Intro to MongoDB

Alex Sharp

twitter: @ajsharp

email: ajsharp@frothlogic.com

So what is MongoDB?

First and foremost...

IT’S THE NEW HOTNESS!!!

omgomgomg
SHINY OBJECTS
omgomgomg

*MongoDB (from "hu**mongo**us") is a scalable, high-performance, open source,* schema-free, document-oriented database.

- mongodb.org

Philosophy

Philosophy

“One size fits all” approach no longer applies

Philosophy

Non-relational DBs scale more easily, especially horizontally

Philosophy

Focus on speed, performance, flexibility and scalability

Philosophy

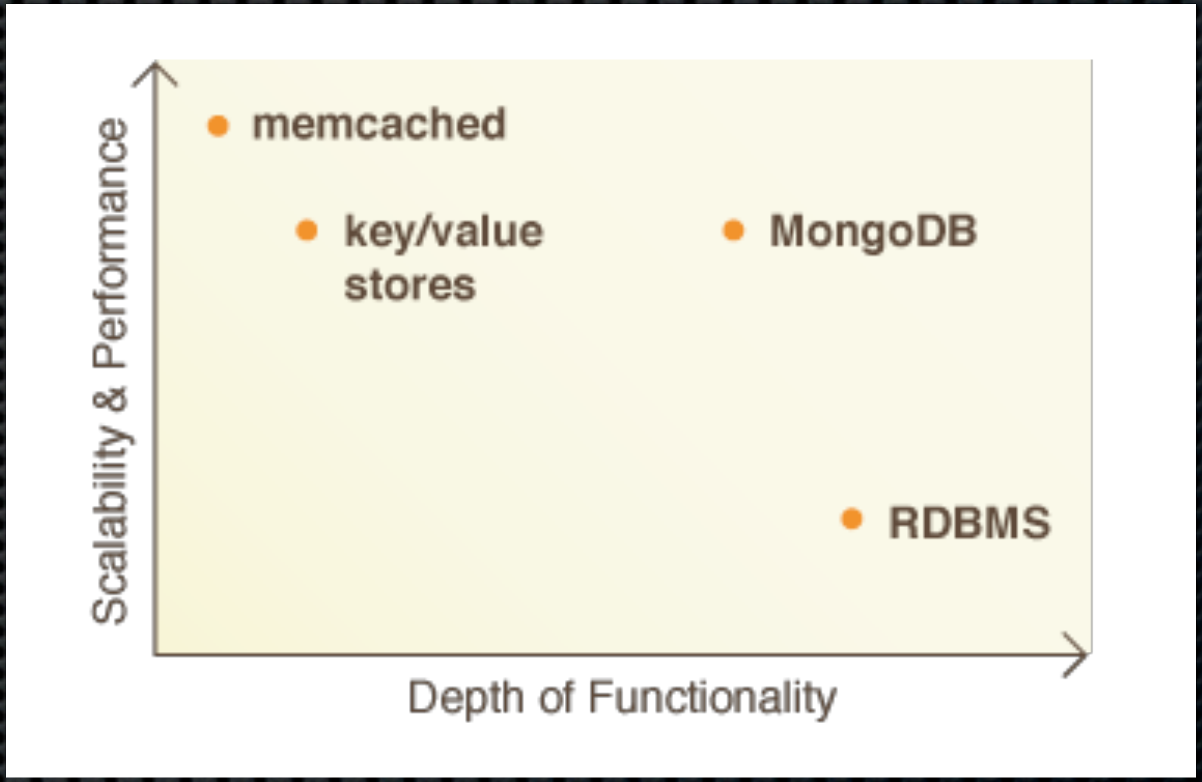
Not concerned with transactional stuff and relational semantics

Philosophy

DBs should be an on-demand commodity, in a cloud-like fashion

Philosophy

Mongo tries to achieve the performance of traditional key-value stores while maintaining functionality of traditional RDBMS



Features

Features

- Standard database stuff

Features

- Standard database stuff
  - Indexing

Features

- Standard database stuff
  - Indexing
  - replication/failover support

Features: Document Storage

Documents are stored in BSON (binary JSON)

Features: Document Storage

BSON is a binary serialization of JSON-like objects

Features: Document Storage

This is **extremely** powerful, b/c it means mongo understands JSON natively

Features: Document Storage

Any valid JSON can be easily imported and queried

Features

Schema-less; very flexible

Features

Schema-less; very flexible

no more blocking ALTER TABLE

Features

Auto-sharding (alpha)

