MongoDB

humongous database

En bref

- Developed par : MongoDB Inc., California
- Under licence : AGPL
- Current version : 2.6
- Written in C++

SQL VS noSQL

SQL	noSQL
Relational BDD	No FK, no joins
Based on tables	Based on collections, key-value pairs, graphs
Based on SQL language	Based on Unstructured Query Language, defined by each systems
Good for complex requests	Bad for complex requests
Vertically scalable	Horizontally scalable
Good for transactional purposes	Does not support transactions

Quick analogy with RDBMS

table ⇔ collection

row ⇔ BSON document

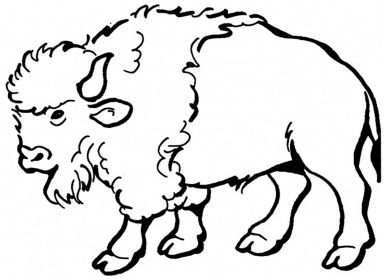
shard ⇔ instance

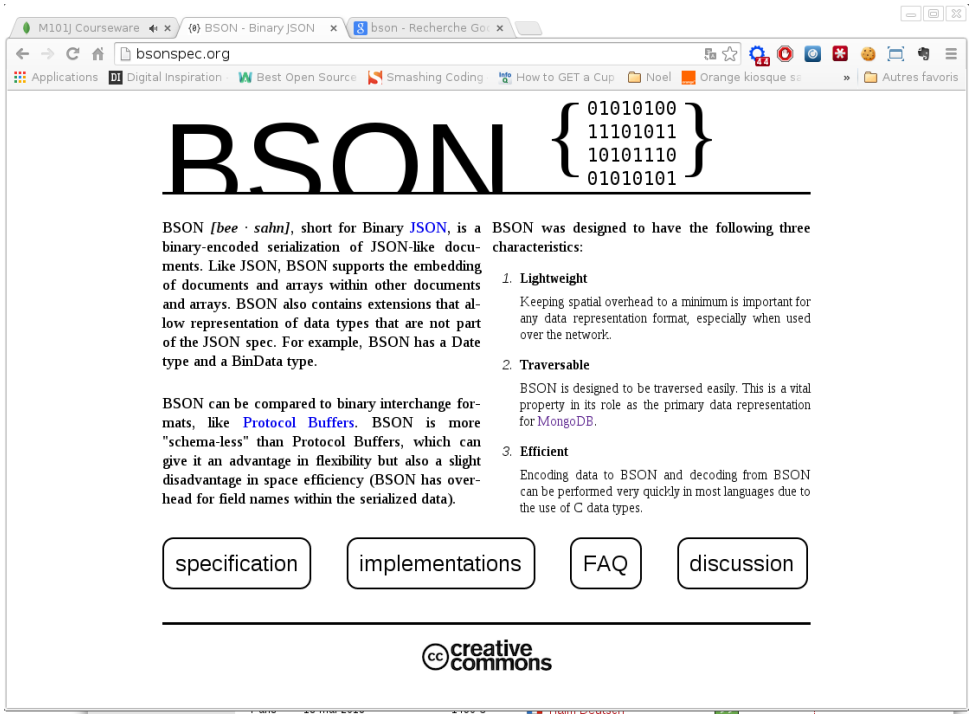
BSON ?

BSON

=

superset(JSON)





Extended types : utf8 strings, binary type, datetime, regular expressions, js code, double precision float, etc

CRUD

MongoDB provides a javascript shell to perform CRUD operations and to configure database

insert : db.collection.**insert**(doc)

select : db.collection.**find**()

where : db.collection.**find**(*conditions*)

update : db.collection.**update**(conditions, modif)

Operators : $gt, $or, $and, $in, $all, $set, $push, $pull

```
db.users.find( { friends : { $all : [ "Joe" , "Bob" ] },
favorites : { $in : [ "running" , "pickles" ] } } )
```

MongoDB cursor

In mongoDB, the result of a request is a cursor

```
cur = db.users.find();
cur.hasNext();
cur.next();
cur.sort({name : -1});
cur.sort({name : -1}).skip(30).limit(3);
cur.count();
```

Schema design

- Schemaless but mostly you use a schema
- Application driven schema
- Pre join/Embed data
- No constraints

NoSQL rule :

"Match the data access patterns of your application"

Schema design

```
{“_id” : “3266F86FD7FD”,
“postContent” : “J’aime les bananes”,
“comments” : [
  {author : “Jacky”, “email”:”j.terin@gmail.com”},
  {author : “Mich”, “email”:”m.terin@gmail.com”},
] ,
tags : [“eating”, “fruits”, “yellow”]
}
```

