Go and MongoDB

Go Language New York Meetup

{ by: "Ryan Nitz", date: "2013-03-12" }

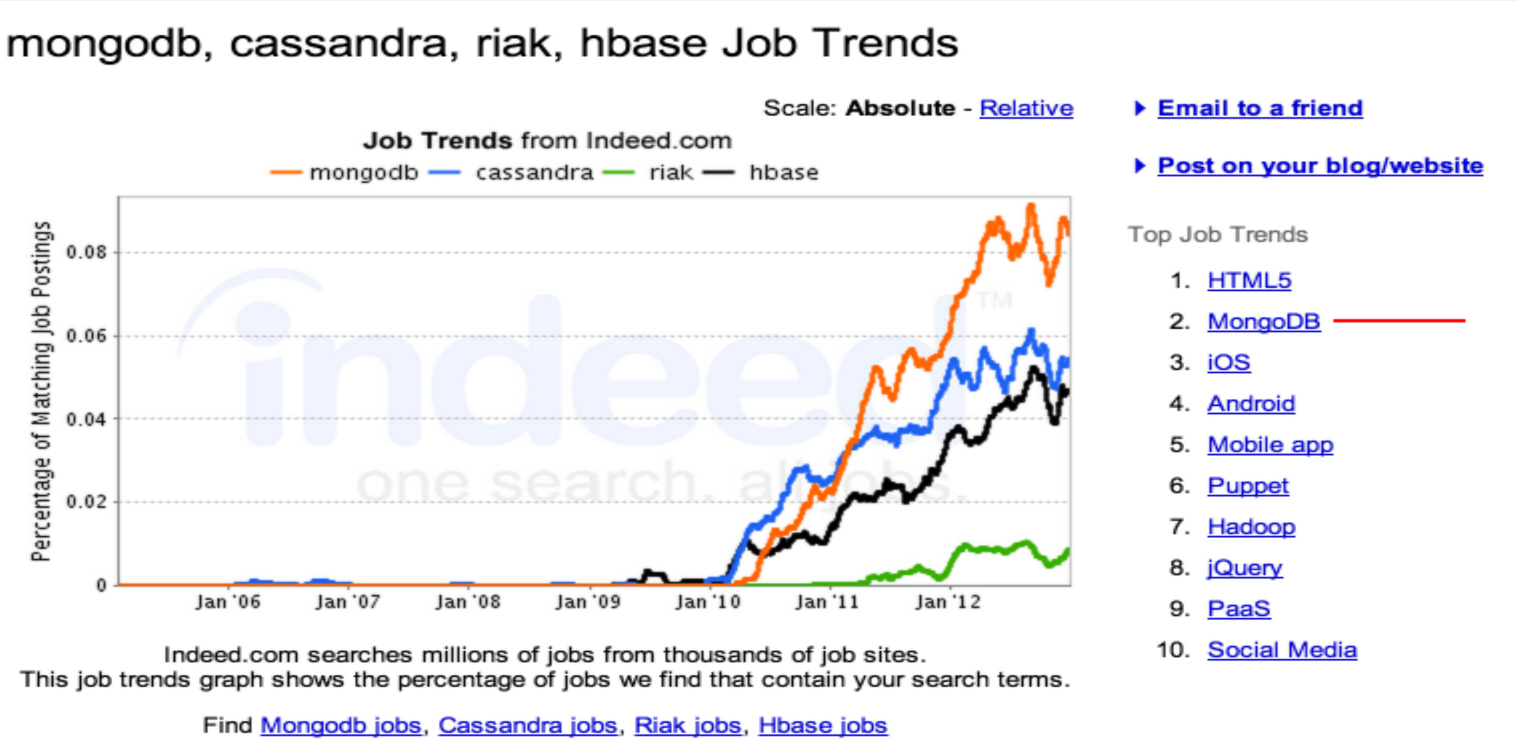
Overview

- MongoDB Introduction
- Brief History of Go
- Available Go Drivers
- Installing
- Dialing
- Commands
- Writing
- Reading
- MMS Agent Design
- About 10gen
- Discussion

MongoDB Introduction

- Document Database

- Clear NoSQL winner



mongodb, cassandra, riak, hbase Job Trends

Scale: **Absolute** - Relative

Job Trends from Indeed.com

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.

