Document-Oriented DBs and MongoDB

Mathias Stearn

10gen

VolcaNoSQL EU – April 20, 2010

1. Document Oriented Databases
   - What they are
   - What they aren’t
2. CouchDB
   - Pros and Cons
3. MongoDB
   - Compared to CouchDB
   - Into to Mongo
   - Sharding

1. Document Oriented Databases
   - What they are
   - What they aren't
2. CouchDB
   - Pros and Cons
3. MongoDB
   - Compared to CouchDB
   - Into to Mongo
   - Sharding

- Only Two Document Oriented DBs Right Now
  - MongoDB and CouchDB
  - Everyone has an opinion on what makes a DODB
    - Hard to choose “defining” characteristics
    - This is my take on the space – deal with it

- Document Oriented
  - Think JSON Documents, not Word/OOo Documents
  - Can store files through Attachments and GridFS
  - Could use XML but XML sucks

```
{
  _id: "mstearn",
  name: "Mathias␣Stearn",
  karma: 42,
  active: true,
  birthdate: new Date(517896000000),
  interests: ["MongoDB", "Python", "Üñíçøđě"],
  subobjects: [{foo: "bar"},
               {foo: "baz", count: 13}]
}
```

- Hierarchical
  - Can nest objects to arbitrary depth
  - Server can reach into objects
  - Whole “Object” stored at one place on disk

```
{
 comments: [
   { by: 'mstearn', body: 'text', tags: ['empty']
       votes: {good: 100, bad: 10, net: 90} },
   { by: 'mdirolf', body: 'what?', tags: ['question']
       votes: {good: 30, bad: 40, net: -10} }
  ]
}
```

- Not Relational
  - Not forced into rows/columns/tables
  - No built-in joins
    - Less need because objects can directly store lists
  - Many-to-Many still possible (learn how at workshop)
  - No SQL (no SQL injections either)
  - No Object-Relational impedance mismatch

- Not Just Key-Value Store
  - Key and value are not separate
  - Supports queries on non-primary keys
    - Secondary Indexes
  - Supports Aggregation
    - Currently via JavaScript MapReduce
    - Both DBs looking into alternatives
  - Can be as fast as a KV store if you only need KV features
    - But still have access to a real database when needed
  - Less custom code needed

- Not the same as stuffing a JSON blob in a database
  - Database understands document format
  - Can query on any field
  - “Use the right tool for the job”

1. Document Oriented Databases
   - What they are
   - What they aren't
2. CouchDB
   - Pros and Cons
3. MongoDB
   - Compared to CouchDB
   - Into to Mongo
   - Sharding

WARNING

I am not an expert on CouchDB!

Pros

- HTTP RESTful Interface
- Stores and communicates in plain JSON
- Query using precomputed JS Map/Reduce views
- Fastest if you use Bulk Insert
- Uses Append-Only File

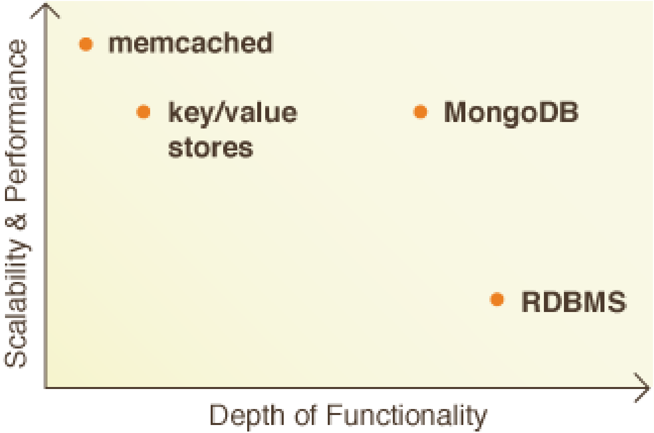
Cons

- HTTP RESTful Interface
- Stores and communicates in plain JSON
- Query using precomputed JS Map/Reduce views
- Fastest if you use Bulk Insert
- Uses Append-Only File

1. Document Oriented Databases
   - What they are
   - What they aren't
2. CouchDB
   - Pros and Cons
3. MongoDB
   - Compared to CouchDB
   - Into to Mongo
   - Sharding

MongoDB

- Custom wire protocol with many supported languages
- Stores and communicates in BSON (Binary JSON)
- Rich Ad-Hoc Query Language
  - MapReduce for aggregation
- Bulk Insert available, but regular insert is very fast
- Data is updated in place


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

- The Mongo Shell
  - `http://try.mongodb.org` ← go here now
- Full JS shell + MongoDB extensions
- Most MongoDB documentation uses shell syntax