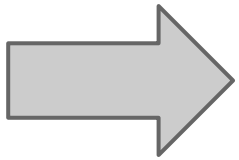
One to many relationship

people

{"_id" : "3266F86FD7FD",
"name" : "Patrick",
"city" : "NYC" }

cities

{"_id" : "NYC",
... }



But embedded arrays should be preferred when the many is not large

Many to many relationship

people

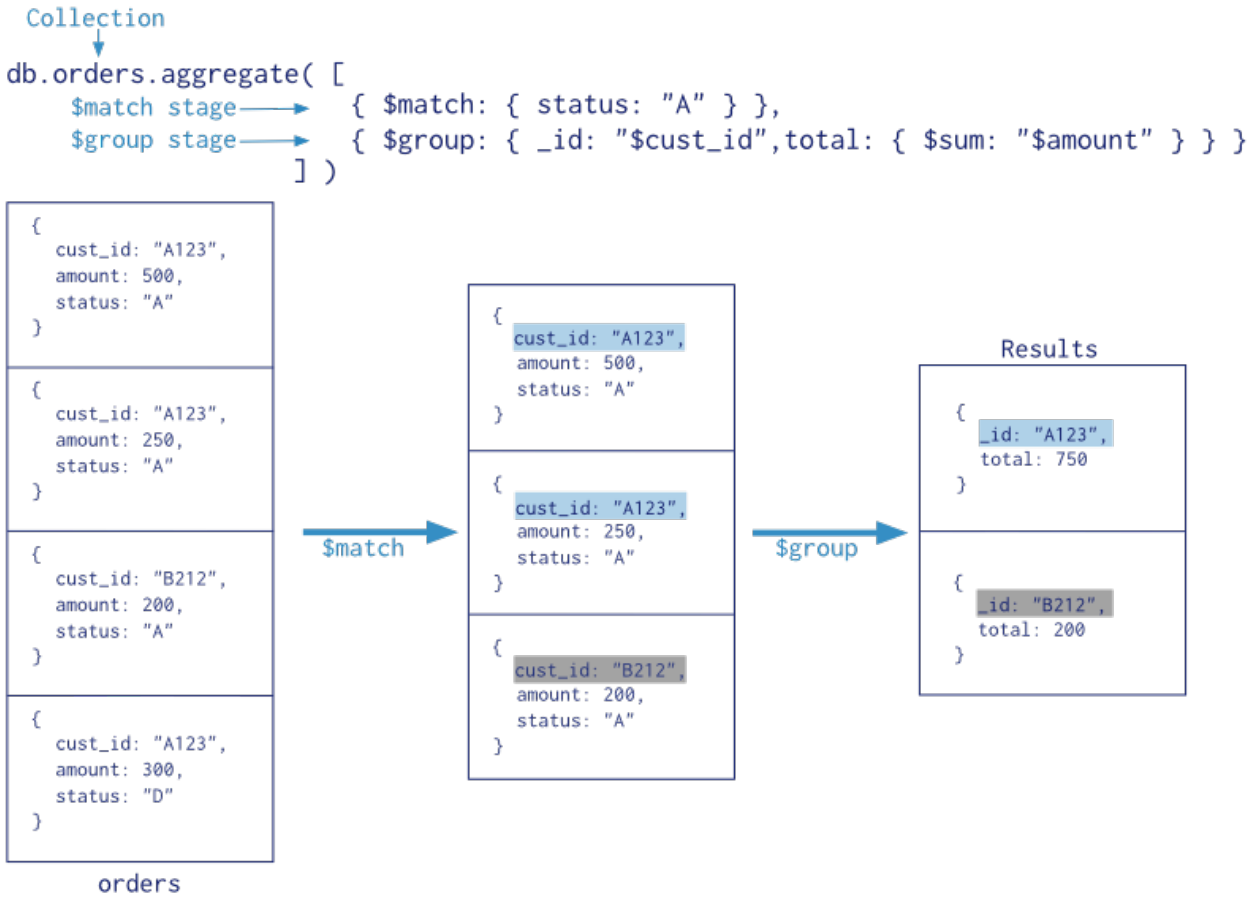
{“_id” : “3266F86FD7FD”,
“name” : “Patrick”,
“lived_in” : [12, 5, 71] }

