Introducing Riak and Ripple

Sean Cribbs

system “whoami”

system “whoami”

- BCS, three years as a musician

system “whoami”

- BCS, three years as a musician
- Started with Ruby in early 2006

system “whoami”

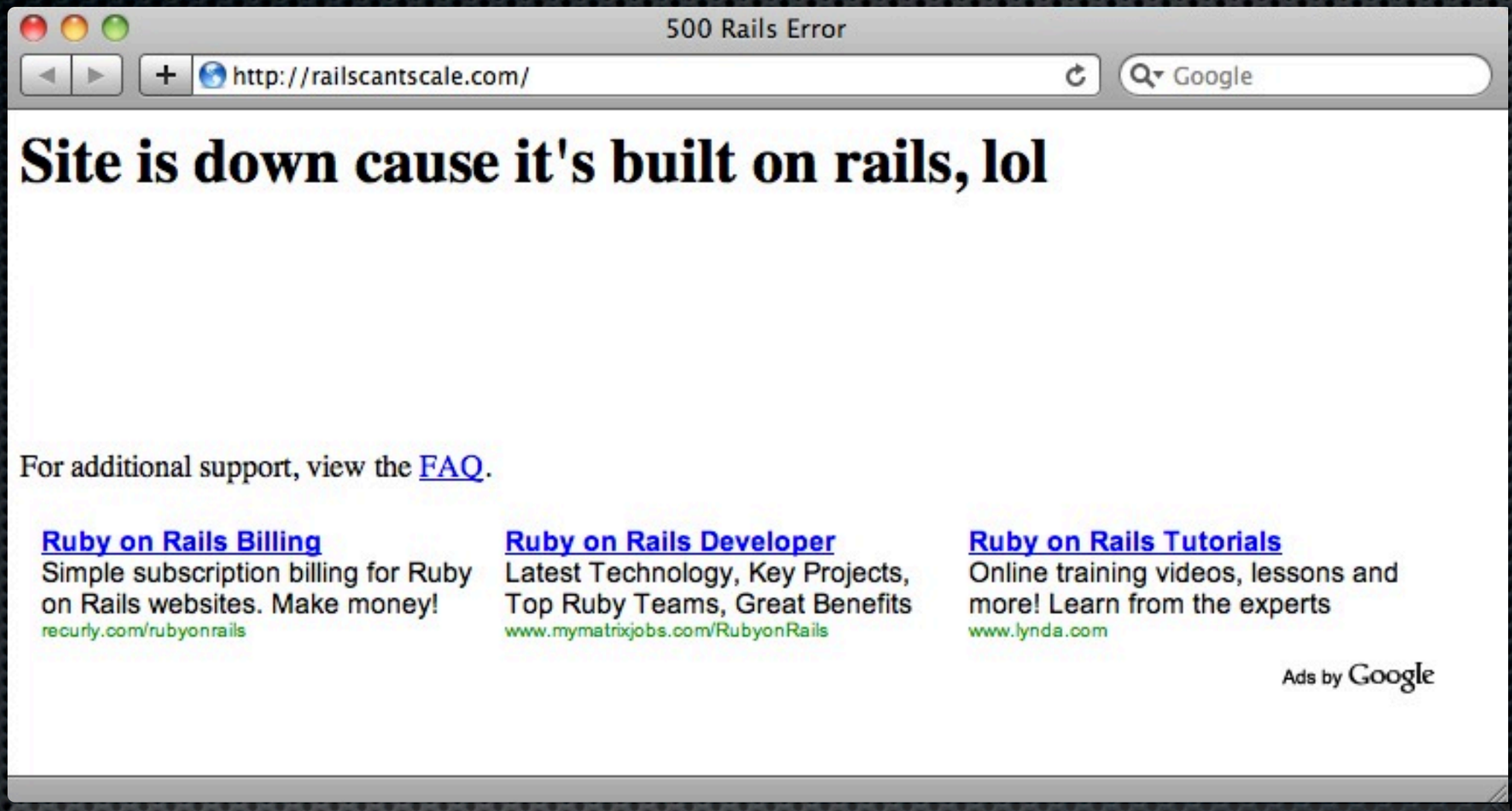
- BCS, three years as a musician
- Started with Ruby in early 2006
- Rails freelancer 2007-2010

system “whoami”

- BCS, three years as a musician
- Started with Ruby in early 2006
- Rails freelancer 2007-2010
- Radiant CMS

system “whoami”

- BCS, three years as a musician
- Started with Ruby in early 2006
- Rails freelancer 2007-2010
- Radiant CMS
- March 2010 - Developer Advocate


500 Rails Error
http://railscantscale.com/
Google
Site is down cause it's built on rails, lol
For additional support, view the FAQ.
Ruby on Rails Billing
Simple subscription billing for Ruby on Rails websites. Make money!
recurly.com/rubyonrails
Ruby on Rails Developer
Latest Technology, Key Projects, Top Ruby Teams, Great Benefits
www.mymatrixjobs.com/RubyonRails
Ruby on Rails Tutorials
Online training videos, lessons and more! Learn from the experts
www.lynda.com
Ads by Google

Rails Can't Scale

Rails Can't Scale

what does it mean?

