Introduction to Riak & Ripple

Scott Lystig Fritchie
Senior Software Engineer

scott@basho.com
@slfritchie

Riak @35,000 ft.

- Amazon Dynamo-inspired
  replicated, distributed, fault-tolerant, masterless, scalable, operations-friendly
- Key-Value / Document
- Schema-less, content-agnostic
- Web-friendly
  HTTP, JSON, Javascript
- Both kinds of free: as in beer, as in liberty

Dynamo-like Scalability

- To get...
  - more storage,
  - more throughput,
  - lower latency...
- ...add more machines.
  aka horizontal & linear

Key-Value++

- Data **objects** are identified by **keys**, and have **metadata**
- Keys are grouped in **buckets**
- **Links** enable lightweight relationships
- Query with **MapReduce**
- (Optional) Automatic, Solr-compatible full-text indexing of the value

Schema-less

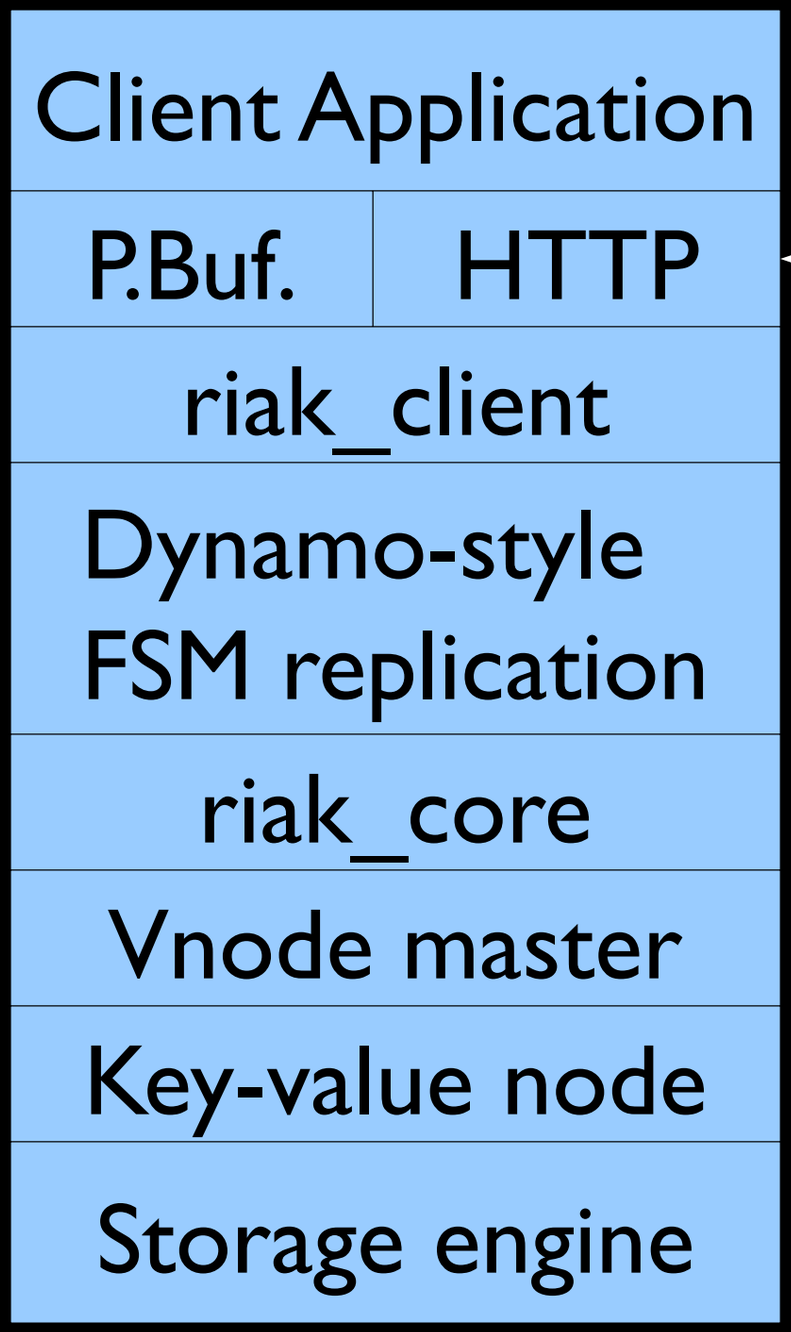
- Buckets created **on the fly**
- Values are **opaque**
- **Content-Type** matters
- The **application** defines the semantics:
  more flexibility, more responsibility
- Schema recommended for full-text search