cities

{“_id” : 12,
“name” : “New York City”
… }

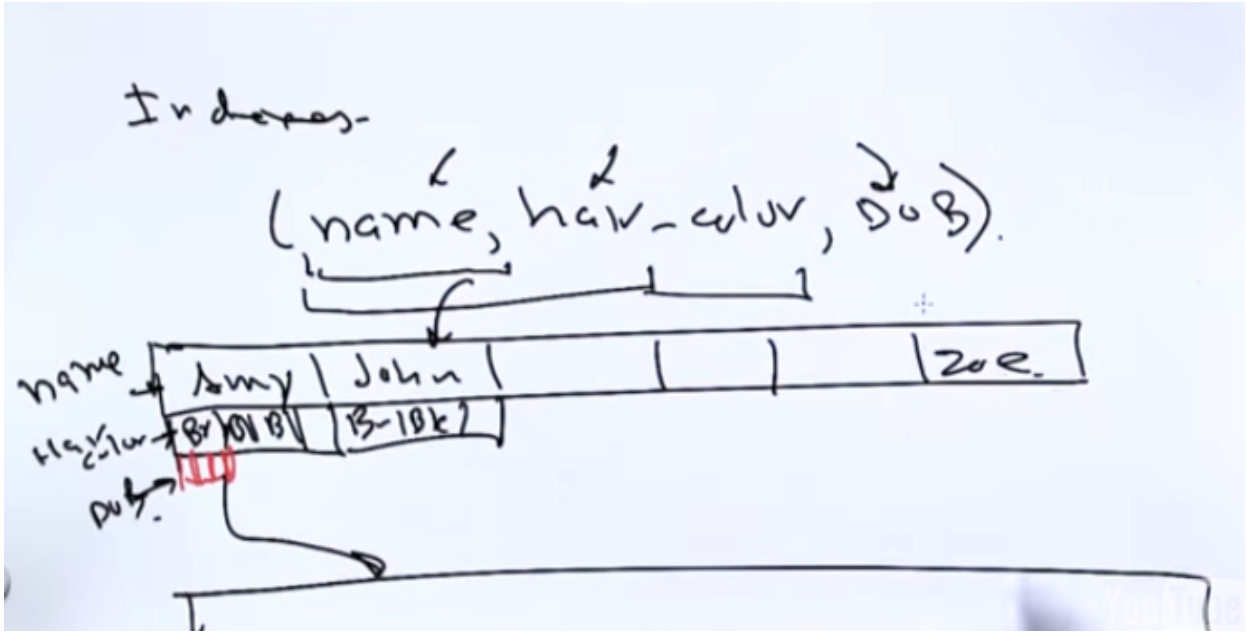
Aggregation in a few words

Aggregations process data records and return computed results



Performance - Indexes

Indexes goal : Reduce database scanning




```
db.collection.ensureIndex({user_name:1})
db.collection.getIndexes()
```

Performance - Explain function

db.collection.**find**(*conditions*).explain()

```
{
 "cursor" : "<Cursor Type and Index>",
 "isMultiKey" : <boolean>,
 "n" : <num>,
 "nscannedObjects" : <num>,
 "nscanned" : <num>,
 "nscannedObjectsAllPlans" : <num>,
 "nscannedAllPlans" : <num>,
 "scanAndOrder" : <boolean>,
 "indexOnly" : <boolean>,
 "nYields" : <num>,
 "nChunkSkips" : <num>,
 "millis" : <num>,
 "indexBounds" : { <index bounds> },
}
```

