



Athens Ruby Meetup #11

Nikos Diamantopoulos | @nolamesa
Giorgos Tsiftsis | @giorgostsiftsis

22 June 2012

Rails Internals



Aaron Patterson

Code Charcuterie

- Rails Internals
- Best job in the world :)
- Uses flog, flay or other code metric tool
- But rails is complex ...
- Personal Doubt Meter
- Present an example difficult to follow ...



Yehuda Katz

Why Rails is hard

- Known Knowns (25% of rails)
- Known Unknowns (25%)
- Unknown Unknowns (50%, core devs)
- Example of Known Unknown: **CSRF Protection**
- Comparison with other frameworks about csrf protection
- Example of Unknown Unknown: **Unicode**
- IE "snowman" hack

Concurrency

- One of the hottest topics at Railsberry
- Goal: Trying to get most out of all cores
- State off, Concurrency off: Everything deterministic => compilers super-optimize. No side effects, i/o etc :(
- State off, Concurrency on: Easily split calculations to threads.
- State on, Concurrency on:
- Shared memory with locks: One thread at a time
 - Downsides: Developer has explicit control, a pessimistic approach
- Transactional memory: Atomic functionality (rollbacks). Optimistic with no race conditions
 - Downsides: potential unnecessary retries, transaction overhead, dealing with side effects
- Message passing concurrency (actors). No shared memory, share through communication. No need to synchronize, easy to distribute.
 - Downsides: large messages affect performance, coordination between actors is tricky



Patrick Huesler and Tim Lossen

Concurrency in JRuby

- MRI is not good at concurrency (only I/O)
- JRuby and Rubinius are your best options.
- In JRuby you can achieve:
 - Multithreading: Make use of all the Java concurrent libraries
 - Actors: Celluloid (Ruby) or Akka (Scala Framework)
 - Software Transactional Memory (STM): Make use of Clojure

Architecture

Stop using development mode

- Development mode is what makes us excited with rails
- What's wrong (it encourages)
 - **cowboy** programming
 - **testless** programming
 - **inside out** programming
- Programming before thinking / pressing reload until it looks ok
- Pressing reload doesn't scale
- Definition of working: test pass
- migration => model => controller => template => route => reload
- build layer before building the layer that uses it
- Work outside-in
- Start writing an acceptance test
- Drill down by writing tests (use mocking)
- it's not about **coverage** it's about better **design**
- *book:*
 - <http://www.amazon.com/Growing-Object-Oriented-Software-Guided-Tests/dp/0321503627>

- **SRP**
- Indicators for Multiple Responsibilities (private methods)
- Extract code (faster tests)
- ActiveRecord violates by definition SRP (data & behaviour)
- Solutions for ActiveRecord and SRP
 - Extract behaviour to module and include them (but this is just a "move")
 - Introduce a separate layer of Domain Objects (model => data, domain => behaviour)
 - Avdi Grim proposal for explicitly hiding the behaviour of AR objects
- Is it easier to add a method to an existing class or create a new class with that method?
- **Domain Language Capture**
- Stay in sync with naming
- **DRY** (its not about code it's about knowledge, eg. comments)
- Stay in sync with DRY



Tammer Saleh

SOA AntiPatterns

- Service Oriented Architecture
- In SOA performance is immediately important
- Custom format tools (json, bson, bert etc)
- Cannot write an SOA without understanding the big picture
- => Tower of Babel problem
- Need to be aware of the tax you pay in case of different language, architecture etc
- => Common bottlenecks: Centralized Database
- => Grand Central API
- SOA must be viewed as a single system -tests that exercise the whole
- Heavy mocking + Integration Tests
- Boutique Mocking
- => Lack of caching
- Parallelize requests (celluloid, dcell, jruby, rubinius)
- Inline requests (background jobs)