Web-Friendly

- **HTTP** is primary interface
- **JSON** is used for structured data
- **Javascript** is used for MapReduce functions
- Plays well with **Varnish**, **Squid**, **HAProxy**, **F5**, etc.

[*see also* **Webmachine**]

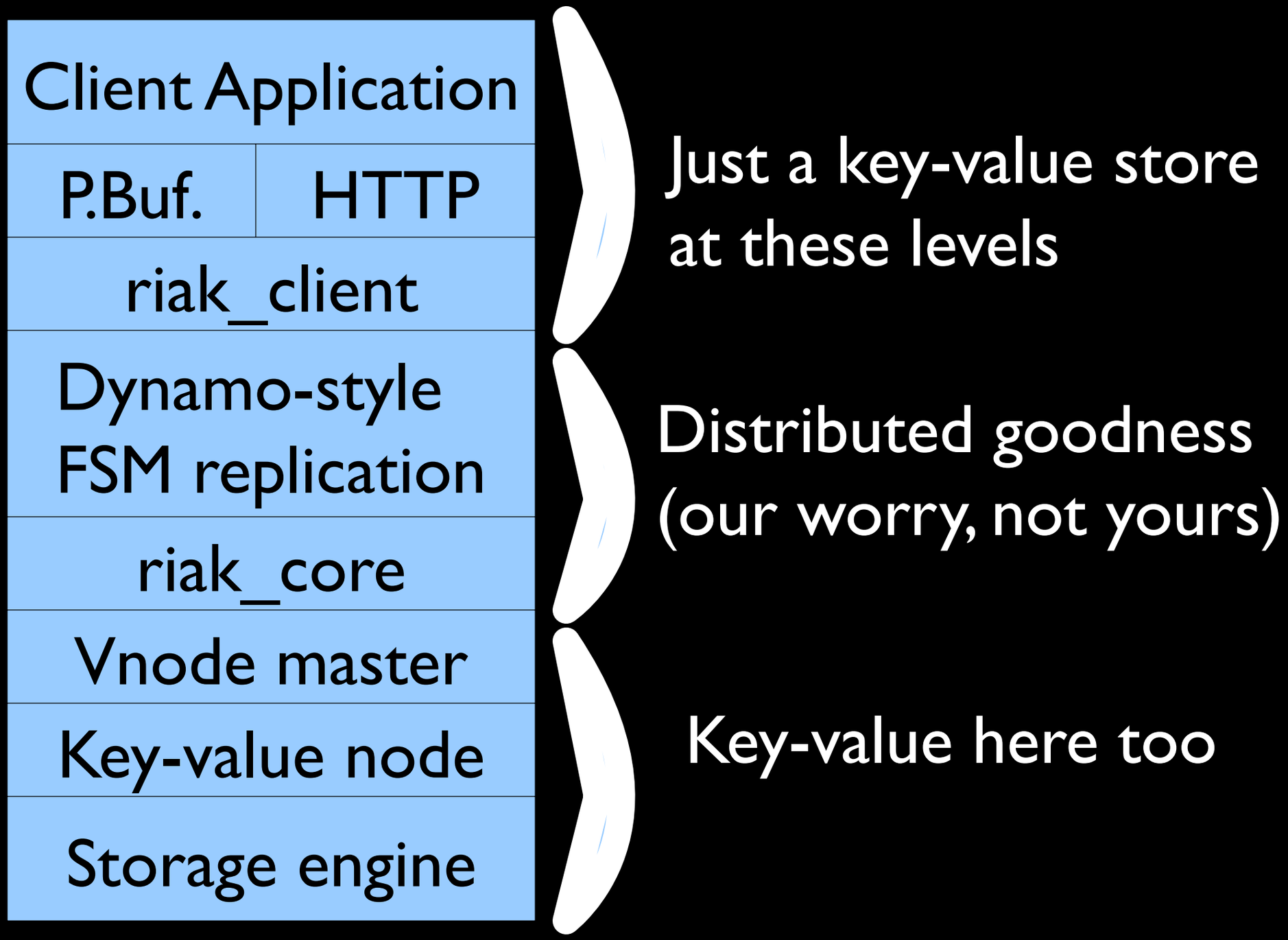