Features

Makes for easy horizontal scaling

Features

Map/Reduce

Features

Very, very fast

Features

Super easy to install

Features

Strong with major languages

Features

Document-oriented = flexible

Features: Querying

- Rich, javascript-based query syntax

Features: Querying

- Rich, javascript-based query syntax
  - Allows us to deep, nested queries

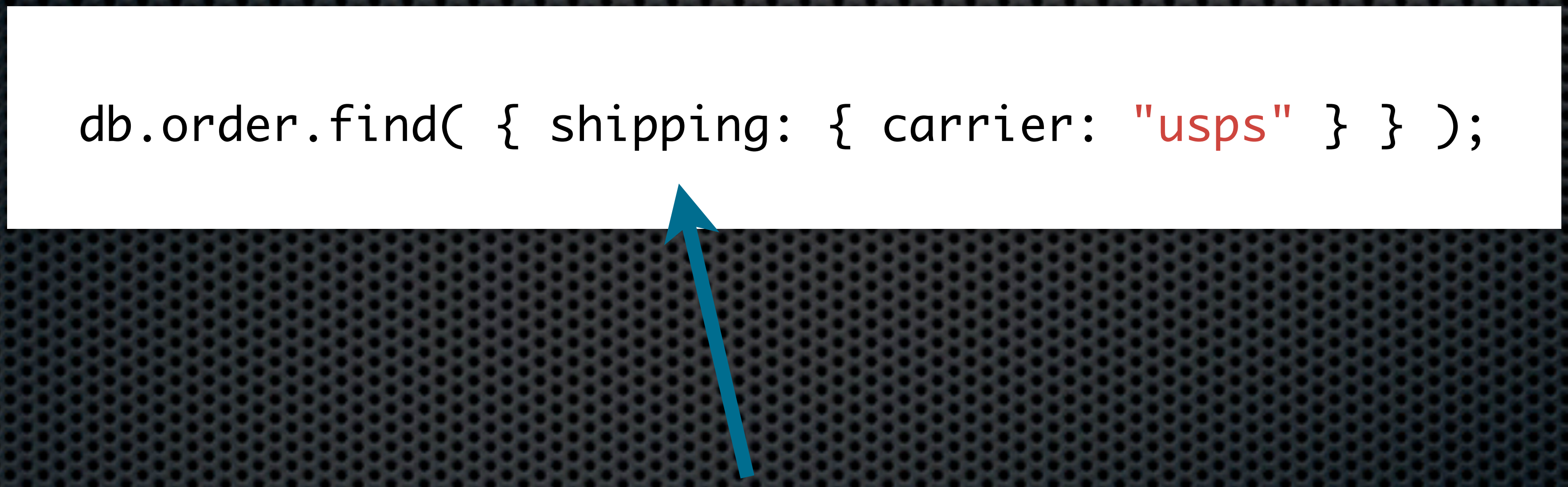
Features: Querying

- Rich, javascript-based query syntax
  - Allows us to do deep, nested queries

```
db.order.find( { shipping: { carrier: "usps" } } );
```

Features: Querying

- Rich, javascript-based query syntax
  - Allows us to deep, nested queries



```
db.order.find( { shipping: { carrier: "usps" } } );
```

shipping is an embedded document (object)

Features: Binary Object Store

- Efficient binary large object store via GridFS

Features: Binary Object Store

- Efficient binary large object store via GridFS
  - i.e. store images, videos, anything

Concepts

Concepts: Document-oriented

- Think of “documents” as database records

Concepts: Document-oriented

- Think of “documents” as database records
- Documents are basically just JSON objects that Mongo stores in binary

Concepts: Document-oriented

- Think of “collections” as database tables

Concept Mapping

RDBMS (mysql, postgres)	MongoDB
Tables	Collections

Concept Mapping

RDBMS (mysql, postgres)	MongoDB
Tables	Collections
Records/rows	Documents/objects

Concept Mapping

RDBMS (mysql, postgres)	MongoDB
Tables	Collections
Records/rows	Documents/objects
Queries return record(s)	Queries return a cursor

Concept Mapping

RDBMS (mysql, postgres)	MongoDB
Tables	Collections
Records/rows	Documents/objects
Queries return record(s)	**Queries return a cursor**

???

Concepts: Cursors

- Queries return “cursors” instead of collections

Concepts: Cursors

- Queries return “cursors” instead of collections
  - A cursor allows you to iterate through the result set

Concepts: Cursors

- Queries return “cursors” instead of collections
  - A cursor allows you to iterate through the result set
  - A big reason for this is performance