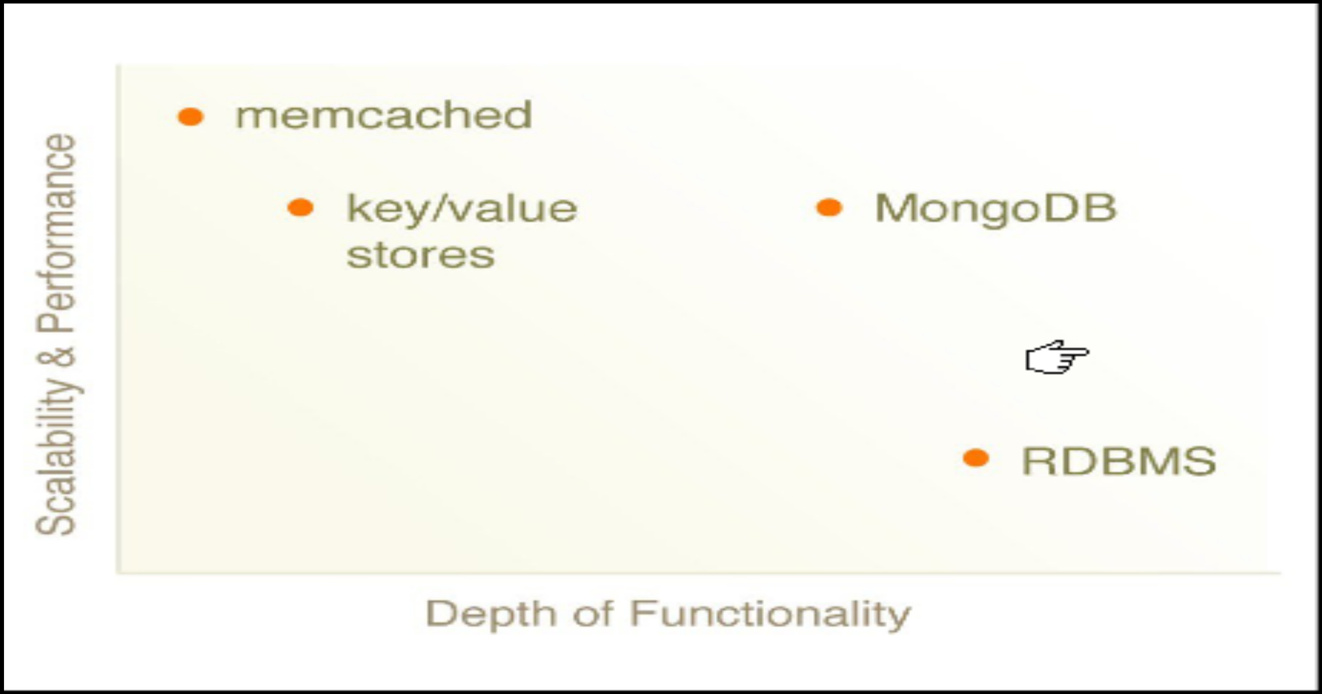
Find Mongodb jobs, Cassandra jobs, Riak jobs, Hbase jobs

