

MongoDB : NoSql to Say Yes To

Joseph Faisal Nusairat
Groovy Sage



WHAT IS NoSQL?



relational database systems

- ◆ Relational Database systems are what developers are most use to using in your daily development.
- ◆ These are the types of databases most studied in universities, using relational algebra and calculus.
- ◆ Oracle, SQL-Server, My-SQL



RDBMS use today

- ◆ RDBMS became the defacto standard that they are used for everything,to solve problems they were not designed to do
- ◆ Every table is just one table that join together to create larger results
- ◆ Designed for large scale interaction of tables, columns, etc



performance issues

- ◆ poor performance on data intense applications when indexing large amount of documents, high traffic web sites, and streaming media
- ◆ transactions in normal rdbms systems are large and complex



what is NoSql?

- ◆ Does not expose a SQL interface
- ◆ Join operations limited or non-existent
- ◆ Scale horizontally easier than a RDBMS
- ◆ Added features in some like geo coded queries



potential issues

- ◆ transactions are fast because there aren't any
- ◆ not necessarily guaranteed write of data to the file system (if one exists)
- ◆ there are no tables, and no easy joins



multiple types of databases

- ◆ Document store
- ◆ Graph
- ◆ Key/value store on Disk / RAM
- ◆ Eventually-consistent key-value store
- ◆ Ordered key-value store



MONGO



MongoDB

- ◆ Document - oriented database written in C++
- ◆ Data is stored as BSON which is easy to convert in the application tier
- ◆ Data is arranged in collections not tables. Collections can be seen as very table like; however, collections can contain other collections.



document example

An Entire Collection

```
{  
  "username" : "bob",  
  "address" : {  
    "street" : "123 Main Street",  
    "city" : "Springfield",  
    "state" : "NY"  
  }  
}
```

