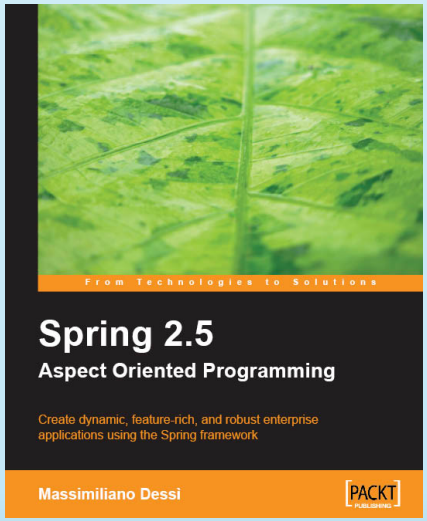
webtech CONFERENCE 2010

mongoDB in 25 min

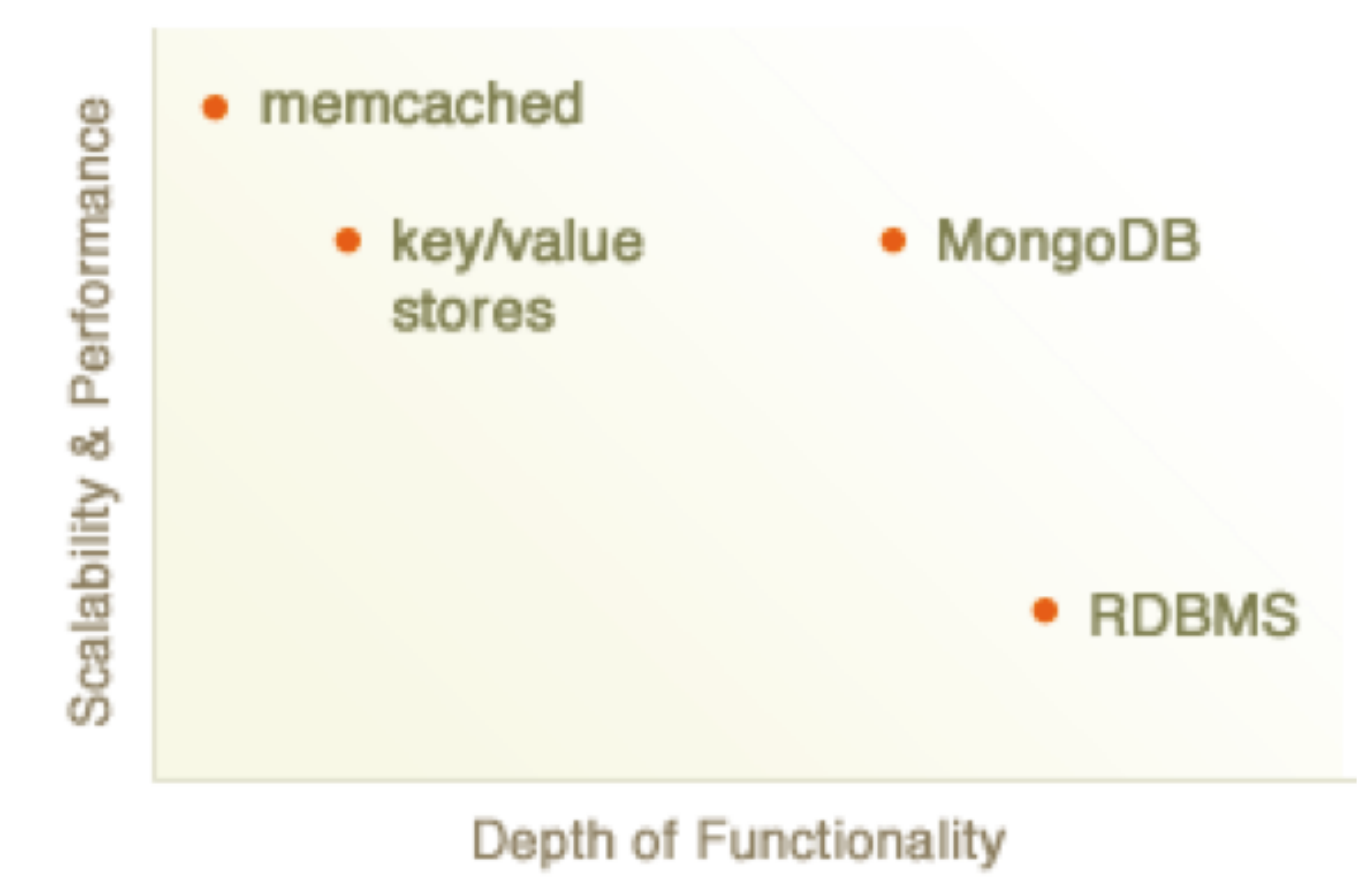
Massimiliano Dessì

Speaker

Software architect/developer	Pronetics/Sourcesense
Founder	Spring Italian User Group
Chairman	JugSardegna Onlus
Committer/Contributor	OpenNMS - MongoDB
Author	Spring 2.5 Aspect Oriented Programming



