Como Turbinamos nossa suíte de testes Rails em mais de 600%

TDC Florianópolis 2018 - Trilha Ruby

$> whoami

Eric Magalhães

DevOps Evangelist @ JobScore Inc

Joinville - SC - Brazil

Tecnólogo em Gestão de TI

SysAdmin que aprendeu a programar

Nascido em Brasília mas feito na Bahia,
Louco por automação, músico do buteco da esquina, marceneiro de fim de semana.

```
$> su thiago
$> whoami
```

Thiago Pradi

Lider Técnico @ JobScore Inc
Jaraguá do Sul - SC

Bacharel em Ciência da Computação - FURB
Mestre em Informática - UFPR
Desenvolvedor Rails desde 2008
Mantenedor - RubyGem Octopus

Agenda

- Problema
- A solução
- Conclusão

- Background
- Desafios
- Setups antigos

Agenda

- Problema
- A solução
- Conclusão

- Tentativa e erro
- Testes em paralelo
- Implementação
- Melhoria

Agenda

- Problema
- A solução
- Conclusão
- Considerações finais

Um Pouco de História

Problema

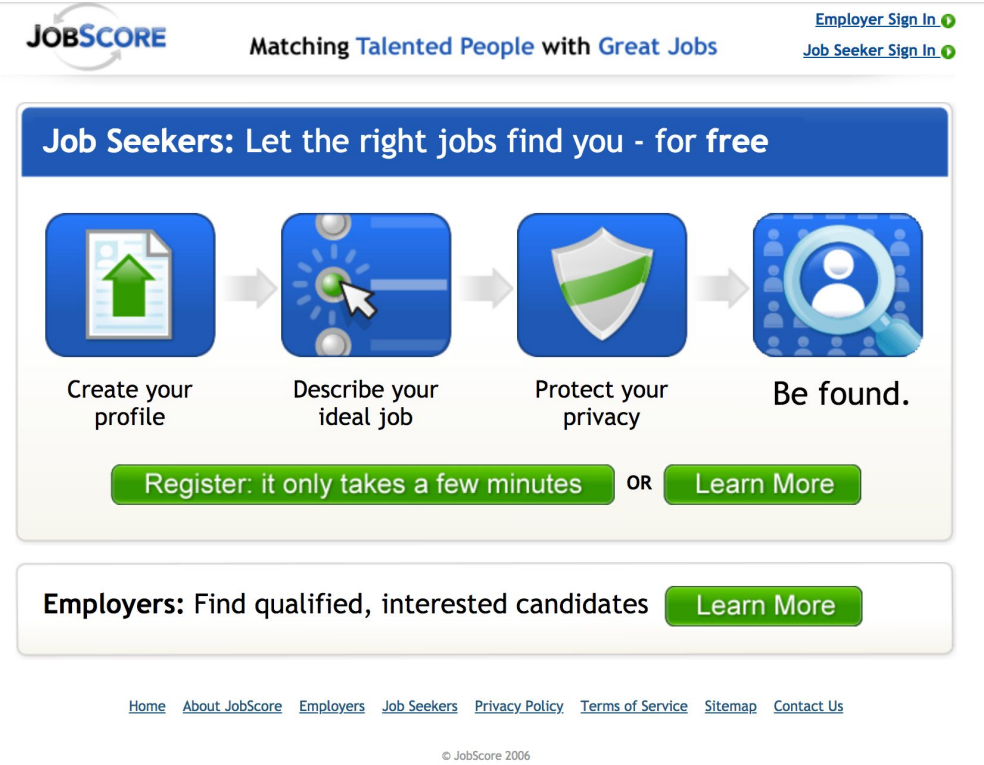
JobScore (www.jobscore.com)

