.mongodb.org/manual/reference/method/>



1.6 Referències de la Shell i operacions

Exemples d'operacions disponibles

show collections: mostra les col·leccions disponibles

explain: després d'un find, ens dona informació de test sobre la consulta



1.7 Comparativa amb SQL

SQL

```
CREATE TABLE users (  
  id MEDIUMINT NOT NULL  
    AUTO_INCREMENT,  
  user_id Varchar(30),  
  age Number,  
  status char(1),  
  PRIMARY KEY (id)  
)
```

MongoDB

```
db.users.insert( {  
  user_id: "abc123",  
  age: 55,  
  status: "A"  
} )  
  
*  
db.createCollection("users")
```



1.7 Comparativa amb SQL

SQL

```
ALTER TABLE users  
ADD join_date DATETIME
```

MongoDB

```
db.users.update(  
  { },  
  { $set: { join_date: new  
Date() } },  
  { multi: true }  
)
```



1.7 Comparativa amb SQL

SQL

```
CREATE INDEX  
idx_user_id_asc  
ON users(user_id)
```

MongoDB

```
db.users.ensureIndex( { user_i  
d: 1 } )
```



1.7 Comparativa amb SQL

SQL

```
SELECT *  
FROM users  
WHERE status = "A"  
ORDER BY user_id DESC
```

MongoDB

```
db.users.find( { status:  
"A" } ).sort( { user_id: -1 } )
```



1.7 Comparativa amb SQL

SQL

```
SELECT COUNT(user_id)  
FROM users
```

MongoDB

```
db.users.count( { user_id:  
{ $exists: true } } )
```

*

```
db.users.find( { user_id:  
{ $exists: true } } ).count()
```




1.7 Comparativa amb SQL

SQL

```
DELETE FROM users  
WHERE status = "D"
```

MongoDB

```
db.users.remove( { status:  
"D" } )
```



1.8 Instal·lació

```
// Dependencies
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | sudo
tee /etc/apt/sources.list.d/10gen.list
sudo apt-get update
// Install
sudo apt-get install mongodb-10gen
// Config
vim /etc/mongodb.conf
// Start/stop server
sudo service mongodb [start/stop/restart]
// CLI
mongo
```



1.9 Driver

Tenim diverses opcions per utilitzar aplicacions en conjunt amb MongoDB. Per homogeneïtat, en aquest curs comptarem amb drivers (o interfícies) per Python.

pymongo

<http://api.mongodb.org/python/current/tutorial.html>

* Podeu trobar implementacions de drivers per MongoDB per molts llenguatges actuals



1.10 Utilitat per l'aprenentatge

Tenim disponible una eina online per aprendre i provar MongoDB.
Aquesta eina ens exposa una consola

- *Sense instal·lació*
- *Interactiva*
- *Disponible sense credencials*

Utilitzarem l'eina per familiaritzar-nos amb les comandes de consola de MongoDB

<http://try.mongodb.org/>



1.11 Desplegament

Aprofitem MongoDB per comentar el desplegament.

MongoDB suporta els desplegaments standalone

Tanmateix, el desplegament distribuït és el desplegament per defecte.

Moltes solucions NoSQL comparteixen conceptes:

- **Replicació**: proporciona un alt rendiment amb un failover automatitzat
- **Sharding**: particiona particionat de datasets grans sobre diverses màquines transparentment.

Les dues polítiques poden combinar-se.



1.11 Desplegament

Sharding:

Consta dels següents mòduls:

- **Shards:** son contenidors que contenen una porció de la col·lecció *de dades*.

- **Servidors de configuració:** *son instàncies de mongod que mantenen informació sobre el cluster, localitzant i mapejant les porcions de dades als shards.*

- **Instàncies mongo:** *son els punts d'accés de les aplicaciones. S'encarreguen d'enrutar les peticions als shards corresponents (les aplicacions no accedeixen directament als shards).*

El sharding es fa a nivell de col·leccions per defecte. No totes les col·leccions han de ser distribuïdes a nivell de sharding.



2. Exercicis sobre MongoDB

- 2.1 Implementació via shell
- 2.2 Implementació per aplicació
- 2.3 Benchmarking



2.1 Implementació via shell

En aquesta primera part pràctica, generarem una serie de projectes a partir de la consola del sistema que estem tractant.

- **Twitter:** generarem un clon del software conegut com Twitter. Com a definició, prendrem l'esquema d'usuari, tweet i seguidor.

- **Registre d'hotel:** necessitem un sistema d'informació per el registre d'un hotel. S'ha de contindre informació sobre totes les habitacions, així com un historial tant d'usuaris com d'ús.

- **Log enriquit:** volem tenir una peça de software que mantingui informació de log, concretament el còdi, una marca de temps, i un objecte, que contindrà informació arbitrària (fins i tot, altres objectes)



2.1 Implementació via shell

- **Item vault:** un banc d'informació sobre objectes varis dins diverses col·leccions. L'objectiu és donar la versatilitat a un negoci d'objectes estranys per poder tenir un sistema d'informació amb totes les característiques d'aquests.

- **Enfermatim:** base de dades d'informació sobre malalties. Aquest ha de contenir informació sobre geografia, causes, públic afectat, i estadístiques.

- **Workouts:** tenim una gran cadena de gyms, i volem mantenir una base de coneixement sobre les màquines que tenim, així com dels clients i el seu creixement personal (muscular, dietes, salut, dies d'entrenament...)

- **Job box:** es tracta d'un directòri d'ofertes de treball sobre tot el món. Volem informació sobre el treball, així com localització, empresa, perfils necessitats, i tota la informació de valor sota aquest concepte.



2.2 Implementació per aplicació

En la segona part, generarem els mateixos projectes anteriors, a partir del llenguatge de programació que nosaltres escollim i els seus bindings sobre el sistema que tractem.



2.3 Benchmarking

Finalment, en aquesta tercera part, generarem scripts que omplirán les nostres bases de dades, i farem benchmarking sobre els processos d'escriptura i de lectura.

Barcelona **a**ctiva



Ajuntament
de Barcelona

bcn.cat/barcelonactiva
bcn.cat/cibernarium