Target


memcached
key/value stores
MongoDB
RDBMS
Scalability & Performance
Depth of Functionality

Deployments

http://www.mongodb.org/display/DOCS/Production+Deployments

Features

- Document-oriented
  - Documents (objects) map nicely to programming language data types
  - Embedded documents and arrays reduce need for joins
  - Dynamically-typed (schemaless) for easy schema evolution
  - No joins and no transactions for high performance and easy scalability
- High performance
  - No joins and no transactions makes reads and writes fast
  - Indexes with indexing into embedded documents and arrays
  - Optional asynchronous writes
- High availability
  - Replicated servers with automatic master failover
- Easy scalability
  - "slaveOK" reads are distributed over replicated servers
  - Automatic sharding (auto-partitioning of data across servers)
  - Reads and writes are distributed over shards
  - No joins and no transactions make distributed queries easy and fast
- Rich query language

{"greeting" : ”Hello world !”}

JSONize your data

http://www.mongodb.org

```
{ "name" : "MongoDB",
   "info" : {  "storage" : "Binary JSON (BSON)",
               "full index" : "true",
               "scale" : "Autosharding",
               "query" : "Rich document-base queries",
               "replication" : "Replica sets",
               "atomic modifiers" : "Fast in place update",
               "binary content" : "GridFS",
               "batch operation" : "Map/Reduce”
               ”js server side" : ”true”
            },
    "_id” : "024x6f279578a64bb0686743"
 }
```

No Join

Data container

Document instead of Row

Collections and subcollections

instead of Tables

Document limit

Larger than 4MB

The entire text of War and Peace is

3.14 MB...

Driver

C, C#, C++, Clojure, D, Delphi, Erlang,

Factor, Fantom, F#, Go, Groovy,

Haskell, Java, Javascript, Lua, Nodejs,

ObjectiveC, Perl, PHP, Python, R,

Ruby, Scala, Scheme (PLT), Smalltalk

http://www.mongodb.org/display/DOCS/Drivers

No constraints

Collections are schema free

Documents within a single collection

can have any number of different "shapes”

No SQL Injection

Mongo is invulnerable to injection attacks, no code execution

Insert

```
db.mycollection.insert({"name" : "foo"})
```

Delete

```
db.mycollection.delete({"name" : "foo"})
```

Update

The schema can be changed (schema free)

```
var item = db.mycollection.findOne({"name" : "foo"});
item.albums = {"2009" : "greatest hits”};
item.members = {”number" : ”5"};
db.mycollection.update({”name” : ”foo"}, item);
```

Upsert

Update or insert if not present

```
db.mycollection.update({”name” : ”sam"}, true);
```

Fastest

Fire and Forget

Command with response

getLastError

Mongo Shell

```
Last login: Sun Nov  7 20:34:05 on ttys002
localhost:~ max$ mongo
MongoDB shell version: 1.6.3
connecting to: test
> help command
        db.help()                    help on db methods
        db.mycoll.help()             help on collection methods
        rs.help()                    help on replica set methods
        help connect                 connecting to a db help
        help admin                   administrative help
        help misc                    misc things to know

        show dbs                     show database names
        show collections             show collections in current database
        show users                   show users in current database
        show profile                 show most recent system.profile entries with time >= 1ms
        use <db_name>                set current database
        db.foo.find()                list objects in collection foo
        db.foo.find( { a : 1 } )     list objects in foo where a == 1
        it                           result of the last line evaluated; use to further iterate
        exit                         quit the mongo shell
> _
```

Modifiers

Partial updates

```
$set   (set a key, or add if not present)
$inc   (with numbers)
$push  (add to the end of an array)
$ne
$addToSet
$each
$pop  (remove from the end)
$pull  (remove element that match criteria)
```

Query

Return all keys in the document

```
db.mycollection.find({"name" : "foo"})
```

Keys to return (second param)

```
db.mycollection.find({}, {”name" : 1})
```

Query criteria

Conditionals

$lt $lte $gt $gte

```
start = new Date("01/01/2010")
db.mycollection.find({"registered" : {"$lt" :
start}})

db.mycollection.find({"name" : {"$ne" : "foo"}})
```