A Sub Collection

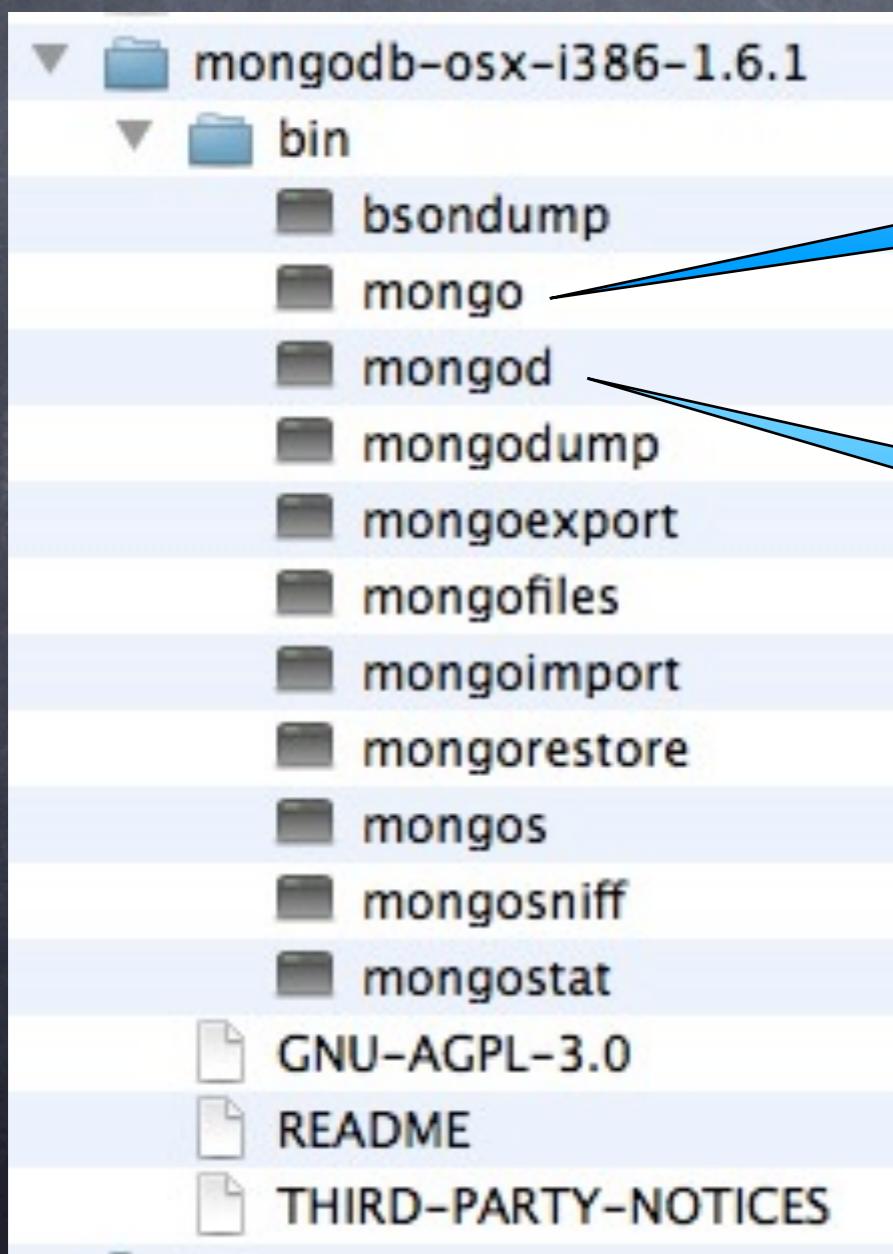


INSTALL & RUNNING



installing

- ◆ MongoDB can be downloaded from mongodb.org



Run Mongo Shell

Execute to start Mongo



mongo shell

- Once in the shell you don't have to explicitly create different projects. Just enter them and they are created automatically.

The screenshot shows two terminal windows side-by-side. The left window is titled 'mongo' and contains the MongoDB shell. It shows several log entries from the 'interruptThread' and a command history at the bottom:

```
Wed Nov 10 04:09:41 [interruptThread]
.
General CRUD Operations
APIS
what APIs are available for
Saving Files
7:bin> ./mongo
MongoDB shell version: 1.6.1
connecting to: test
> use phxjug-test
switched to db phxjug-test
> show collections
Best Practices
```

The right window is titled 'bash' and shows the shutdown process of the MongoDB instance:

```
going to close listening socket: 5
going to close listening socket: 6
going to close listening socket: 7
shutdown: going to flush oplog...
shutdown: going to close sockets...
shutdown: waiting for fs preallocator...
shutdown: closing all files...
shutdown: removing fs lock...
dbexit: really exiting now
```



CRUD



inserting

- ◆ Insert into the database using a JSON version

```
{  
    "firstName" : "joseph",  
    "lastName" : "nusairat",  
    "address" : {  
        "street" : "123 Main Street",  
        "state" : "NY"  
    }  
}
```



inserting call

- ◆ If you want to insert that item into a collection called “people”

```
db.people.insert({  
    "firstName" : "joseph",  
    "lastName" : "nusairat",  
    "address" : {  
        "street" : "123 Main Street",  
        "state" : "NY"  
    }  
})
```



result in db

- ◆ The result in the database is the BSON object as well as an `_id`

```
{  
  "_id" :  
    ObjectId("4cda8571b5da950b52727746")  
  "firstName" : "joseph",  
  "lastName" : "nusairat",  
  "address" : {  
    "street" : "123 Main Street",  
    "state" : "NY"  
  }  
}
```



updating

- ◆ When updating a record, unlike in RDMBS, the entire record will be updated thus you will need pass in the entire record you want updated.
- ◆ The Following will update our previous item but will reduce the entire record to firstName

```
db.people.update(  
  {_id : ObjectId("4cda8571b5da950b52727746")},  
  { firstName : 'joseph' } )
```



inserting new column

- ◆ However, we can update only a specific column(s) with the set operator

```
db.people.update(  
{}, { $set : {lastName : 'Smith'} } )
```

- ◆ By default that will only update one column though.
- ◆ In addition we can also use the set operator to do an insert of a new column as well.



