

MongoDB 101

Dan Patsey

Capital District Java Developers Network

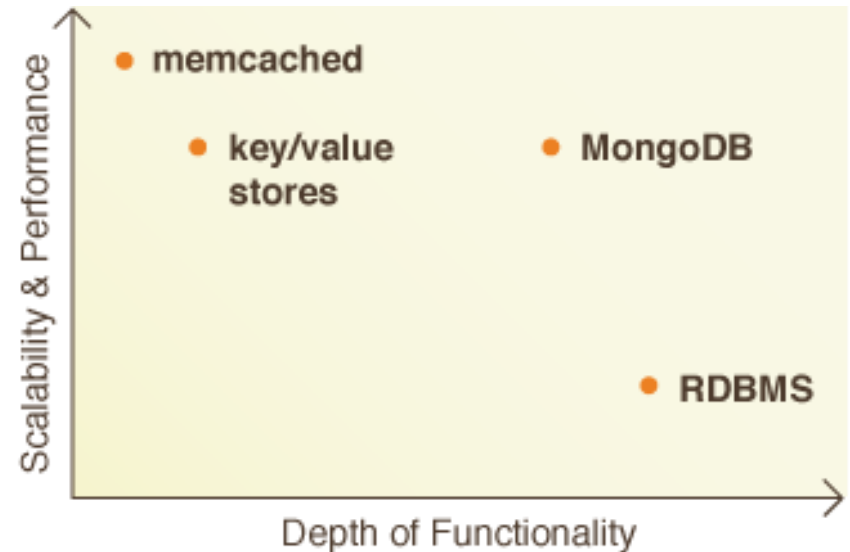


Agenda

- * MongoDB Basics
- * Querying
- * Administration

What is MongoDB?

- * Scalable
- * Schemaless
- * Document-oriented



How is MongoDB different?

- * No Joins
- * Transactions
 - * Atomic single documents

Data Model

- * Database
- * Collection
- * Document
 - * Ordered Set of Key/Value Pairs

Data Types

JSON

- * Number
- * Object
- * **String**
- * **Boolean**
- * **Array**
- * **null**

BSON

- * **32/64-bit Integer, 64-bit Float**
- * **Embedded Document**
- * **Date**
- * **Binary Data**
- * **Regular Expression**
- * **JavaScript Code**
- * **ObjectId**

Object ID

```
{  
  "_id": ObjectId("4d7f81a98b9c33ec579385d8"),  
  "name": "Jane Smith"  
}
```

- * Every document must have an `_id` key
- * 12-byte binary value

0	1	2	3	4	5	6	7	8	9	10	11
Time				machine			pid		inc		

Indexes

- * `_id` is always indexed
 - * Embedded Documents
 - * Arrays
 - * Compound
 - * Unique
 - * Sparse
-
- * Explain/Hint

DBRefs

Posts

```
{
  "_id": ObjectId("4b866..."),
  "author": "dpatsey",
  "media": [
    {$ref: "Pictures", $id: ObjectId("5c3462...")},
    {$ref: "Images", $id: ObjectId("4j8dk3...")}
  ]
}
```

- * Don't know type of reference
- * Client-side dereferencing

Querying

Querying (Insert, Remove)

```
db.myColl.insert({"name" : "Bob Smith"})
```

```
db.myColl.insert({  
  "_id" : "bsmith"  
  "name" : "Bob Smith"  
})
```

* Supports Batch

```
db.myColl.remove()
```

```
db.myColl.remove({"name" : "Bob Smith"})
```

Querying (Update)

```
db.myColl.update({"name" : "Bob Smith"}, bob)
```

* 2 params (Query Document, Modifier Document)

```
db.myColl.update({"name" : "Bob Smith"},  
  {"$set" : {"age" : 21, "weight": "431 lbs"}})
```

```
db.myColl.update({"name" : "Bob Smith"},  
  {"$inc" : {"age" : 1}})
```

```
db.myColl.update({"name" : "Bob Smith"},  
  {"$push" : {"emails" : "bsmith@gmail.com"}})
```

Querying (Find)

```
db.myColl.find({"age" : 22})
```

```
db.myColl.find({"age" : {"$gte" : 18, "$lte" : 30}})
```

```
db.myColl.find({"age" : {"$in" : [18, 20, 22]}})
```