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

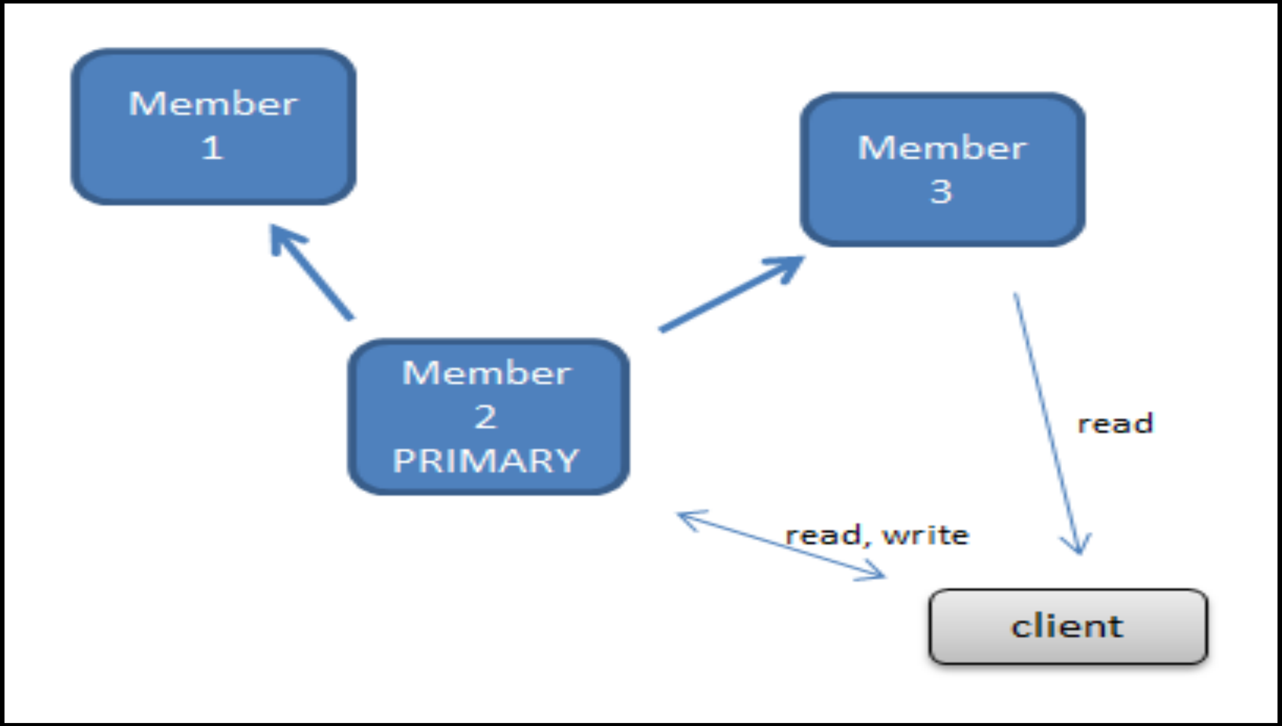
General Purpose

- large and small
- Key/value and rich query syntax
- Update-in-place



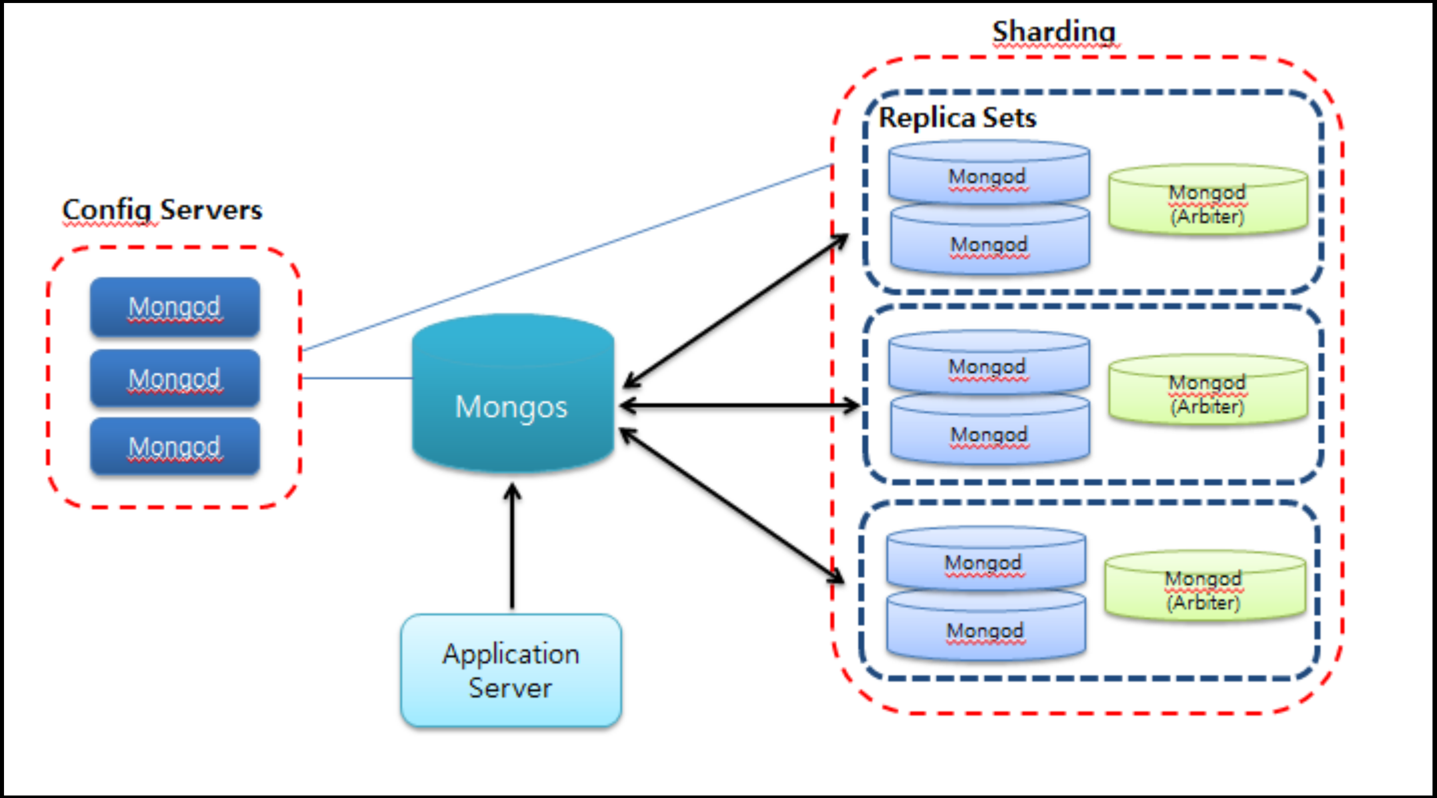
Replica Sets

- Built In Fault Tolerance via Replica Sets
  - Horizontal Read Scalability



Sharded Clusters

- Horizontal Scalability
  - Horizontal Read/Write Scalability



Document/BSON Overview

- BSON == Binary Json
- Full Spec - http://bsonspec.org/

```
{
    _id     : ObjectId("4d1b6314e528c81a1f200e03"),
    name : "BSON Example",
    date   : ISODate("2013-03-13T19:51:34.672Z"),
    count : NumberLong(999999999999)
    historicalCounts : [ 20, 50, 100, 200, 300 ],
    types   : { first : { name : "test" }, second : { name : "foo" } }
}
```

Brief Go History

- Development started in 2007
- TIOBE Programming Language of the Year - 2009
- 1.0 Released in 2012
- Statically typed compiled language with the ease of programming of a dynamic language (source: Wikipedia)
- Garbage collection - low overhead, no significant latency
- Intuitive concurrency
- Developers
    - Robert Griesemer
    - Rob Pike
    - Ken Thompson

MongoDB Go Drivers

- All Go Drivers are Community Supported

- mgo appears to be the best choice right now
  - pronounced "mango" #shrug

- **mgo** - http://labix.org/mgo

- **gmongo** - https://github.com/poiati/gmongo

- **go-mongo** - https://github.com/garyburd/go-mongo

Installing

- Install go - http://golang.org/doc/install

- Install bzr - http://bazaar.canonical.com/en/
  - Required by mgo
  - OS X: brew install bzr

- Create a dev directory and set GOPATH
  - export GOPATH="$HOME/your-app/"

- Create the "src" directory and pull mgo in the src dir
  - go get labix.org/v2/mgo

Dialing

- Dial is a Go term for connect

Dialing

```
session, err := mgo.Dial("mongodb://localhost:27017/test")