Updating a specific item

```
db.people.update(  
{}, { $set : {middleName : 'Javier'}},  
true, true )
```

If all documents matched in the criteria are to get updated

Upsert : if the record does not exist we should insert it



find

- ◆ One can find a record by searching for specific record. This will find all matches.

```
db.people.find( { firstName : 'joe' } )
```

Defines the syntax to search for on removals.

Syntax of {} indicates all



find one

- ◆ Like find but will return only the first record it matches to.

```
db.people.findOne( { firstName : 'joe' } )
```

Defines the syntax to search for on removals.

Syntax of {} indicates all



removing

- ◆ Removing a record works with the same syntax as finding.

```
db.people.remove( { firstName : 'joe' } )
```

Defines the syntax to search for on removals.

Syntax of {} indicates all



MONGO APIS



APIs

- ◆ Mongo supports APIs to interact with many programming languages.
- ◆ C / C# / C++
- ◆ Haskell
- ◆ Java / JavaScript
- ◆ Perl / PHP / Python
- ◆ Ruby



Framework Support

- ◆ In addition to language support, a few frameworks have also created wrappers for mongo (also GMongo)
- ◆ Grails - the core team is creating tons of wrappers for NoSql support lately allowing most of GORM operations
- ◆ Rails - Mongoid and MongoMapper



CUSTOM FEATURES



SAVING FILES



GridFS

- ◆ GridFS is the specification for storing files in Mongo that are over 4MB
- ◆ This allows for safely storing large files by dividing them up among multiple documents
- ◆ Stores the files in different buckets that are in different folders. The default bucket is called “fs”



MAP REDUCE



map - reduce

- ◆ Map / Reduce is useful for batch manipulation and aggregate querying of data
- ◆ Map Reduce can be used in a situation where you would normally have used “Group By” in SQL



map - reduce

- ◆ Map / Reduce works by creating a temporary collection to hold the output data.
- ◆ We can then aggregate and reduce them afterwards using pure JavaScript



map portion

- ◆ mapping is designed to take a large input and then divide them up into smaller pieces.
- ◆ The map will specifically identify for us the item that is going to be aggregated on.



reduce portion

- ◆ The reduce will basically aggregate what the map did. By taking those smaller pieces and bringing them back into one final answer.
- ◆ The reduce uses the key passed in by the map to do some kind of value aggregation with it.



GEO LOCATION



geospatial indexing

- ◆ Mongo allows one store geo locations in the database.
- ◆ These are the 2 dimensional latitude / longitude coordinates of any particular place on the earth.
- ◆ Mongo provides the ability to search on these coordinates.



geospatial indexing

- ◆ Searching normally occurs on a flattened earth although as of 1.7.0 there is a spherical model.
- ◆ This allows the ability to store a location and then find items within a certain distance
- ◆ Distance is stored in radians