Concepts: Cursors

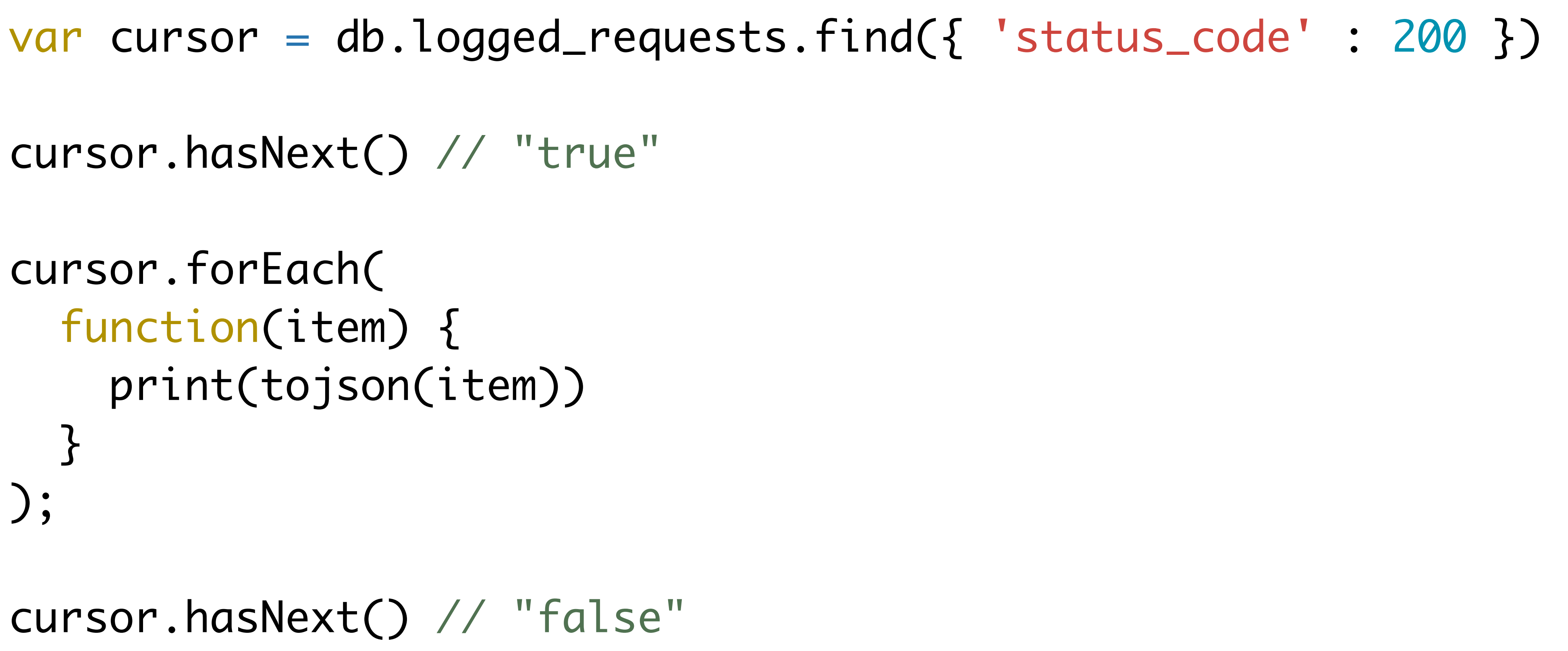
- Queries return “cursors” instead of collections
    - A cursor allows you to iterate through the result set
    - A big reason for this is performance
    - Much more efficient than loading all objects into memory

Concepts: Cursors

- The find() function returns a cursor object

Concepts: Cursors

- The find() function returns a cursor object



```
var cursor = db.logged_requests.find({ 'status_code' : 200 })

cursor.hasNext() // "true"

cursor.forEach(
  function(item) {
    print(tojson(item))
  }
);

cursor.hasNext() // "false"
```

Cool Features

Cool Features

- Capped collections

Cool Features

- Capped collections
  - Fixed-sized, limited operation, auto-LRU age-out collections

Cool Features

- Capped collections
  - Fixed-sized, limited operation, auto-LRU age-out collections
  - Fixed insertion order

Cool Features

- Capped collections
  - Fixed-sized, limited operation, auto-LRU age-out collections
  - Fixed insertion order
  - Super fast

Cool Features

- Capped collections
  - Fixed-sized, limited operation, auto-LRU age-out collections
  - Fixed insertion order
  - Super fast
  - Ideal for logging and caching

Cool Uses

- Data Warehouse
  - Mongo understands JSON natively

Cool Uses

- Data Warehouse
  - Mongo understands JSON natively
  - Very powerful for analysis

Cool Uses

- Data Warehouse
  - Mongo understands JSON natively
  - Very powerful for analysis
  - Query a bunch of data from some web service

Cool Uses

- Data Warehouse
  - Mongo understands JSON natively
  - Very powerful for analysis
  - Query a bunch of data from some web service
  - Import into mongo (mongoimport -f filename.json)

Cool Uses

- Data Warehouse
  - Mongo understands JSON natively
  - Very powerful for analysis
  - Query a bunch of data from some web service
  - Import into mongo (mongoimport -f filename.json)
  - Analyze to your heart’s content

Cool Uses

- Harmonyapp.com
  - Large rails app for building websites (kind of a CMS)

Cool Uses

- Hardcore debugging
  - Spit out large amounts of data

Limitations

- Transaction support

Limitations

- Transaction support
- Relational integrity

Resources

- http://mongodb.org
  - http://www.mongodb.org/display/DOCS/Tutorial
  - http://www.mongodb.org/display/DOCS/Use+Cases
- http://blog.mongodb.org/post/172254834/mongodb-is-fantastic-for-logging
- http://github.com/ajsharp/mongo-conf