Obligatory Diagram



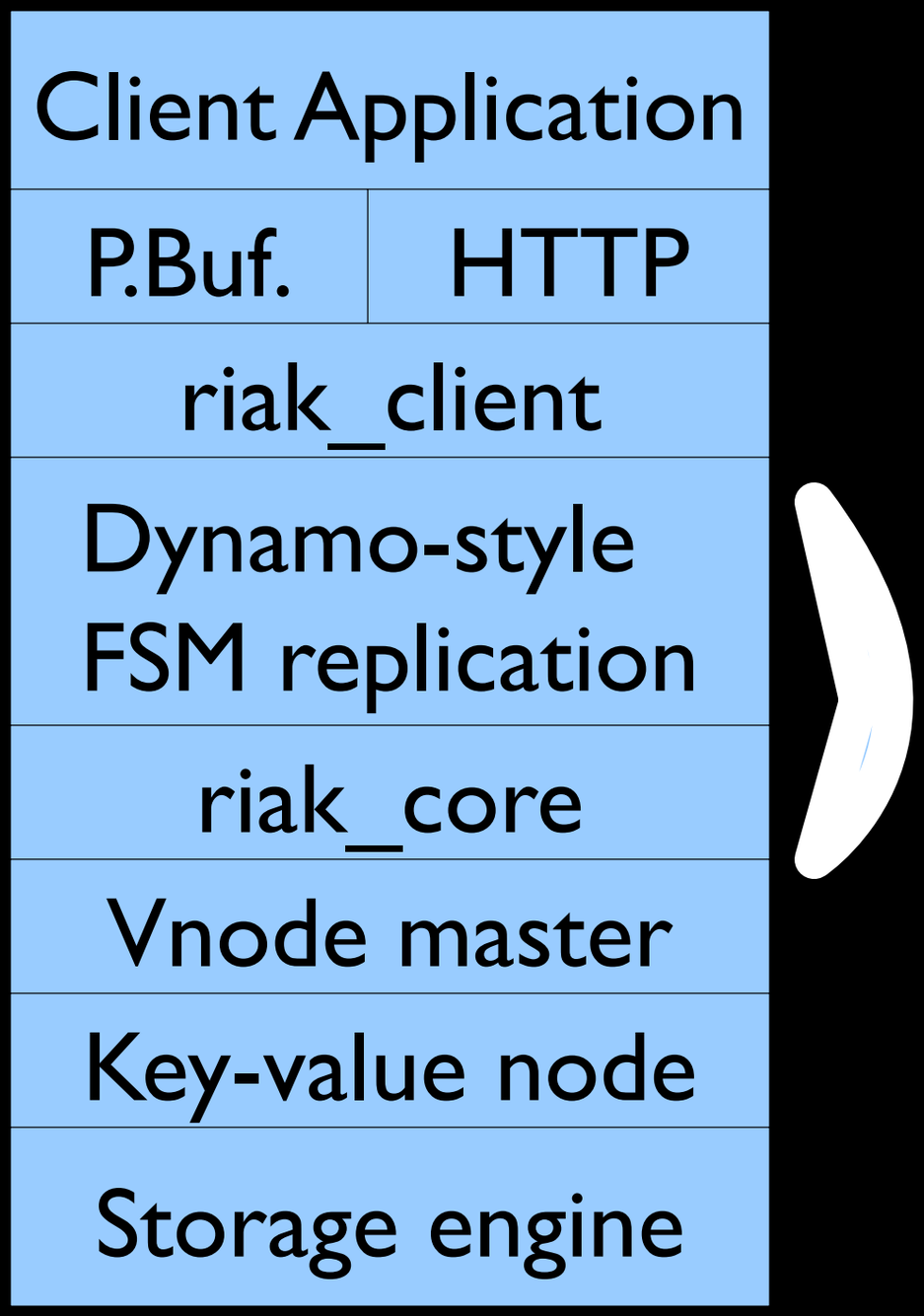
Choose either protocol

* Protocol Buffers: faster
* RESTful HTTP: ubiquitous

Obligatory Diagram



Obligatory Diagram



- Consistent hashing
- Data handoff
- Gossip
- Ring membership
- Node liveness
- Buckets