ADVANCE TOPICS



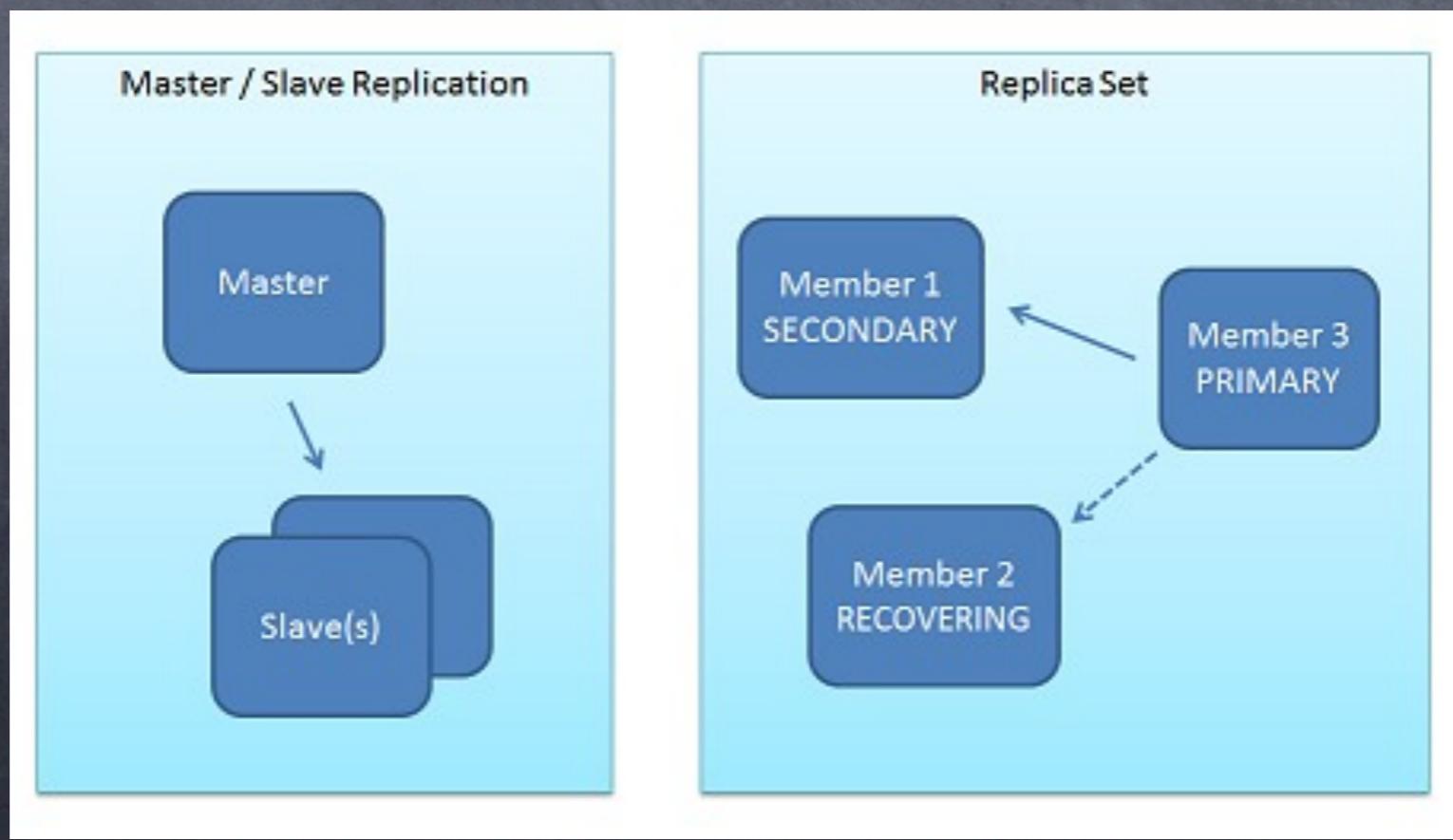
how insertion works

- ◆ With MongoDB there are a few major design considerations to consider when switching from RDBMS.
- ◆ JOINS - all have to be done through Map - Reduce
- ◆ COLLECTIONS - these are like tables but different in their dynamic nature and sub-collection ability



replication

- ◆ Mongo also supports asynchronous replication of data between servers for failover and redundancy.
- ◆ Only one master is active at a time.