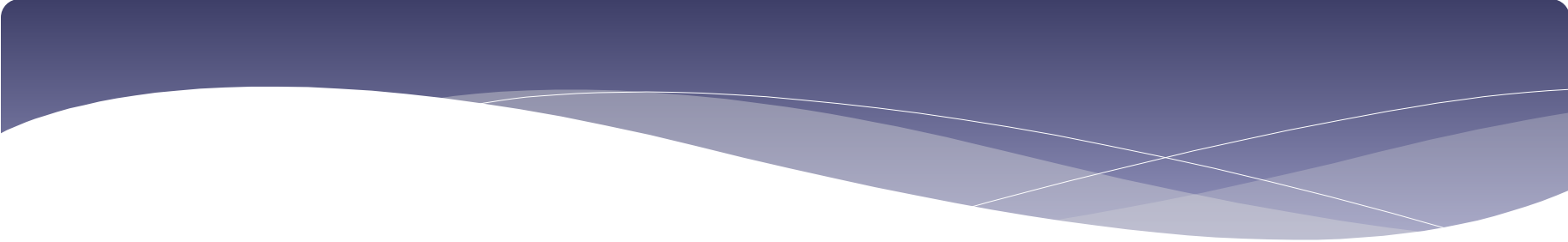
```
db.myColl.find({"name" : /bob.*i})
```

```
db.myColl.find().skip(50).limit(10)
```

* **and, or, exists, size, elemMatch, Embedded, where**

Querying Tools

- * MongoDB Shell
- * MongoVUE
- * Drivers (Java)
- * Morphia



Querying Demo

Administration

Backup / Restore

Journaling Off

- * Stop, Copy Data, Start
 - * Cold
- * **mongodump**
 - * — Not Point In Time
- * **fsync** & Write Lock, Copy Data, Unlock
 - * Blocks writes
- * Secondary / Slave Backup
 - * Ideal

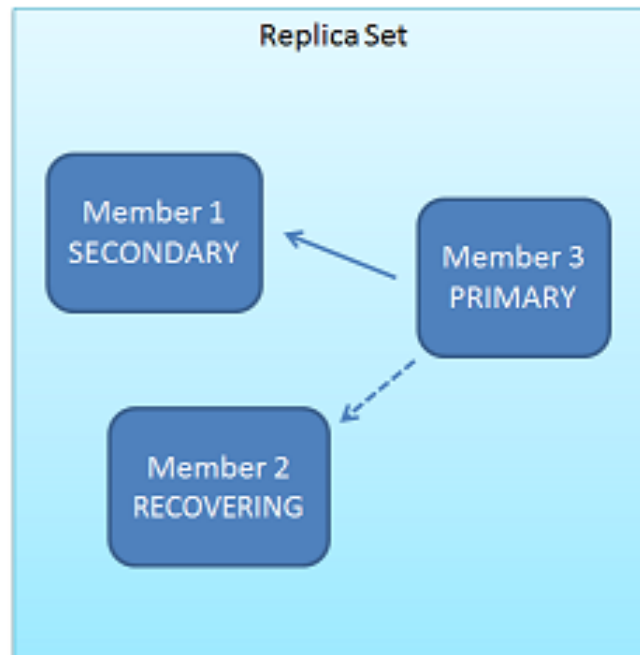
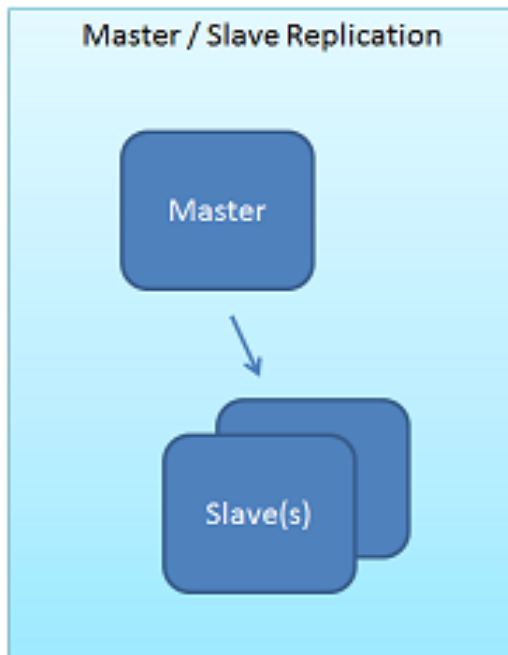
Journaling On