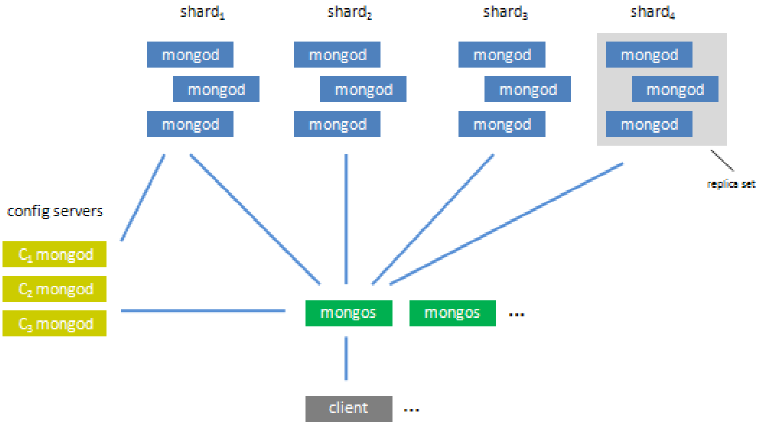
```
db.users.insert({_id:'mstearn',
                 name: {first:'Mathias',
                        last:'Stearn'}
                 company: '10gen',
                 knows: ['MongoDB', 'Python', 'C++'],
                 posts: 42})

db.users.find({_id: 'mstearn'})
db.users.find({company: '10gen'})
db.users.find({posts: {$gte: 40, $lt: 50}})
db.users.find({'name.last': 'Stearn'})
db.users.ensureIndex({knows: 1})
db.users.find({knows: 'MongoDB'})
db.users.find({knows: {$in: ['MongoDB', 'Mongo']}})
db.users.find({knows: {$all: ['MongoDB', 'Python']}})
db.users.find({knows: /^Mongo/})
db.users.find().sort({posts: -1}).skip(10).limit(10)
```

Shameless self-promotion

`http://github.com/RedBeard0531/MongoMagic/`

```
db.users.find( M._id == 'mstearn' )
db.users.find( M.company == '10gen' )
db.users.find( 40 <= M.posts < 50 )
db.users.find( M.name.last == 'Stearn' )
db.users.find( M.knows.IN('MongoDB', 'Mongo') )
db.users.find( M.knows.ALL('MongoDB', 'Python') )
db.users.find( M.knows.STARTSWITH('Mongo') )
db.zips.insert({_id: '10011', loc: [43, -74]})

db.zips.ensureIndex({loc: '2d'})

db.zips.find({loc: {$near: [43, -74]}})

var box = [[x1, y1], [x2, y2]]
db.zips.find({loc: {$within: {$box: box}}})

var circle = [[x,y], radius]
db.zips.find({loc: {$within: {$center: circle}}})
db.posts.insert({_id: ObjectId(123),
                 by: 'mstearn',
                 title: 'Why␣MongoDB␣is␣Awesome',
                 body: 'It␣just␣is␣MASSIVE␣TYPO',
                 tags: [] })

db.posts.update({_id: ObjectId(123)},
                {$set: {body: 'It␣just␣is' }})

db.posts.update({_id: ObjectId(123)},
                {$push: {tags: 'Citation␣Needed'}})

db.tags.update({_id: 'Citation␣Needed'},
               {$inc: {count: 1}},
               {upsert: true})
```

- You (probably) don’t need sharding!
- My desktop can handle 100k inserts/s or 160k queries/s
  - For comparison, linkedin gets 55M views/day [quantcast]
  - $55000000/8/60/60 = 1736$ views/sec
- Largest Mongo install is 12TB on single server
  - Mostly large objects
- Wordnik.com has 1.5TB in over 5 Billion docs
  - Sustained 100,000 inserts per second during loading
  - Queries are 4x faster than old MySQL setup
- Speed and Scalability are different things
  - But you only need scalability if you’re too slow


shard$_1$
shard$_2$
shard$_3$
shard$_4$
mongod
mongod
mongod
mongod
mongod
mongod
mongod
mongod
mongod
mongod
mongod
mongod
replica set
config servers
C$_1$ mongod
C$_2$ mongod
C$_3$ mongod
mongos
mongos
...
client
...

Questions?

Links

- http://media.mongodb.org/zips.json (for workshop)
- http://try.mongodb.org (Try mongo in your browser)
- http://www.mongodb.org
- #mongodb on irc.freenode.net
- mongodb-user on google groups

Contact

- mathias@10gen.com
- @mathias_mongo ← follow me!