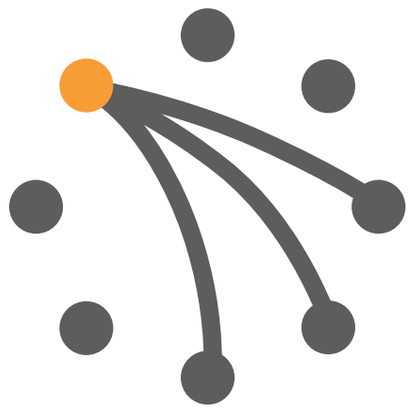
Riak and Games

$ whoami


rigk



riak

Enterprise

riakcs

Distributed, incrementally scalable, masterless, highly-available key/value store.

Distributed, **incrementally scalable, masterless, highly-available key/value store.**

Distributed,

incrementally scalable,

masterless, highly-available key/value store.

Distributed, incrementally scalable, masterless, highly-available key/value store.

Distributed, incrementally scalable, masterless, highly-available key/value store.

Distributed, incrementally scalable, masterless, highly-available key/value store.

Why should games use Riak?

Massive

Concurrency

Predictable Performance

Ease of Operations

Death to downtime

Scaling up (or down)

Global Availability

Handles Edge Cases

“The network is reliable”

Peter Deutsch
The Eight Fallacies of Distributed Computing, #1

Coming soon in 2.0

Data types

Strong consistency

Full-text search

Security

Data types

Strong consistency

Full-text search

Security

Simplified configuration

{ Counters
Sets
Maps

{ Registers
Flags

{

Counters
Sets
Maps

{

Registers
Flags

Data can be divergent, but will eventually converge to the same result as if you had applied the operations in a serial order.

No more dealing with siblings. No more conflict resolution functions.

Data types

Strong consistency

Full-text search

Security

Simplified configuration

What problems does strong consistency solve?

Recency

Partial write failure

Atomicity

(a,1)

(a,1)

(a,1)

+1

(a,2)

(a,1),(b,3)

+3

=2

=4

Recency

No partial write failure

Atomicity

Recency

No partial write failure

Atomicity

```
Client A: GET /balance/saxton => $500
Client B: GET /balance/saxton => $500
Client A: PUT $550 into /balance/saxton => ok
Client B: PUT $600 into /balance/saxton => error
```

```
Client A: PUT /locks/txn123 => ok
Client B: PUT /locks/txn123 => error
Client C: PUT /locks/txn123 => error
Client A: DELETE /locks/txn123 => ok
Client B: PUT /locks/txn123 => ok
```

Data types

Strong consistency

Full-text search

Security

Simplified configuration


Apache
Solr

Write it like Riak.

Query it like Solr.

```
$ curl -X PUT 'http://localhost:10018/types/my_type/buckets/
my_bucket/keys/name' -d “Saxton Hale" -H 'content-type: text/plain'
```

Write it like Riak.

Query it like Solr.

```
$ curl 'http://localhost:10018/search/my_index?q=text:Hale'
```

Facets

Language support (32+)

Geospatial

Sorting
Highlighting
Statistics

Riak's Active Anti-Entropy (AAE) feature helps stale Solr indexes heal themselves.

Data types

Strong consistency

Full-text search

Security

Simplified configuration

Encryption

Authentication

Authorization

SSL / TLS

Encryption

Authentication

Authorization

```
$ riak-admin security add-user saxton
$ riak-admin security add-source all 127.0.0.1/32 trust
$ riak-admin security add-source saxton 10.0.0.0/24 password
$ riak-admin security add-source all 0.0.0.0/0 pam service=login
$
$ riak-admin security add-user dev
$ riak-admin security add-user ops
$ riak-admin security add-user saxton roles=dev,ops
```

Encryption

Authentication

Authorization

```
$ riak-admin security grant riak_kv.get ON default bucket TO saxton
$ riak-admin security revoke riak_kv.put ON ANY TO saxton
$ riak-admin security grant riak_kv.delete ON ANY to saxton
```

Data types

Strong consistency

Full-text search

Security

Simplified configuration

```
## The enabled Yokozuna set this 'on'.
yokozuna = off

## The port number which Solr binds to.
yokozuna.solr_port = 8093

## The port number which Solr JMX binds to.
yokozuna.solr_jmx_port = 8985

## The data under which to store all Yokozuna related data.
## Including the Solr index data.
yokozuna.data_dir = ./data/yz

--- snip --- snip --- snip ---

## Name of the riak node
nodename = riak@127.0.0.1

## Cookie for distributed node communication.  All nodes in the same cluster
## should use the same cookie or they will not be able to communicate.
distributed_cookie = riak

erlang.async_threads = 64
```

```
$ cat riak.conf | grep anti_entropy
```

```
$ cat riak.conf | grep anti_entropy
anti_entropy = on
anti_entropy.build_limit.number = 1
anti_entropy.build_limit.per_timespan = 1h
anti_entropy.expire = 1w
anti_entropy.concurrency = 2
anti_entropy.tick = 15s
anti_entropy.data_dir = ./data/anti_entropy
anti_entropy.write_buffer_size = 4MB
anti_entropy.max_open_files = 20
$
```

Who else uses Riak?

FLYCLOPS

Questions?

Booth: 2330

E-mail: seth@basho.com

Twitter: @saxton

Web: http://docs.basho.com