Ruby ~~Rails~~ Can't Scale

what does it mean?

MySQL
~~Ruby~~ ~~Rails~~ Can't Scale

what does it mean?

insert straw man here

~~MySQL~~ ~~Ruby~~ ~~Rails~~ Can't Scale

what does it mean?

Mythbusting “scalability”

Mythbusting “scalability”

- Scalability is not a yes/no question

Mythbusting “scalability”

- Scalability is not a yes/no question
- It’s a ratio of **benefit** to **cost**

Mythbusting “scalability”

- Scalability is not a yes/no question
- It’s a ratio of **benefit** to **cost**
- **Benefits**: low latency, throughput, uptime, concurrency, reliability

Mythbusting “scalability”

- Scalability is not a yes/no question
- It’s a ratio of **benefit** to **cost**
  - **Benefits**: low latency, throughput, uptime, concurrency, reliability
  - **Costs**: CPU, RAM, disk, bandwidth, power, hw/sw

Mythbusting “scalability”

- Scalability is not a yes/no question
- It’s a ratio of **benefit** to **cost**
  - **Benefits**: low latency, throughput, uptime, concurrency, reliability
  - **Costs**: CPU, RAM, disk, bandwidth, power, hw/sw
- **scalability = bang for your buck (ROI)**

Scaling goes both ways

Scaling goes both ways

- **Up** - high traffic, “Big Data”, multi-datacenter

Scaling goes both ways

twitter YAHOO! Google™

- **Up** - high traffic, “Big Data”, multi-datacenter
- **Down** - laptop, set-top, mobile

How do you scale?

How do you scale?

- Vertically - get bigger, expensive machines

How do you scale?

- Vertically - get bigger, expensive machines
- Horizontally - get more commodity machines

What is Riak?

(the horizontal scaling bits)

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)
  - key-value storage

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)
  - key-value storage
  - masterless, peer-to-peer replication

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)
  - key-value storage
  - masterless, peer-to-peer replication
  - consistent hashing

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)
  - key-value storage
  - masterless, peer-to-peer replication
  - consistent hashing
  - eventual consistency

What is Riak?

(the horizontal scaling bits)

- Based on Amazon's Dynamo (2007)
  - key-value storage
  - masterless, peer-to-peer replication
  - consistent hashing
  - eventual consistency
  - failover - quorums, hinted handoff

"O/RM is the Vietnam of Computer Science."

-- Ted Neward, 2004/6

"O/RM is the Vietnam of Computer Science."

-- Ted Neward, 2004/6

slippery slope

"O/RM is the Vietnam of Computer Science."

-- Ted Neward, 2004/6

slippery slope diminishing returns

Impedance mismatch

Impedance mismatch

- Relational = predicate logic, truth statements

Impedance mismatch

- Relational = predicate logic, truth statements
- Object = identity, state, behavior, encapsulation

Pick your poison

Pick your poison

- Make the objects fit the database

Pick your poison

- Make the objects fit the database
- Active Record

Pick your poison

- Make the objects fit the database
  - Active Record
- Make the database fit the objects

Pick your poison

- Make the objects fit the database
  - Active Record
- Make the database fit the objects
  - OODBMS, Graph DBs

Middle ground

Middle ground

- Document databases (*identity* and *state)*

Middle ground

- Document databases (*identity* and *state)*
- No JOINs - denormalized / composed

Middle ground

- Document databases (*identity* and *state)*
  - No JOINs - denormalized / composed
  - Loosely structured

Middle ground

- Document databases (*identity* and *state)*
  - No JOINs - denormalized / composed
  - Loosely structured
  - Friendly transport formats, e.g. JSON

What is Riak?

(the document database bits)

What is Riak?

(the document database bits)

- Store your objects as JSON (or any format)

What is Riak?

(the document database bits)

- Store your objects as JSON (or any format)
- Link between objects (like hypertext)

What is Riak?