Hammer, screwdriver, etc. by Basho and by you

K-V Application		Search App.		Big File App.		Your App.
P.Buf.	HTTP	P.Buf.	HTTP	P.Buf.	HTTP	PB/HTTP
riak_client		search_client		Luwak app		Your code
Dynamo-style FSM replication				riak_client		riak_client
riak_core						←
Vnode master						←
Key-value node						Your code
Storage engine	Merge Index engine					Your code

Use Cases

- **Document storage**
- **Distributed file storage**
  (via Luwak bundled app)
- **Session storage**
- **Distributed cache**
- **Browser-only/Mobile Apps** (yakriak)
- **Full-text indexing & search** (via Solr-compatible bundled app)

Luwak: GB-sized files

- https://wiki.basho.com/display/RIAK/Luwak
- PUT new file (with/without name)
- GET file (with range request)
- DELETE file
- Example:
  - `curl -i -d 'This is a 1-line file' -H "Content-Type: text/plain" http://127.0.0.1:8098/luwak/myfile.txt`

Luwak: bits still wet

- Luwak is new software, still drying...
  - Apache2 license, go hack it!
- File pieces stored in Merkle tree
  - Luwak piece = Riak key & value
  - (optional) File metadata headers
- Riak practical limit: 10MBytes
- Luwak practical limit: 250GBytes (?)

Riak Search

- **Still beta**, but mostly usable
- Goal: Lucene & Solr compatibility with seamless distribution and replication.
- Automatically index all values stored in a Riak k-v bucket.
- And/or index file F, or all files in a dir.
- https://wiki.basho.com/display/RIAK/Riak+Search

Riak Search

- Analyzers in Erlang or Java
  - Stemming, stop words, …
- Search by term + field, boolean operators, grouping, lexico-graphical range queries, and wildcards (end of a word only)
- Future releases: facets, wider range & wildcard queries, more base data types.

Riak vs. MongoDB

- Check wiki.basho.com for more details.
- Riak Core is distributed app platform
- Riak is content & structure agnostic
- Riak's vector clocks give more flexibility for reconciling conflicting writes.
- Riak nodes all offer same services/roles
- Riak has REST HTTP API option

Getting Started

- Download a package from http://downloads.basho.com
  - Ubuntu/Debian 32/64, RHEL 5, Solaris 10, OpenSolaris, source pkg
- Set the node name, IPs
- Start up the node
- Join a cluster
- Start storing/retrieving values

Riak in Ruby

- Three gems
  - riak-client (basic ops)
  - ripple (ODM)
  - riak-sessions
    (Rack/Rails session stores)
- All HTTP - Protobuffs coming

Scott, Check Your Remaining Time!

Plug for … Support & consulting, enterprise features, EE pricing for startups.

Email info@basho.com or go to http://www.basho.com/contact.html to talk with us.

www.basho.com

require ‘riak’

- Make a client object
  client = Riak::Client.new
- Get a bucket
  bucket = client.bucket(‘foo’) # Riak::Bucket
- Get an object from the bucket
  obj = bucket.get(‘bar’) # Riak::RObject
- Initialize a new object
  obj = bucket.new(‘baz’)

Riak::RObject

- Get/set object key
  obj.key = “bar”
- Get/set content-type
  obj.content_type = ‘application/json’
- Get/set the object body data
  obj.data = {“name” => “Scott, but this presentation is originally by Sean 'Way Awesome' Cribbs”}
- Store the object
  obj.store

More RObject

- Get object's bucket
  obj.bucket
- Delete the object
  obj.delete # freezes the object
- Detect/extract siblings (more later)
  obj.conflict? && obj.siblings
  # Array<RObject>
- Follow/traverse links (more later)
  obj.walk(:tag => "friend")

