mongoDB in 5 min

will shulman
XLDB - SLAC
10.06.2010
Scalable, open-source, high-performance, document-oriented database
mongoDB designed to address two modern DB problems

1. object / relational “impedance mismatch”

2. horizontal scalability (going humongous)
- memcached
- key / value
- RDBMS

mongoDB
key architectural differentiators

1. relational model -> “document” model
2. no joins
3. no “complex” transactions
relational -> document-oriented

• relational
  – database
  – table
  – row

• mongoDB
  – database
  – collection
  – document / object
documents (JSON objects)

{
    _id: 1234,
    author: { name: "Bob Jones", email: "b@b.com" },
    post: "In these troubled times I like to …",
    date: { $date: "2010-07-12 13:23UTC" },
    location: [ -121.2322, 42.1223222 ],
    rating: 2.2,
    comments: [
        { user: "jgs32@hotmail.com",
          upVotes: 22,
          downVotes: 14,
          text: "Great point! I agree" },
        { user: "holly.davidson@gmail.com",
          upVotes: 421,
          downVotes: 22,
          text: "You are a moron" }
    ]
}
flexible ‘“schemas”’

```json
{
  _id: 1234,
  author: { name: "Bob Jones", email: "b@b.com" },
  post: "In these troubled times I like to …",
  date: { $date: "2010-07-12 13:23UTC" },
  location: [ -121.2322, 42.1223222 ],
  rating: 2.2,
  comments: [
    { user: "jgs32@hotmail.com",
      upVotes: 22,
      downVotes: 14,
      text: "Great point! I agree" },
    { user: "holly.davidson@gmail.com",
      upVotes: 421,
      downVotes: 22,
      text: "You are a moron" }
  ],
  tags: [ "politics", "Virginia" ]
}
```

note on internals...

- documents stored as BSON (Binary JSON)

```json
{
  "hello": "world"
}
```

- memory-mapped files

- indexes are B-Trees
dynamic queries

db.posts.find({ author.name: "mike" })
db.posts.find({ rating: { $gt: 2 } })
db.posts.find({ tags: "software" })
db.posts.find().sort({date: -1}).limit(10)
atomic update operators

Comment c = {author: “will”,
             date: new Date(),
             text: “great post!”}

db.posts.update({_id: post._id},
                 {$push: {comments: c}})
aggregation and map/reduce
going humongous
scaling

• replication for read scalability
  – Master / slave
  – Replica sets with automatic failover

• sharding for write scalability
  – auto-sharding
  – designed to support thousands of nodes
    • (current limit of 20 PB)
clustering architecture
real-world use cases of scale

• Server Density
  – 17k collections
  – 43k indexes
  – 1B documents
• Bit.ly
  – 1.25B shortens per month
• Wordnik
  – 5 billion documents
using the database

• Interactive Javascript shell
• Drivers in most languages
  – Java
  – Javascript
  – Python
  – Ruby
  – C#
  – PHP
  – and more...
other great stuff

- geospatial indexing
- regex search
- gridFS
street cred

- Shutterfly
- Foursquare
- bit.ly
- Sourceforge
- Etsy
- The New York Times
- Business Insider
- Github
- Gilt Groupe
- Sugar CRM
- Electronic Arts
- Evite
- CollegeHumor
- Disqus
- Justin.tv
- Chartbeat
- Hot potato
- Eventbrite
for more info...

• MongoDB / 10Gen
  – http://mongodb.org
  – http://10gen.com

• MongoLab (MongoDB cloud hosting)
  – http://mongolab.com
  – will@mongolab.com