(the document database bits)

- Store your objects as JSON (or any format)
- Link between objects (like hypertext)
- No SQL - query with Javascript Map-Reduce

“DevOps” - agile infrastructure

“DevOps” - agile infrastructure

programming your infrastructure

“DevOps” - agile infrastructure

programming your infrastructure
rapid “cloud” deployments

Befriend your sysadmin

Befriend your sysadmin

- Integrate with existing infrastructure

Befriend your sysadmin

- Integrate with existing infrastructure
- Reduce the amount of hands-on work

Befriend your sysadmin

- Integrate with existing infrastructure
- Reduce the amount of hands-on work
- Have a growth plan

What is Riak?

(the ops-friendly bits)

What is Riak?

(the ops-friendly bits)

- Web-shaped storage

What is Riak?

(the ops-friendly bits)

- Web-shaped storage
  - Store data in its original format

What is Riak?

(the ops-friendly bits)

- Web-shaped storage
  - Store data in its original format
  - It's just HTTP - same techniques apply

What is Riak?

(the ops-friendly bits)

- Web-shaped storage
  - Store data in its original format
  - It's just HTTP - same techniques apply
    - load balancing, proxy caches, round-robin DNS

What is Riak?

(the ops-friendly bits)

- Web-shaped storage
  - Store data in its original format
    - It's just HTTP - same techniques apply
      - load balancing, proxy caches, round-robin DNS
- No node is special - grow horizontally

What is Riak?

(the ops-friendly bits)

- Web-shaped storage
  - Store data in its original format
    - It's just HTTP - same techniques apply
      - load balancing, proxy caches, round-robin DNS
- No node is special - grow horizontally
- Get some sleep, fix it in the morning

To review, Riak is...

To review, Riak is...

- Magical Horizontally-Scaling Unicorns

To review, Riak is...

- Magical Horizontally-Scaling Unicorns
- Application-Friendly Rainbows in the Cloud

To review, Riak is...

- Magical Horizontally-Scaling Unicorns
- Application-Friendly Rainbows in the Cloud
- Data-Shredding MapReduce Ninjas

http://downloads.basho.com/riak/

hg clone http://hg.basho.com/riak/

gem install ripple

git clone git://github.com/seancribbs/ripple.git

CRUD in Riak

CRUD in Riak

- PUT /riak/bucket/key (C,U) (also POST)

CRUD in Riak

- PUT /riak/bucket/key (C,U) (also POST)
- GET /riak/bucket/key (R)

CRUD in Riak

- PUT /riak/bucket/key (C,U) (also POST)
- GET /riak/bucket/key (R)
- DELETE /riak/bucket/key (D)

CRUD Demo

Links

Links

- One-way, qualified “pointers” to other objects

Links

- One-way, qualified “pointers” to other objects
- “Link” header in HTTP

Links

- One-way, qualified “pointers” to other objects
- “Link” header in HTTP
- </riak/bucket/key>; riaktag=”tag”

Link-walking

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket
  - tag = scope to a tag

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket
  - tag = scope to a tag
  - keep = return results (last one always)

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket
  - tag = scope to a tag
  - keep = return results (last one always)
  - “_” means match anything

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket
  - tag = scope to a tag
    - keep = return results (last one always)
    - "_" means match anything
- Example:

Link-walking

- GET /riak/bucket/key/[bucket,tag,keep]*
  - bucket = scope to a bucket
  - tag = scope to a tag
    - keep = return results (last one always)
    - “_” means match anything
- Example:
  - GET /riak/artists/TheBeatles/albums,_,1/tracks,_,1

Links Demo

Riak Map-Reduce

I believe I did, Bob.

Riak Map-Reduce

Riak Map-Reduce

- Based on Google's Map-Reduce idea

Riak Map-Reduce

- Based on Google's Map-Reduce idea
- Some similarities to Hadoop and CouchDB

Riak Map-Reduce

- Based on Google's Map-Reduce idea
- Some similarities to Hadoop and CouchDB
- High data-locality

Riak Map-Reduce

- Based on Google's Map-Reduce idea
- Some similarities to Hadoop and CouchDB
- High data-locality
- Phases can be Javascript or Erlang

Map-Reduce Terminology

Map-Reduce Terminology

- **Job:** a query, composed of inputs and phases

Map-Reduce Terminology

- **Job:** a query, composed of inputs and phases
- **Phase:** a single map or reduce function application

Map-Reduce Terminology

- **Job:** a query, composed of inputs and phases
- **Phase:** a single map or reduce function application
- **Map:** phase applied to each item - filter, transform