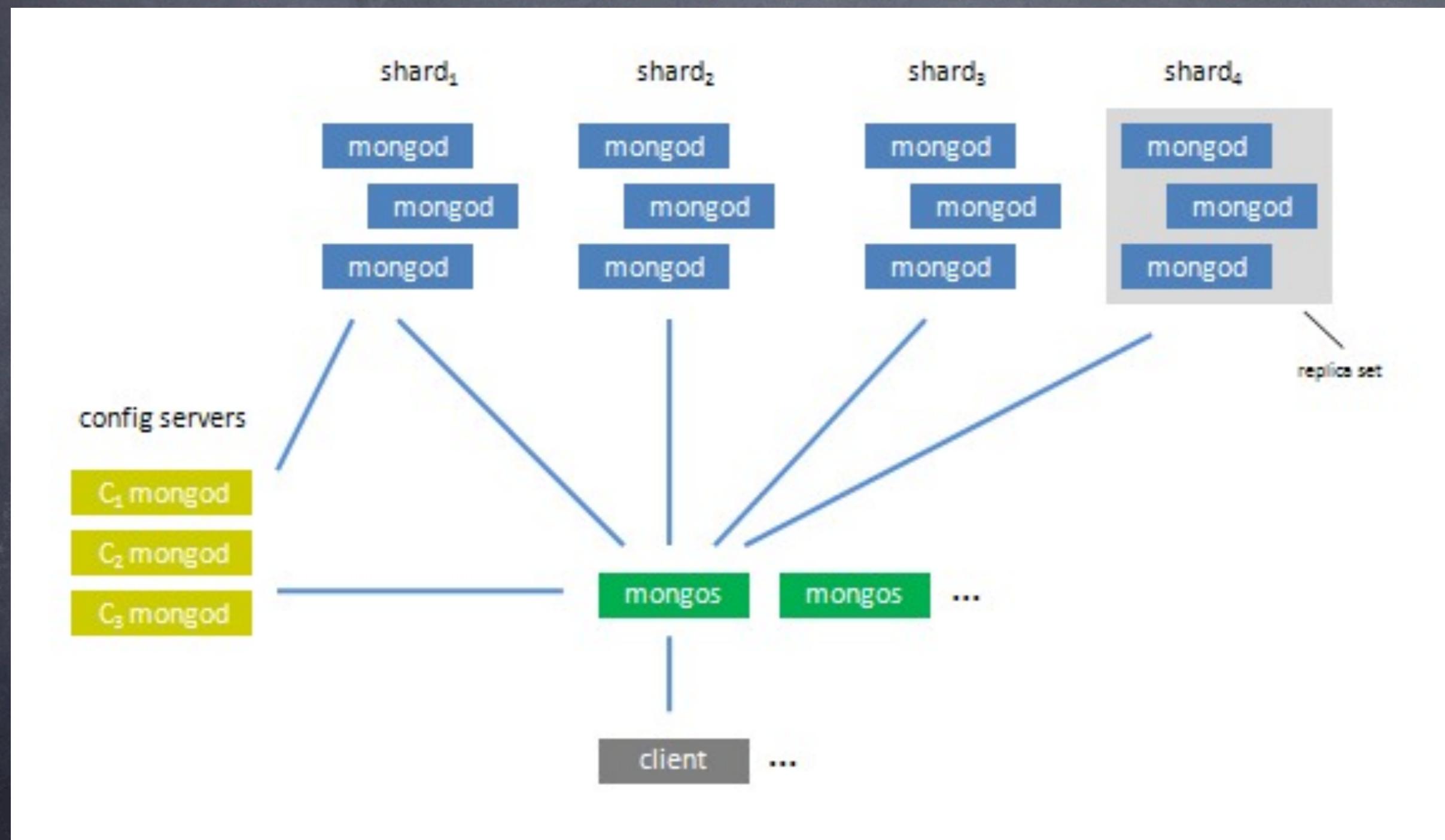


sharding

- ◆ Sharding is enabling horizontal scrolling over multiple nodes.
- ◆ Sharding allows us to designate different nodes for different data on a per collection basis.
- ◆ One identifies what properties they want to shard on, much like partitioning in Oracle works.



sharding



BEST PRACTICES



Design Considerations

- ◆ With MongoDB there are a few major design considerations to consider when switching from RDBMS.
- ◆ JOINS - all have to be done through Map - Reduce
- ◆ COLLECTIONS - these are like tables but different in their dynamic nature and sub-collection ability



Q & A