- * OS Snapshot
 - * Performance Overhead

Replication

- * Motivations
 - * Scale out Reads
 - * Backups / Maintenance
 - * Offline Batch Processing
 - * Failover / Disaster Recovery
- * Types
 - * Master / Slave
 - * Manual
 - * Replica Set
 - * Automatic failover and recovery

Replication





Replication Demo

Replication

printReplicationInfo

configured oplog size:	50MB
log length start to end:	1714secs (0.48hrs)
oplog first event time:	Thu Jun 10 2010 10:22:01
oplog last event time:	Thu Jun 10 2010 10:50:35
now:	Thu Jun 10 2010 10:50:45

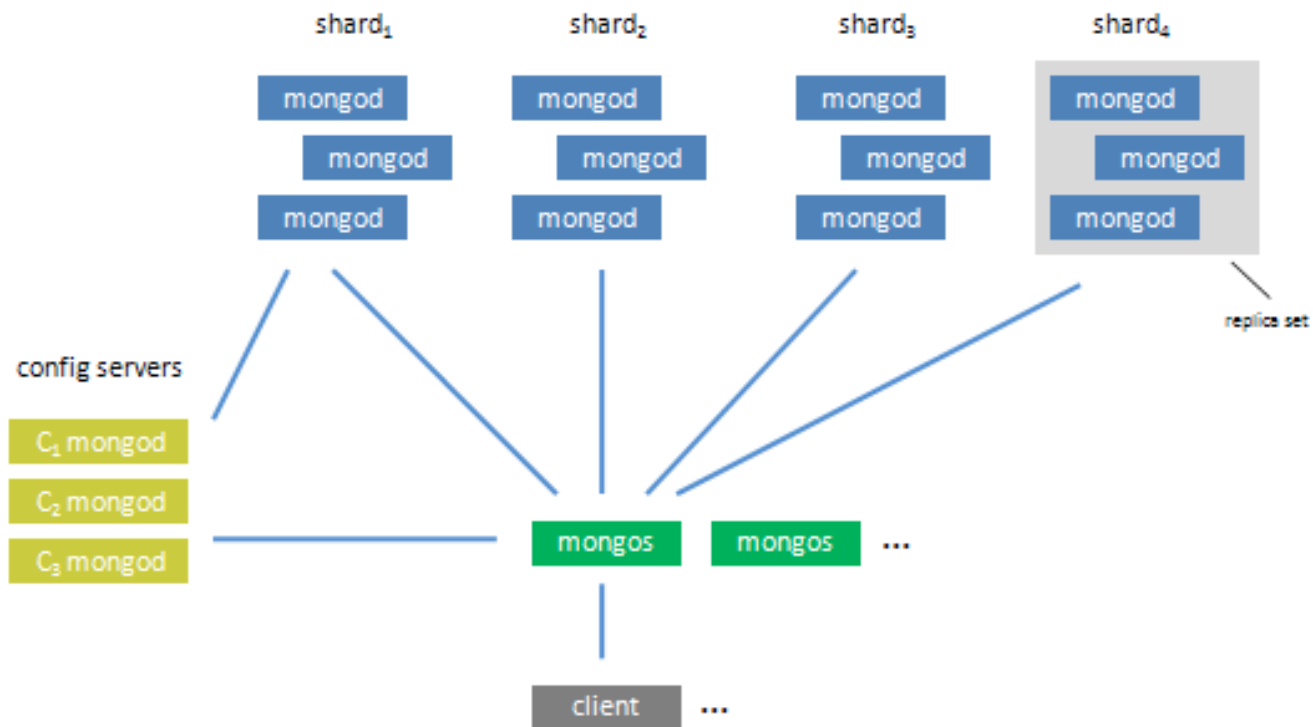
printSlaveReplicationInfo

source:	localhost:27017
syncedTo:	Thu Jun 10 2011 10:49:25 = 1secs ago (0hrs)