“Software de Recrutamento” (ATS)
Criada em 2005 utilizando Rails 0.5

Problema

```
commit ebf9181466e34d5f35ace05e3f983b436b08bb5b
Author: asoto <asoto@7cfd7551-89f0-0310-b4ca-9f776ed0d428>
Date:   Wed Oct 19 05:40:40 2005 +0000

    Initial checkin of prototype


    git-svn-id: https://jobscore.svn.cvsdude.com/jsn/itn/trunk/prototype@336 7cfd7551-89f0-0310-b4ca-9f776ed0d428

commit 9a788d6ea38b6a314760f3288ac25ea8f1f19a10
Author: asoto <asoto@7cfd7551-89f0-0310-b4ca-9f776ed0d428>
Date:   Thu Oct 20 08:15:29 2005 +0000

    removed log files that should have not been checked in to begin with


    git-svn-id: https://jobscore.svn.cvsdude.com/jsn/itn/trunk/prototype@337 7cfd7551-89f0-0310-b4ca-9f776ed0d428

commit 1ffeb059ac61d4d02928122d4436711df6ab7056
Author: asoto <asoto@7cfd7551-89f0-0310-b4ca-9f776ed0d428>
Date:   Thu Oct 20 08:18:46 2005 +0000

    set svn:ignore prop to ignore *.log files
```

Problema


JOBSCORE
Matching Talented People with Great Jobs
Employer Sign In
Job Seeker Sign In
Job Seekers: Let the right jobs find you - for free
Create your profile
Describe your ideal job
Protect your privacy
Be found.
Register: it only takes a few minutes
OR
Learn More
Employers: Find qualified, interested candidates
Learn More
Home
About JobScore
Employers
Job Seekers
Privacy Policy
Terms of Service
Sitemap
Contact Us
© JobScore 2006

Problema

13 anos de história em números

Profitable ($$$)
Mais de 15 Bilhões em aquisições
Mais de 1M de aplicações

Stack Atual

Ruby on Rails / Python
PostgreSQL / MySQL / Redis
Passenger / AWS

Problema

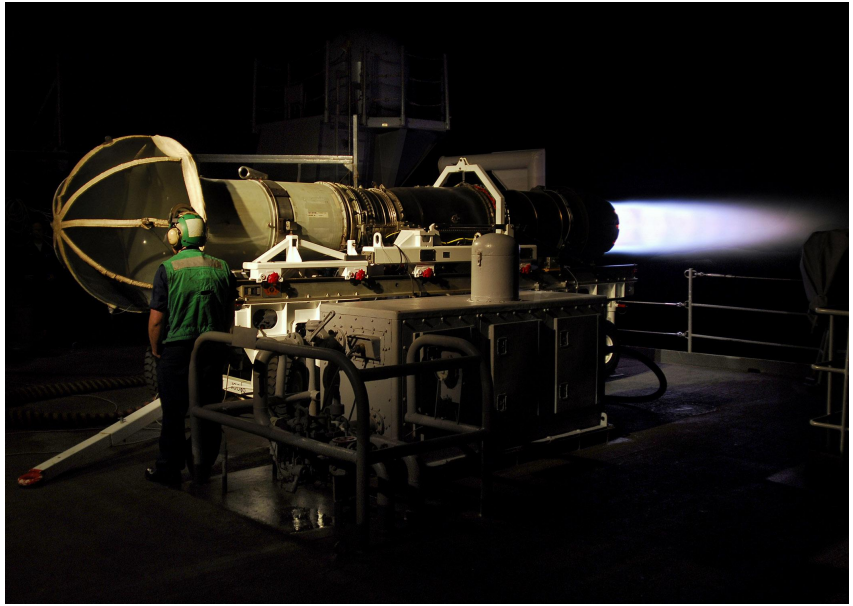
Antes de 2012/2013...

Problema

Decisões Antigas de Engenharia (exemplo: Fat Controllers)

Problema

Criação de dados para testes utilizando somente Fixtures



Problema

Ausência de testes de Aceitação

Utilização de testes de Integração

Problema

```
require 'test_helper'¬
¬
class AssumeJobseekerTest < ActionDispatch::IntegrationTest¬
  before do¬
    @ats_user = ats_users(:system_admin_ats_user)¬
    post '/login', email: @ats_user.primary_email, password: 'pass'
  end¬
¬
  test 'assume_jobseeker' do¬
    tom = candidates(:tom_job_seeker_candidate)¬
    get "/admin/accounts/assume_jobseeker/#{tom.id}"¬
    follow_redirect!¬
    assert_redirected_to '/job_seeker/home'¬
  end¬
end¬
```