if err != nil {
    panic(err)
}
```

- Default timeout is ten seconds
- To control timeout, use mgo.DialWithTimeout

```
var timeout = 30 * time.Second
session, err := mgo.DialWithTimeout("mongodb://localhost:27017/test", timeout)
if err != nil {
    panic(err)
}
```

Dialing

- Configure the session using SetMode func
  - Eventual - reads go to secondaries
  - Monotonic - reads go to secondaries until a write is issued
  - Strong - all reads/writes go to primaries
- Monotonic - If you refresh the session, it will reset the state back to reading from secondaries

Commands

- Commands are executed via the Run function

```
var serverStatus = &bson.M{}
if err := session.Run("serverStatus", serverStatus); err != nil {
    panic(err)
}
```

- A command with a param

```
var startupWarnings = &bson.M{}
if err := session.Run(bson.D{{"getLog", "startupWarnings"}},
startupWarnings); err != nil {
      panic(err)
}
```

Inserting

- Create the map/bson object
- Set the fields
- From the session, access the database
- From the database access the collection
- Perform operation

```
var testDoc = bson.M{}
testDoc["name"] = "Ryan Nitz"
testDoc["testId"] = bson.NewObjectId()
testDoc["slice"] = []string { "one", "two", "three", "four" }
session.DB("test").C("golangny").Insert(&testDoc)
```

Inserting

- mgo also supports struct marshalling
  - Field case is important

```
type Dog struct {
    Name string
    Drools bool
    AlwaysHungry bool
    HoursSleptPerDay int
}
session.DB("test").C("golangny").Insert(&Dog{Name: "Loo", Drools: false, AlwaysHungry: true, HoursSleptPerDay: 18})
```

Reading

- Reading a single document

```
var doc = &bson.M{}
if err := coll.Find(bson.M{"testId": testId}).One(doc); err != nil {
    if err == mgo.ErrNotFound {
      fmt.Println("The document was not found")
    } else {
        panic(err)
    }
}
```