Sharding

- * Start with none and convert if and when you need it
- * Introduces:
 - * Shard - Holds a subset of Documents from a Collection
 - * Config Server - Stores which data is on which Shard
 - * Mongos - Routes requests & aggregates responses
- * Document location determined by Shard Key

Sharding



Sharding

printShardingStatus

shards:

```
{ "_id" : "shardo", "host" : "localhost:10001" }  
{ "_id" : "shard1", "host" : "localhost:10002" }
```

databases:

```
{ "_id" : "admin", "partitioned" : false, "primary" : "config" }  
{ "_id" : "mydb", "partitioned" : true, "primary" : "shardo" }
```

mydb.products chunks:

```
{ "num" : { $minKey : 1 } } --> { "nr" : 0 } on : shard1  
{ "num" : 0 } --> { "nr" : 99 } on : shard1  
{ "num" : 100 } --> { "nr" : 199 } on : shard1  
{ "num" : 200 } --> { "nr" : 299 } on : shard1  
{ "num" : 300 } --> { "nr" : 399 } on : shardo  
{ "num" : 400 } --> { "nr" : 499 } on : shardo  
{ "num" : 500 } --> { "nr" : { $maxKey : 1 } } on : shardo  
{ "_id" : "test", "partitioned" : false, "primary" : "shard1" }
```

Other Topics

- * Geospatial Indexing
- * MapReduce
- * JavaScript
- * GridFS
- * Capped Collections

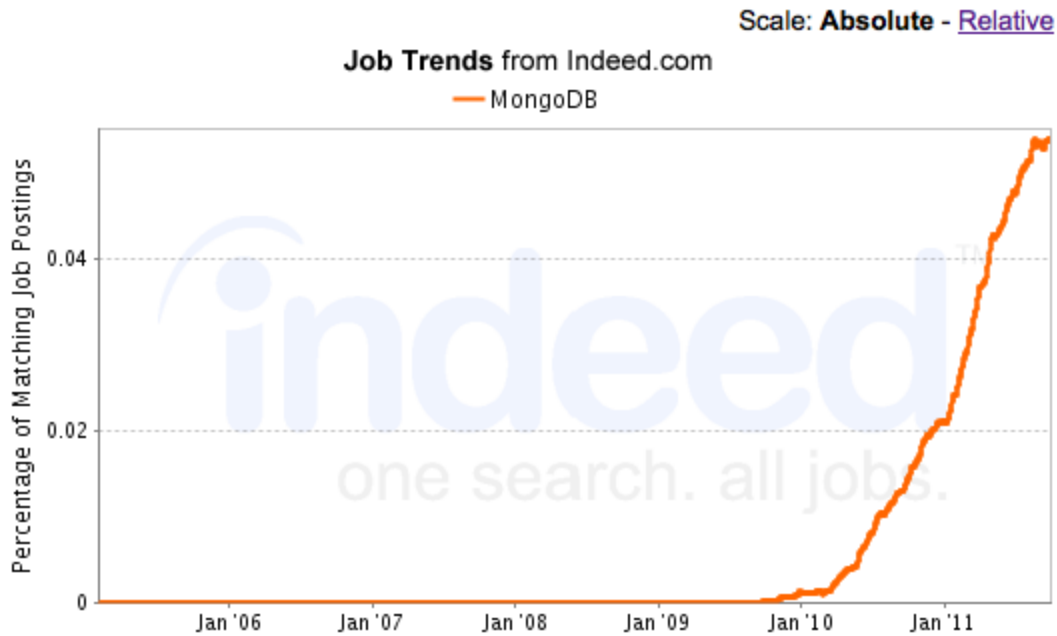
Administration Tools

- * MongoDB Web Console
- * Mongo Live
- * MongoDB Monitoring Service

Why MongoDB?

- * Web applications
- * Agile development
- * Logging
- * Unstructured Data

MongoDB Job Trends



Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

▶ [Email to a friend](#)

▶ [Post on your blog/website](#)

Top Job Trends

1. [HTML5](#)
2. **MongoDB**
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)



Questions

?