MongoDB

Or how I learned to stop worrying and love the database

Mathias Stearn

10gen

Bay Area Hadoop Meetup – February 17, 2010

> *The resulting [MongoDB] application has literally changed the way the pharma company conducts business. Whereas in the past, patient queries could take minutes to hours, results are now essentially real-time.*

`http://news.cnet.com/8301-13846_3-10451248-62.html`

> *Why bother [with] memcached for caching HTTP sessions when you have an authoritative MongoDB? The performance is there.*

Carl Byström, @cgbystrom on twitter

> *Compared to hadoop, Mongo's speed and startup time make developing new queries much easier; what took us two weeks to get working on hadoop was done in two days on mongo.*

Emmett Shear, CTO (and developer) at Justin.tv

> *It took me half a day to go from not touching MongoDB to writing some fairly good functionality against it. It makes setting up, configuring, and interfacing with MySQL look archaic – ridiculously archaic.*

`http://geekaustin.org/2010/01/31/mongodb-day-geek-austin-data-series`

- Document Oriented
- Organized into Databases and Collections (like Tables)
- JSON-like (BSON)
- Schemaless
- Dynamic, Strong Typing
- Database can “reach into” objects

```
db.people.insert({
  _id: "mstearn",
  name: "Mathias Stearn",
  karma: 42,
  active: true,
  birthdate: new Date(517896000000),
  interests: ["MongoDB", "Python", "Ünîçødĕ"],
  subobject: {foo: "bar"}
});
```

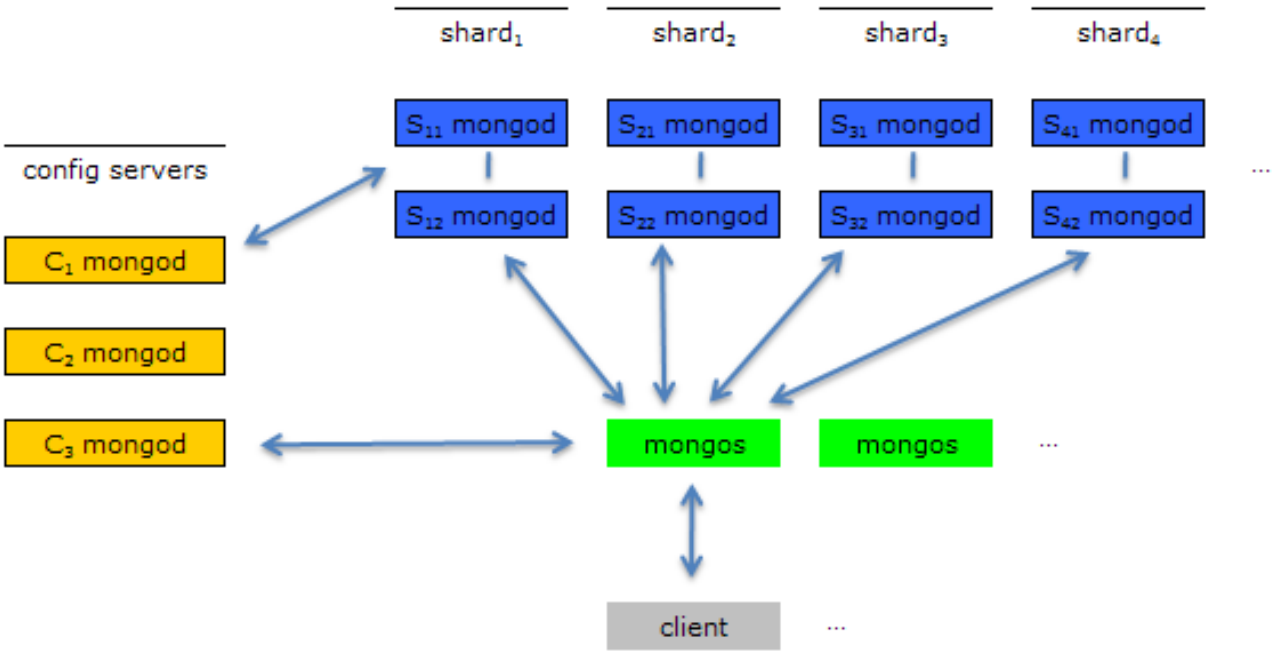
JavaScript used for:

- Shell and Documentation
- (Very) Advanced Queries
- “Group By” Queries
- MapReduce

```
db.users.find({$where: "this.a + this.b >= 42"});
db.posts.group(
  { key: "user"
  , initial: {count:0, comments:0}
  , reduce: function(doc,out){
      out.count++;
      out.comments += doc.comments.length; }
  , finalize: function(out){
      out.avg = out.comments / out.count; }
  });
```