Problema

Coverage abaixo da média (< 50%)

Problema

Abuso de soluções temporárias

(rescue nil, monkeypatches, gems antigas)

Problema

Problema

2013: Adotadas práticas mais modernas na suíte de testes

Problema

Stack da suíte de testes

Minitest (migrado do test/unit)
Aceitação - Capybara / Poltergeist
Dados - Fixtures/Factories

2016/2017:
problemas diferentes

Problema

Testes lentos pelo abuso de Factories

Problema

Testes quebradiços devido ao exagero de testes de aceitação com Capybara / Poltergeist

Problema

Builds levando mais de uma hora.

Solução: Utilização de servidor de integração contínua.

Problema

1) Jenkins - Self-hosted;
2) Semaphore;
3) Circle CI;
4) Travis.

Problema

Testes divididos em 4 “workers” pelo tipo do mesmo (unitário, aceitação, controller e mailers)

Problema

Alto Custo para upgrades.

Espera de quase 1 hora para cada merge.

A Solução

Histórico

Contratado para ser owner de várias coisas, incluindo o CI

Histórico

~~Nenhum~~ Pouco conhecimento sobre testes para Rails

Primeiro objetivo: Testes em menos de 10 minutos

Primeiros erros

Pull Request #9608 Improving CI/CD

Commit 80c3833

#9608: Improving CI/CD

Branch master

Eric Magalhães authored and committed

#18608 errored

Restart build

Ran for 46 min 45 sec

Total time 1 hr 33 min 37 sec

4 months ago

Build Jobs

# 18608.1	no language set	TEST_SUITES="--unit-tests"	22 min 6 sec	
# 18608.2	no language set	TEST_SUITES="--controller-tests...	20 min 37 sec	
# 18608.3	no language set	TEST_SUITES="--acceptance-tes...	35 min 43 sec	
# 18608.4	no language set	TEST_SUITES="--integration-test...	15 min 11 sec	

Manter o paralelismo existente usando Jenkins + EC2

Primeiros erros

2 vCPU / 4GB = 35min

8 vCPU / 16GB = 4x faster

CPU 1
CPU 2
CPU 4
CPU 0
Pad.

You know nothing, Jon Snow.

Primeiros erros

Paralelismo é a solução!

Estratégia: Ambientes isolados em uma máquina maior

Testes com Docker

Sai Jenkins e entra Buildbot

Primeiros erros

Builds independentes

5 of 6 checks passed

●	**continuous-integration/travis-ci/pr** The Travis CI build is in progress	Details
✓	**buildbot/acceptance_tests** Build done.	Details
✓	**buildbot/controller_tests** Build done.	Details
✓	**buildbot/integration_tests** Build done.	Details
✓	**buildbot/unit_tests** Build done.	Details
✓	**codeclimate** All good!	Details

Testes ainda demoravam demais

Entre 5-16min

You know nothing, Jon Snow.