Reading

- Reading multiple documents

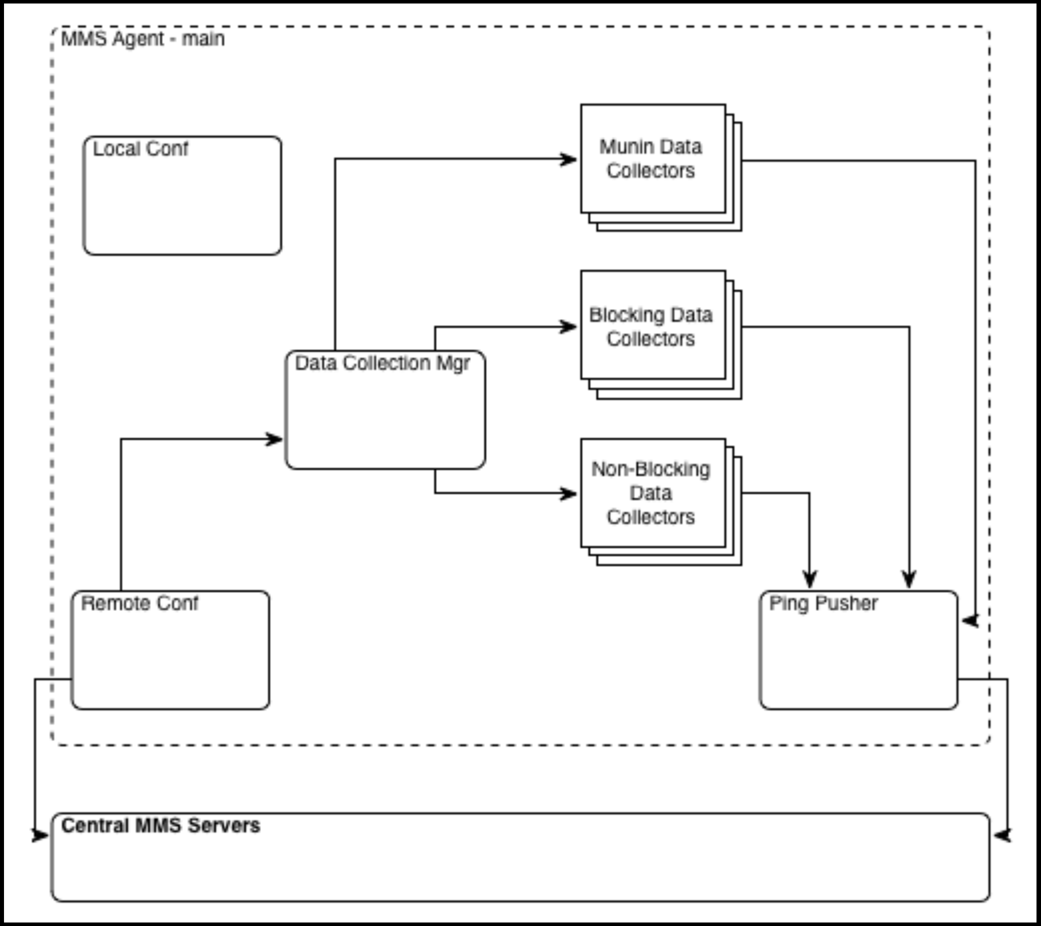
```
iter := session.DB("test").C("golangny").Find(nil).Iter()
var result = &bson.M{}
for iter.Next(&result) {
    fmt.Printf("Result: %v\n", *result)
}
if iter.Err() != nil {
    panic(iter.Err())
}
```

MMS Agent Project

- Current impl in Python
  - Installing Python packages is painful
  - Python global lock is painful
  - Python memory consumption is high
  - Python is slow
  - Non-compiled languages are a PITA
  - Supporting python 2.4.3 - 3+ is a PITA
    - Python 3 broke backwards compatibility
  - Supporting pymongo 1.9+ is challenging
  - Users modify source

- New impl in Go :-)

MMS Agent Project



About 10gen - www.10gen.com

- Supports MongoDB OS project
- Founded in 2007
- Funding to date: $73.4mm
- ~ 190 employees
  - ~ 280 employees by EOY
  - 10gen is hiring - www.10gen.com/careers
  - Offices in: NYC, Palo Alto, London, Dublin, Reston, Barcelona, Sydney
- Funds OS project by providing
  - Commercial Support
  - Consulting
  - Development Support
  - Training
  - Enterprise Edition

Questions

Thanks!

Ryan Nitz - https://github.com/rgnitz

Presentation - http://bit.ly/13RynfY