Fast, Scalable, Available, and Reliable

- Master-Slave replication for Availability and Reliability
  - Replica-Pairs support auto-negotiation for master
- Auto-Sharding for Horizontal Scalability
  - Distributes based on specified field
  - Currently alpha
- MMAP database files to automatically use available RAM
- Asynchronous modifications


shard1
shard2
shard3
shard4
S11 mongod
S21 mongod
S31 mongod
S41 mongod
S12 mongod
S22 mongod
S32 mongod
S42 mongod
config servers
C1 mongod
C2 mongod
C3 mongod
mongos
mongos
client

1. What is MongoDB?
   - Document Oriented
   - JavaScript Enabled
   - Fast, Scalable, Available, and Reliable
2. What Makes Mongo Special?
   - Native Language Integration
   - Rich Data Types
   - Atomic Modifiers
   - Dynamic Queries
3. MapReduce
   - Built-In MapReduce
   - Easy Hadoop-Mongo Integration
   - Better Hadoop-Mongo Integration

Official

- Java/JVM
- Python
- Ruby
- C/C++
- Perl
- PHP

Community Supported

Closure, Scala, C#, Haskell, Erlang, *and More*

JSON

- String (UTF8)
- Double
- Object (hash/map/dict)
- Array
- Bool
- Null / Undefined

Extras

- Date
- Int32 / Int64
- ObjectID (12 bytes: timestamp + host + pid + counter)
- Binary (with type byte)

- $set
- $inc
- $multiply (soon)
- $push / $pushAll
- $pull / $pullAll

```
db.posts.update({_id:SOMEID}, {$push:{tags:"mongodb"}})
db.tags.update({_id:"mongodb"}, {$inc:{count:1}},
                                {upsert:true})
```

Simple

```
db.posts.findOne({ user: "mstearn" });

var cursor = db.posts.find({ user: "mstearn" });
cursor.forEach(function(){
  doSomething(this.text);
});
```

Sorted

```
db.posts.find(
   { user: "mstearn" }
).sort({timestamp:-1})
```

Paginated

```
db.posts.find(
   { user: "mstearn" }
).sort({timestamp:-1}).skip(10).limit(10);
```

Simple Tag Search

```
db.posts.find(
   { user: "mstearn"
   , tags: "mongo"
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

Complex Tag Search

```
db.posts.find(
   { user: "mstearn"
   , tags: {$in: ["mongo", "mongodb"]}
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

Nested Objects

```
db.posts.find(
   { user: "mstearn"
   , tags: {$in: ["mongo", "mongodb"]}
   , comments.user: "mdirolf"
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

Regular Expressions

```
db.posts.find(
   { user: "mstearn"
   , tags: {$in: ["mongo", "mongodb"]}
   , comments.user: "mdirolf"
   , text: /windows/i
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

Ranges

```
db.posts.find(
   { user: "mstearn"
   , tags: {$in: ["mongo", "mongodb"]}
   , comments.user: "mdirolf"
   , text: /windows/i
   , points: {$gt: 10, $lt: 100}
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

Arbitrary JavaScript

```
db.posts.find(
   { user: "mstearn"
   , tags: {$in: ["mongo", "mongodb"]}
   , comments.user: "mdirolf"
   , text: /windows/i
   , points: {$gt: 10, $lt 100}
   , $where: "this.a + this.b >= 42"
   }
).sort({timestamp:-1}).skip(10).limit(10);
```

```
db.posts.mapReduce(
   function() {
     this.comments.forEach(c){
       emit(c.user,
            {count:1, words:c.text.split().length; } }
 , function(key, values){
     for (var i=1; i<values.length; i++){
       values[0].count += values[i].count;
       values[0].words += values[i].words; }
     return values[0]; }
 , { finalize: function(out){
       out.avg = out.words / out.count;
       return out; }
   , query: {posted: {$gt: new Date(2010,0,1)}}
   , out: 'posts.comment_stats'
   }
 });
```

Easy Hadoop-Mongo Integration

- mongoexport can export to JSON/CSV/TSV
  - Can also easily use a custom script
- Process in Hadoop
- Use mongoimport to get data back into MongoDB

Better Hadoop-Mongo Integration

- mongodump writes a stream of BSON to a file
- Write an InputFilter and RecordReader to read BSON
- Write a BSONWriter class to directly use the data
  - Just added two methods to driver to make this easier
- Process the data with the Java/Scala/Closure driver
- Write a custom RecordWriter to either:
  - Dump to a file and use mongorestore
  - Dump the output directly to MongoDB
- Optional: use renameCollection to mimic our MapReduce

Upcoming events

- NoSQL Live! from Boston (March 11)
- MongoDB Training in San Francisco (March 25)
- San Fransisco MySQL Meetup (April 12)

Links

- http://mongo.kylebanker.com (Try mongo in your browser)
- http://www.mongodb.org
- #mongodb on irc.freenode.net
- mongodb-user on google groups

- mathias@10gen.com
- @mathias_mongo on twitter