

Getting Started with Couchbase Ruby

Sergey Avseyev

sergey@couchbase.com



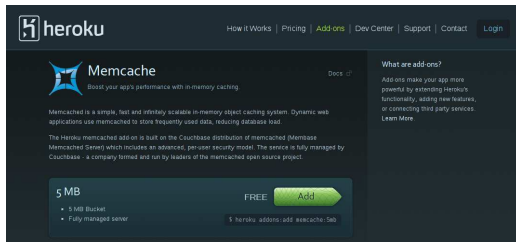
COUCHBASE

Couchbase Server

What Is Couchbase Server

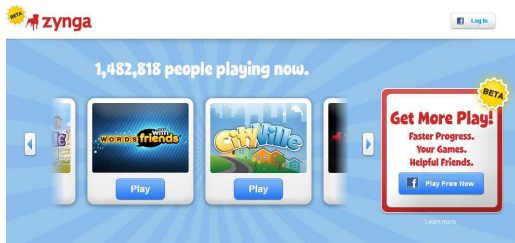
- NoSQL database solution
- No fixed schema
- Automatic key sharding
- Automatic replication
- Low latency optimized
- No multi-operation transaction support

Who Uses Couchbase Server: Heroku



- Leading cloud service (PAAS) provider
- Over 1,500,000 hosted applications
- Couchbase Server serving over 11,800 Heroku customers

Who Uses Couchbase Server: Zynga



- Social game leader — FarmVille, Mafia Wars, Empires and Allies, Café World, FishVille
- Over 230 million monthly users
- Couchbase Server is the primary database behind key Zynga properties

Current Version: 1.1.2

Installation

Install stable version of libcouchbase-dev with dependencies using instructions at <http://www.couchbase.com/develop/c/current>

```
$ gem install couchbase
```

```
Fetching: couchbase-1.1.2.gem (100%)
```

```
Building native extensions. This could take a while...
```

```
Successfully installed couchbase-1.1.2
```

```
1 gems installed
```

Connect to the Cluster

Pass connection parameters explicitly

```
conn = Couchbase.connect(:bucket => 'test')
```

Use thread-local connection instance

```
Couchbase.connection_options = :bucket => 'test'  
conn = Couchbase.bucket
```


Simple CRUD

Write key or fail if it *exists already*

```
cas = conn.add('key', 'value')
```

Write key *unconditionally*

```
cas = conn.set('key', 'value')
```

Write key *only if it exists*

```
cas = conn.replace('key', 'value')
```

```
value = conn.get('key')
```

Optimistic Locks

All mutators accepts CAS value (some kind of version or checksum of the key)

```
value, flags, cas = conn.get('key', 'value',  
                             :extended => true)
```

Setting with wrong CAS value will raise Couchbase::Error::KeyExists

```
conn.set('key', 'newvalue', :cas => 1234)  
#=> Couchbase::Error::KeyExists: failed to store value
```

Expiration

Write key and set expiration

```
conn.set('key', 'value', :ttl => 1.minute)
```

Read key and set expiration

```
conn.get('key', :ttl => 1.minute)
```

Only update expiration

```
conn.touch('key', :ttl => 1.minute)
```

Miscellaneous

Multi get

```
foo, bar = conn.get('foo', 'bar')
```

Multi touch

```
conn.touch('foo' => 30.seconds, 'bar' => 10.minutes)
```

Increment/Decrement

```
conn.incr('counter')
```

```
conn.decr('counter')
```

Next Version: 1.2.0.dp5

Installation

Install next version of libcouchbase-dev with dependencies using instructions at <http://www.couchbase.com/develop/c/next>

```
$ gem install couchbase --pre
Fetching: couchbase-1.2.0.dp5.gem (100%)
Building native extensions. This could take a while...
Successfully installed couchbase-1.2.0.dp5
1 gems installed
```

Map/Reduce Analysis

Define **design document** containing single **map** function (using admin panel or `Bucket#save_design_doc`):

```
{
  "_id": "_design/users",
  "views": {
    "all_by_email": {
      "map": "function(doc) {
                if (doc.type == 'User')
                  emit(doc.email, null);
              }"
    },
  }
}
```

Executing views

Pick up design document:

```
ddoc = conn.design_docs['users']
```

Iterate over view results

```
ddoc.all_by_email(:include_docs => true).each do |doc|  
  puts doc.key  
  puts doc.doc.inspect  
end
```


Pessimistic Locks

```
value = conn.get('key', :lock => 5.seconds)
```

All subsequent mutators are forced to use corresponding CAS value or fail

Rack Session Storage

```
# config.ru
require 'rack/session/couchbase'
use Rack::Session::Couchbase,
  :expire_after => 5.minutes,
  :couchbase => {:bucket => "sessions"}
```

Rails Session Storage

```
# config/initializers/session_store.rb
require \
  'action_dispatch/middleware/session/couchbase_store'

Name::Application.config.session_store :couchbase_store,
  :expire_after => 5.minutes,
  :couchbase => {:bucket => 'sessions'}
```

Rails Cache Storage

```
# config/application.rb
cache_options = {
  :bucket => 'protected',
  :username => 'protected',
  :password => 'secret',
  :expires_in => 2.hours
}
config.cache_store = :couchbase_store, cache_options
```

Even More :)

ODM (Object Document Model) for Rails

<https://github.com/couchbaselabs/ruby-couchbase-model>

```
$ gem install couchbase-model
```

```
class Post < Couchbase::Model
  attribute :title
  attribute :body
  attribute :created_at, :default => lambda { Time.now }
  view :by_created_at
end
```

Couchbase for EventMachine (experimental)

<https://github.com/couchbaselabs/couchbase-ruby-client-em>

```
$ gem install em-couchbase
```

Thanks

Questions?

Find these slides at:
github.com/avsej/getting-started-with-couchbase-ruby