Riak::Bucket

- Set the replication factor
  ```
  bucket.n_val = 5
  ```
- Set default request quorums
  ```
  bucket.r = 3 # w,dw,rw
    # number or one/all/quorum
  ```
- List keys
  ```
  bucket.keys # pass block to stream
  ```
- Set consistency flag (allow sibling generation)
  ```
  obj.allow_mult = true
  ```

Links: Lightweight Relationships

Link Header

bucket

tag

Link: </riak/demo/test1>; riaktag=”userinfo”

key

Links in Ruby API

- Create a link
  ```
  Riak::Link.new("/riak/bucket/key",
                 "tag")
  ```
- Read an object's links
  ```
  obj.links # Set<Riak::Link>
  ```
- Convert an object to a link
  ```
  obj.links << obj2.to_link("next")
  obj.store
  ```

Link-Walking

- Asks Riak to traverse a sequence of links via special URL
- Filter by bucket/tag
- Can return results from intermediate traversals
- Response is nested multipart/mixed (riak-client handles this for you)

L/W Example

GET /riak/demo/test1/_,friend,1

Start at demo/test1, follow all links tagged “friend” and return the objects

```
obj = client[‘demo’][‘test1’]
obj.walk(:tag => “friend”,:keep => true)
```

L/W Example

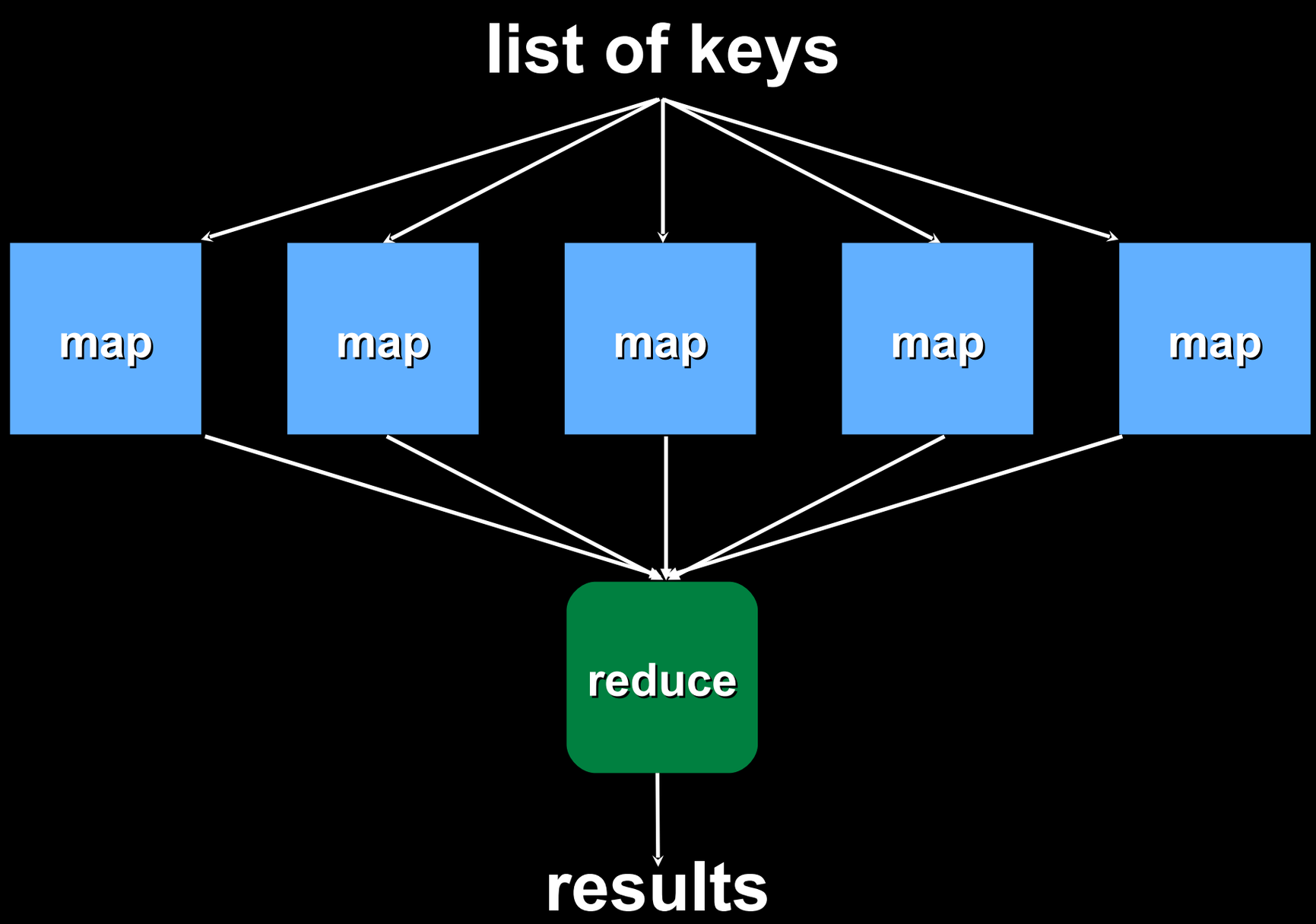
GET /riak/demo/test1/_,_,_/_,_,1

Start at demo/test1, follow all links, follow all links from those, return everything from the last set

```
obj.walk({},{:keep => true})
```

Fault-tolerance

Map-Reduce



Map-Reduce

Map
in Javascript

```
function(value, keyData, arg) {
    var object = Riak.mapValuesJson(value)[0];
    for (field in arg) {
        if (object[field] != arg[field]) {
            return [];
        }
    }
    return [object];
}
```

```
{
 bucket:"foo",
 key:"bar",
 vclock:"...",
 values:[
  {metadata:{...},
   data:"...."}
 ]
}
```

Reduce

in Javascript

```
function(values,arg) {
    return values.sort(function(a,b){
      return a[arg] - b[arg];
    });
}
```