- Choose best language for the best job
- translation != meaning
- You could not have THE MODEL
- Big 5 Database models (Relational, Document, Graph, Key-Value, Column Family)
- Data Models Distorted by
 - Query Requirements
 - Technical limitations
- Know your options, the problem will define the db
- Postgres, Redis, Riak, MongoDB, CouchDB, Neo4j, HBase
- Speaker Emphasis: **Neo4j**
- *book*: <http://pragprog.com/book/rwdata/seven-databases-in-seven-weeks>



- <https://github.com/futuresimple/representer>
- Object Oriented approach to views.
- Take control of your view layer when you are creating API and use `to_json` or `to_xml`
- Define Representer class and produce serialized output of model
- Very simple to customize output in different situations, test and get sick speeds.
- “Lightning” version of representer raw SQL
- Other approaches:
 - <https://github.com/apotonick/roar>
 - https://github.com/josevalim/active_model_serializers

You hate your codebase and it's your fault

- Testing controllers is hard and slow
- We should test them as normal Ruby classes (without rack, etc)
- Controllers "break" some OO patterns:
 - Before filters work like magic (with except and only)
 - Instance variables are magically shared with views
 - Single Responsibility Pattern
- Focused Controller gem
- Every controller action is a class
 - You can access its internals by methods
 - You can have filters by mixing in modules
 - Unit-testable => 4x faster tests

Community

- <https://github.com/holman/hopper>
- Sinatra app that fetches Ruby projects and analyses them
- 15.698 Projects
- Some Stats:
- 0.5 Swear words per project
- 4.27 define_methods per project
- **most projects are lonely!** (2 contributors)
- 22 followers
- 5 forks
- **3 inactive or ignored forks**
- Soft tabs vs Hard tabs
- 563 ruby code / 63 comment lines / 1.96 branches
- 8 classes / 2 modules / 2 class methods / 28 instance methods
- 47.1 % **without** license / MIT 44%



Xavier Noria

Rails Contributors

- Effort to truly enlist rails contributors
- <http://contributors.rubyonrails.org/>
- Try to enlist contributors from svn days
- Look for comments, duplications or nicknames
- Finally: 1300 contributors
- Some thoughts about large commits Vs small commits



Linda Liukas

1

What every girl needs to know about programming

- Demystify that technology is some black magic where we are crunching some numbers and bits in our heads and to present it more as art where you are creating things
- <http://railsgirls.com/>
- Introduce creative games and simple puzzles
- Make confs and talks around the world
- Use help from mentors



Josh Kalderimis

Funding open source

2

- Again some blah blah about Travis CI
- <http://www.kickstarter.com/>
- <http://pledgie.com/>
- SoundCloud announcement

Various



Ben Orenstein

Write code faster: expert-level vim

1

- Avoid of extra typing
- Map CTRL + S to :w
- Make a small tmp cheat sheet of commands you are learning
- Grow your vimrc over time
- Learn vim script – it's not that hard
- Don't open file by hand (command-t & rails.vim FTW)
- Stop using arrows
- <https://github.com/r00k/dotfiles/blob/master/vimrc> (his vimrc)
- *books:*
 - <http://learnvimscriptthehardway.stevelosh.com/>
 - <http://pragprog.com/book/dnvim/practical-vim>



Corey Donohoe

Enterprise Me

1

- GitHub Enterprise
- GitHub firewall (jRuby implementation to protect code)
- Hell to maintain
- Branches just kept diverging
- Huge problems with their test suite on jRuby
- Killed jRuby branch and switched back to MRI
- Use proprietary software that obfuscates Ruby code
- Combined with virtual machines and
- Some package distribution system
- They are now very happy with their enterprise release!

Boring ...



Sebastian Deutsch

All about CoffeeScript

- Better go there: <http://coffeescript.org/>



Elliott Kember

Rails and friends

- Backbone
- MVC Javascript Framework
- JQuery
- Underscore.js
- It's quick to build nice things well
- Again better go there: <http://backbonejs.org/>



Konstantin Haase

Message in a bottle

1

- A presentation of how they pre-tested pulled requests in Travis CI

Thank you!