Query criteria

Conditionals

$in $or, $nin, $not

<, <=, >, >=, $ne, $mod, $all, $size, $exists

```
db.mycollection.find({”external_num" : {"$nin" :
[1356, 525, 874]}})
db.mycollection.find({"$or" : [{"ticket_no" :
525}, {”name" : foo}]})
```

Query REGEX

Perl Compatible Regular Expression

```
db.mycollection.find({"name" : /foo?/i})
```

Query on Array

Inside a document

`$all $size $slice`

Grouping

`count distinct group finalize $key`

Where Query

JavaScript as part of your query

```
db.mycollection.find({"$where" : function () {
   for (var current in this) {
   }
…}
```

Indexing

```
db.mycollection.ensureIndex({"name" : 1,
"info" : 1 }
```

Geospatial indexing

```
db.mycollection.ensureIndex({"gps" : "2d"})
```

```
db.star.trek.ensureIndex(
{"light-years" : "2d"}, {"min”:-10, "max”:10})
```

Map Reduce

```
map = function() {
    for (var key in this) {
      emit(key, {count : 1});
    }
};


reduce = function(key, emits) {
     total = 0;
     for (var i in emits) {
          total += emits[i].count;
     }
     return {"count" : total};
}
```

Replication

Replica set, clustering with automatic failover

Master is elected by the cluster and may change to another node if the current master goes down

Sharding

Splitting data and storing different portions of the data on different machines, also know as partitioning

Option: auto/manual

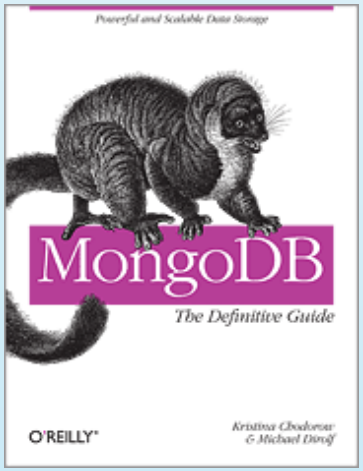
GridFS

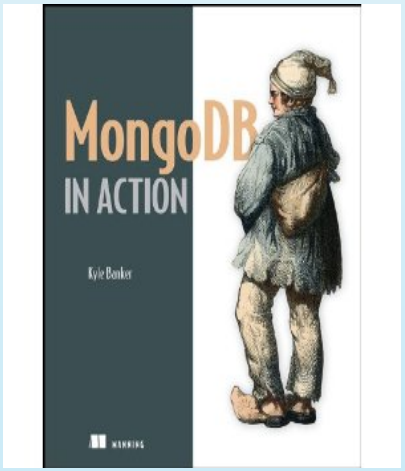
GridFS is a specification for storing large files such video, photos, blob in MongoDB

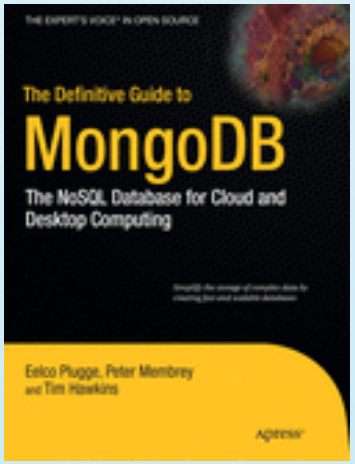
GridFS uses two collections to store data:

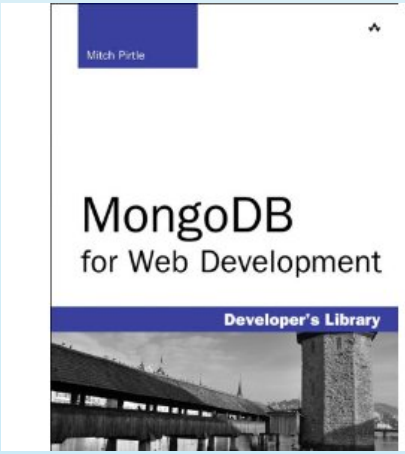
- files contains the object metadata
- chunks contains the binary chunks with some additional accounting information

Books









http://www.mongodb.org/display/DOCS/Books

Thanks !

Massimiliano Dessì

desmax74 at yahoo.it

http://twitter.com/desmax74

http://jroller.com/desmax

http://www.linkedin.com/in/desmax74

http://wiki.java.net/bin/view/People/MassimilianoDessi

http://www.jugsardegna.org/vqwiki/jsp/Wiki?MassimilianoDessi