Map-Reduce Terminology

- **Job:** a query, composed of inputs and phases
- **Phase:** a single map or reduce function application
- **Map:** phase applied to each item - filter, transform
- **Reduce:** phase applied to the collection - collate, aggregate, compute

Map Phases

Map Phases

- Inputs are bucket-key pairs (with optional key-specific data)

Map Phases

- Inputs are bucket-key pairs (with optional key-specific data)
- Must return a list

Map Phases

- Inputs are bucket-key pairs (with optional key-specific data)
- Must return a list
- Parallel results are aggregated

Map Built-ins

Map Built-ins

- Riak.mapValues

Map Built-ins

- Riak.mapValues
- Riak.mapValuesJson

Reduce Phases

Reduce Phases

- Performed on a single node

Reduce Phases

- Performed on a single node
- Two processes per reduce phase increase parallelism

Reduce Phases

- Performed on a single node
- Two processes per reduce phase increase parallelism
- Must return a list

Reduce Built-ins

Reduce Built-ins

- `Riak.reduceMin`

Reduce Built-ins

- `Riak.reduceMin`
- `Riak.reduceMax`

Reduce Built-ins

- `Riak.reduceMin`
- `Riak.reduceMax`
- `Riak.reduceSort`

Build a M/R Job

Build a M/R Job

- Specify inputs

Build a M/R Job

- Specify inputs
  - bucket/key, bucket/key/key-specific-arg, bucket

Build a M/R Job

- Specify inputs
  - bucket/key, bucket/key/key-specific-arg, bucket
- Specify phases

Build a M/R Job

- Specify inputs
  - bucket/key, bucket/key/key-specific-arg, bucket
- Specify phases
  - Each can receive a static argument

Build a M/R Job

- Specify inputs
  - bucket/key, bucket/key/key-specific-arg, bucket
- Specify phases
  - Each can receive a static argument
  - Each can return intermediate results

Map-Reduce on HTTP

Map-Reduce on HTTP

- Job is a JSON object

Map-Reduce on HTTP

- Job is a JSON object
- POST /mapred

A simple M/R job

```
{“inputs”: “goog”,
 “query”: [{“map”:
          {“language”:”javascript”,
           “name”: “Riak.mapValuesJson”,
           “keep”: true}]}
```

A simple M/R job

```
{“inputs”: “goog”,
 “query”: [{“map”:
            {“language”:”javascript”,
             “name”: “Riak.mapValuesJson”,
             “keep”: true}]}

Riak::MapReduce.new(c).add(‘goog’).
     map(‘Riak.mapValuesJson’, :keep => true).run
```

Another M/R Job

```
{“inputs”: “stocks”,
 “query”: [{“map”:{“language”:”javascript”,
                   “name”: “App.extractTickers”,
                   “arg”: “GOOG”,
                   “keep”: false},
           {“reduce”:{“language”:”javascript,
                      “name”: “Riak.reduceMin”,
                      “keep”: true}]}
```

Another M/R Job

```
{“inputs”: “stocks”,
 “query”: [{“map”:{“language”:”javascript”,
                   “name”: “App.extractTickers”,
                   “arg”: “GOOG”,
                   “keep”: false},
           {“reduce”:{“language”:”javascript,
                      “name”: “Riak.reduceMin”,
                      “keep”: true}]}

Riak::MapReduce.new(c).add(‘stocks’).
     map(‘App.extractTickers’, :arg =>”GOOG”).
     reduce(“Riak.reduceMin”, :keep => true).run
```

Map-Reduce Demo

Ripple

Ripple

- Document-style persistence (like MongoMapper)

Ripple

- Document-style persistence (like MongoMapper)
- ActiveModel niceties

Ripple

- Document-style persistence (like MongoMapper)
- ActiveModel niceties
- Callbacks

Ripple

- Document-style persistence (like MongoMapper)
- ActiveModel niceties
- Callbacks
- Validations

Ripple

- Document-style persistence (like MongoMapper)
- ActiveModel niceties
- Callbacks
- Validations
- ActionPack compatibility

Ripple TODO

Ripple TODO

- **Associations**

Ripple TODO

- **Associations**
- **Indexes**

Ripple TODO

- **Associations**
- **Indexes**
- Use a property/method as the key

Ripple TODO

- **Associations**
- **Indexes**
- Use a property/method as the key
- Dynamic finders

Ripple TODO

- **Associations**
- **Indexes**
- Use a property/method as the key
- Dynamic finders
- Session/cache stores

Ripple Demo

sean@basho.com

@seancribbs

“seancribbs” on Freenode IRC #riak

Questions?

http://wiki.basho.com

riak-users@lists.basho.com