Reduce potentially called multiple times!!

Example Query

```
{“inputs”: “goog”,
 “query”: [{“map”:{“language”:”javascript”,
                   “name”: “App.findHighGreater”,
                   “arg”: 600.0,
                   “keep”: false},
          {“reduce”:{“language”:”javascript”,
                     “name”: “Riak.reduceMax”,
                     “keep”: true}]}
```

```
Riak::MapReduce.new(c).add(‘goog’).
        map(‘App.findHighGreater’, :arg => 600.0).
        reduce(“Riak.reduceMax”, :keep => true).run
```

Built-in Functions

- Riak.mapValues
- Riak.mapValuesJson
- Riak.mapByFields
- Riak.reduceSum
- Riak.reduceSort
- Riak.reduceMin/reduceMax

Ripple - ODM

Document Models

```
class Person
  include Ripple::Document

  property :name, String, :presence => true
  many :addresses
  many :friends, :class => Person
end

class Address
  include Ripple::EmbeddedDocument

  property :street, String
end
```

Rails Setup

```
# Gemfile
gem 'curb'      # Faster HTTP
gem 'yajl-ruby' # Faster JSON
gem 'ripple'
```

```
$ gem install curb yajl-ruby ripple
```

Rails Setup (cont.)

```
# config/ripple.yml
development:
  host: 127.0.0.1
  port: 8098

# config/application.rb
require 'ripple/railtie'
```

Ripple Roadmap

- Testing server (edge on Github)
- Protocol Buffers support
- Streaming MapReduce
- Ripple-specific built-ins
- Better ActionView support (form_for)
- Better JRuby support

Happy Fun

Demo Time

Cluster Demo

Yakriak

- Pure HTML/CSS/JS polling chat
- Stored in Riak, no special abstraction (unlike CouchApp)
- On Github:

```
$ git clone \
  git://github.com/seancribbs/yakriak.git
$ ./load.sh
$ open http://localhost:8098/riak/yak/index.html
```

Yakriak Demo

Plug

Support & consulting? Enterprise features, EE pricing for startups?

Email info@basho.com or go to http://www.basho.com/contact.html to talk with us.

www.basho.com

Questions