Testes em paralelo

grosser / parallel_tests

Code | Issues 81 | Pull requests 6 | Projects 0 | Wiki

Rails: 2 CPUs = 2x Testing Speed for RSpec, Test::Unit and Cucumber

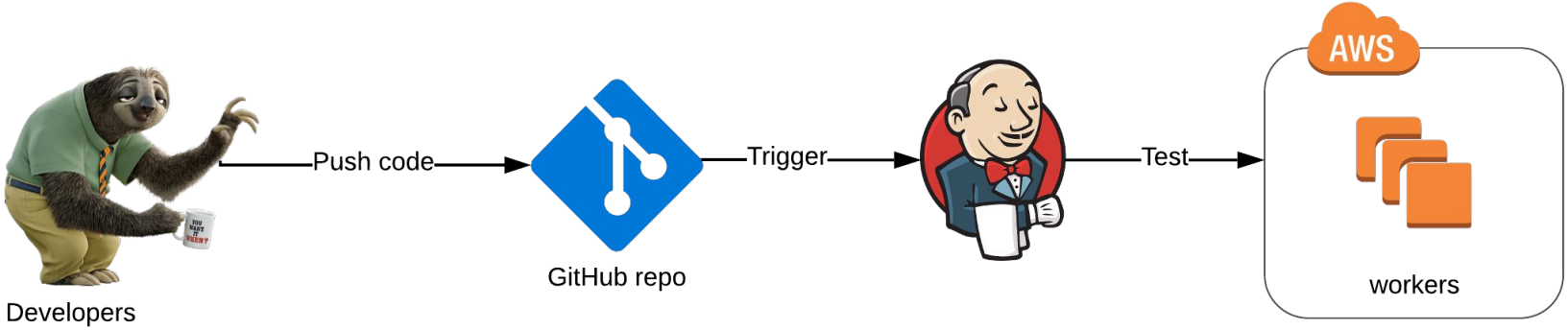
https://github.com/grosser/parallel_tests

Setup relativamente simples

Testes em paralelo

Os tempos eram bons!

Implementação



Testes problemáticos falham com mais frequência

Auto-retry: Minitest reporter + script em Python

Três categorias de teste:

Pass Danger Fail

Usando instâncias spot

Por volta dos USD 300

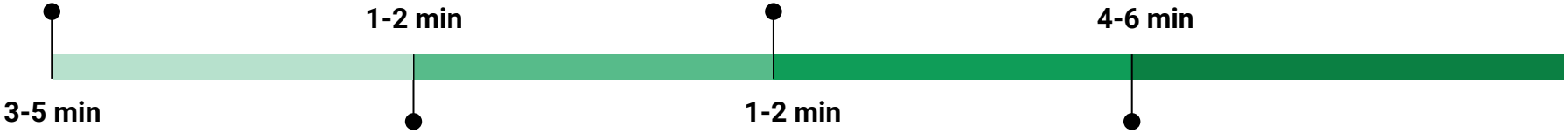
Tempos atuais

Boot

Tempo de boot de uma instância EC2

Preparando ambiente de teste

Criar banco, rodar migrations, testes javascript



Dependências

Tempo para baixar as dependências do código (Gems, NPM, etc.)

Testes

Tempo para rodar todos os testes

Operação / Manutenção

Em torno de 10h/mês

Como monitorar o CI?

Metrics to the rescue!

O CI precisa de carinho

Conclusão

Solução

Aplicações antigas requerem manutenção constante.

Solução

Manutenções incluem refatorar código ou inclusão de novas tecnologias.

Solução

Suas soluções irão refletir no futuro, de forma positiva ou negativa.

Solução

Por isso, saiba medir os prós e contras de cada solução.

Solução

Parallel testing

In Rails 6.0.0, new application will run tests in parallel by default. The number of parallel workers is customizable, and which one will have its own temporary database. I recommend reading the awesome pull request description. Good job!

https://weblog.rubyonrails.org/2018/2/17/this-week-in-rails-rails-5-1-5-parallel-testing-and-more/

Parallel testing #31900

New issue

Merged **eileencodes** merged 1 commit into `rails:master` from `eileencodes:parallel-testing` on Feb 16

Conversation 37 | Commits 1 | Files changed 12 | +415 −7

eileencodes commented on Feb 5 • edited — Owner

@tenderlove and I worked on this which adds parallel testing to Rails applications by default. New applications will have parallel testing enabled by default, and older applications can add it to their test helper:

```
class ActiveSupport::TestCase
  parallelize(workers: 2)
end
```

Parallel testing in this implementation utilizes forking processes over threads. The reason we (tenderlove and eileencodes) chose forking processes over threads is forking will be faster with single

Reviewers

- dogweather
- simi
- tjschuck
- dapicester
- bogdanvlviv

Assignees

- eileencodes

https://github.com/rails/rails/pull/31900

```
$> su ericovis -c 'whereis ${USER}'
```

Social media: @ericovis

Web: https://emagalha.es

Email: eric@emagalha.es

```
$> su thiago -c 'whereis ${USER}'
```

Social media: @thiagopradi

Web: https://www.thiagopradi.com

Email: thiago.pradi@gmail.com

Wrap up