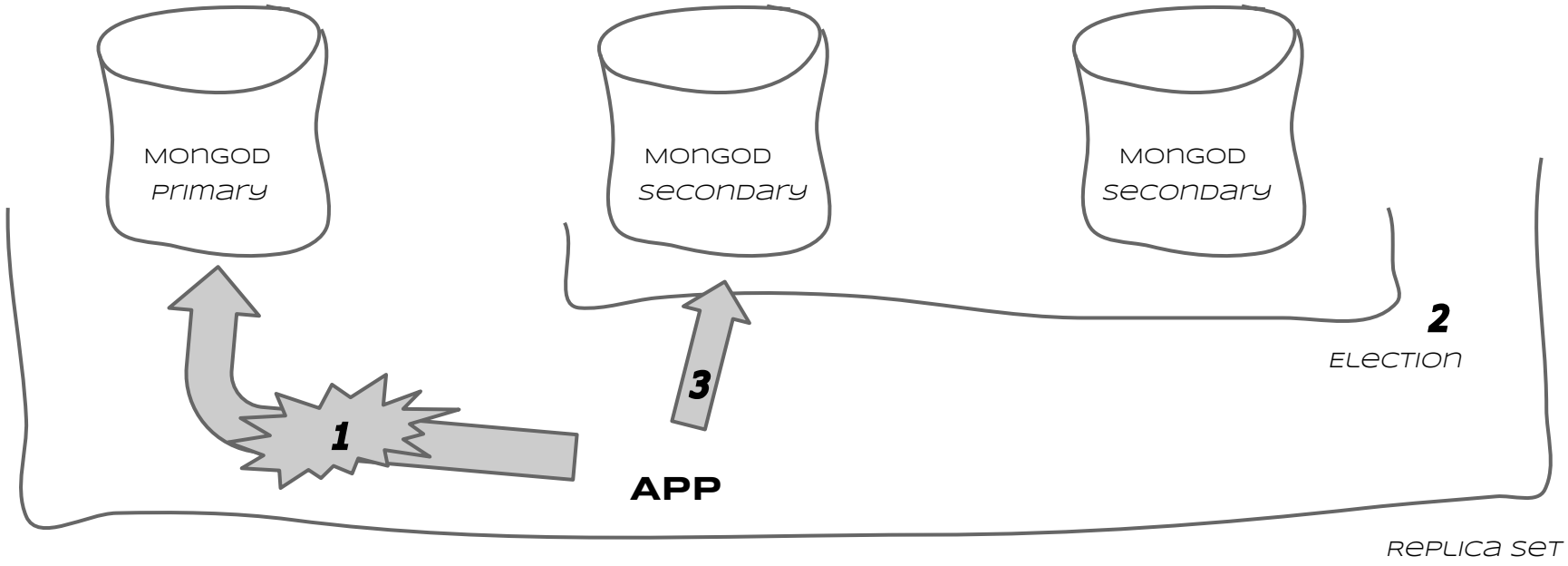
Performance - MongoDB Profiler

- Start your mongod process with profile command
- Allow you to save all the queries, or queries that takes longer than X ms, where X is a parameter
- Insert queries information in a collection called system.profile

db.system.profile.**find**()

Application Engineering - Replication

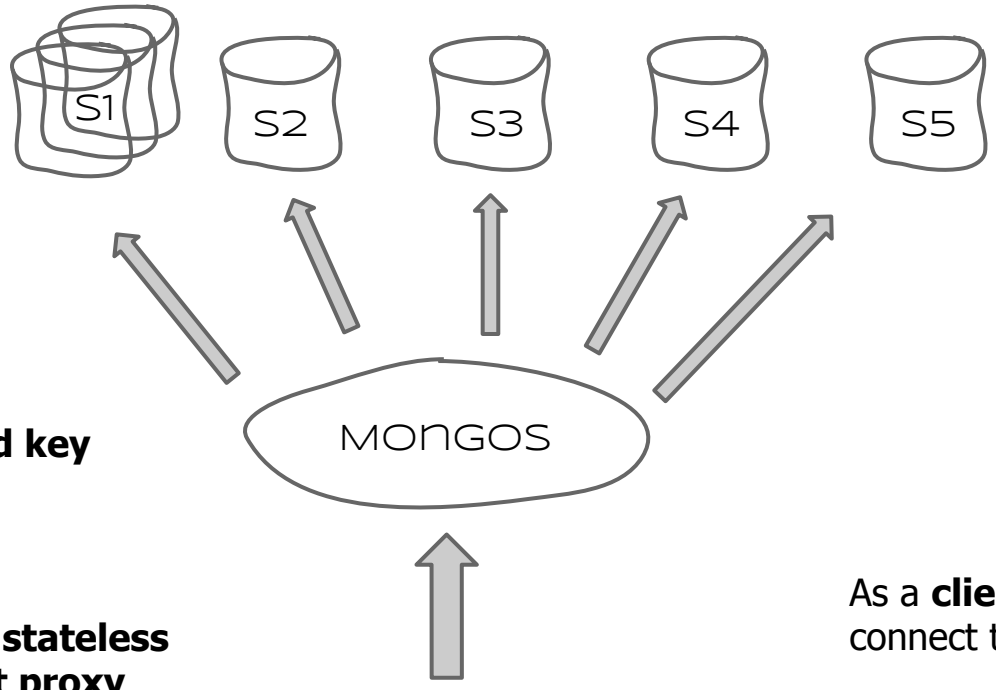
Replication needed for Availability and Fault tolerance



Three nodes are needed to assure the election of a new Primary

Application Engineering - Sharding

- Sharding is the way you can scale out
- Data is distributed on multiple servers



The splitting is based on a **shard key**

Collection is splitted in **chunks** and each shard handle multiples chunks

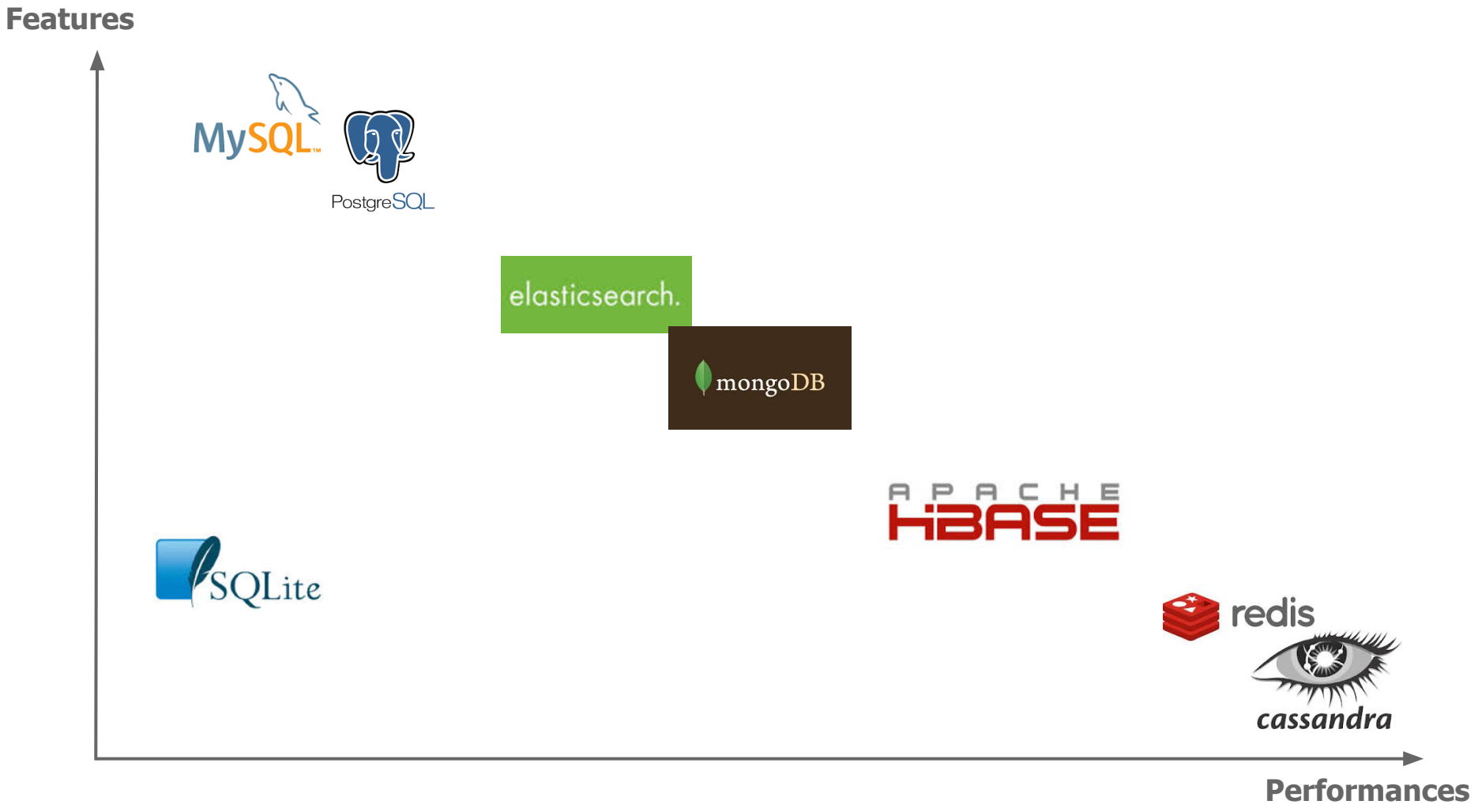
mongos is a **stateless lightweight proxy**

As a **client** you directly connect to mongos

Application Engineering - Sharding

- How to choose a shard key ?
  - Sufficient cardinality for a good distribution
  - Avoid hotspotting (ie, a shardkey that always increase)
- Who set the chunk ranges ?
  - By default, mongos create chunks based on shardey type (ie, integer from 1 to 2^{32})
  - Chunks can be defined manually
- And if the mongos is down ?
  - Actually the mongos has to be replicated too

NoSQL comparison



Online test

goo.gl/Xs2zMR

Start working!