THE DEVELOPER'S CONFERENCE

Aplicações orientadas a eventos com EventMachine

Diego Garcia

DATACOM

linkedin.com/in/**diego-cardoso-garcia**
dcg.cardoso@gmail.com
@diego_cgarcia

- Engenharia de Computação (IPA), Especialização em Engenharia de Software (UFRGS)
- Na Datacom desde 2009. Nestes 9 anos:
  - Desenvolvedor Ruby, Python
  - Scrum Master
  - DevOps

O que é

THE DEVELOPER'S CONFERENCE

EventMachine é uma implementação de alto desempenho do Padrão de Reator

Características

THE DEVELOPER'S CONFERENCE

- Lida com todo material de baixo nível
  - soquetes
  - conexões de rede
  - timers
  - simultaneidade
- leva em consideração **C10K problem**

Ele é bom para

- Servidores event-driven escaláveis. Exemplo: Thin ou Goliath.
- Clientes assíncronos escaláveis para vários protocolos, APIs RESTful e etc. Exemplo: HTTP
- Event logs e ferramentas de monitoramento de redes. Exemplos: eventmachine-tail e logstash.

Reactor

THE DEVELOPER'S CONFERENCE

O padrão de projeto reator é um padrão de programação concorrente para

solicitações de serviço entregues simultaneamente a um manipulador de serviços.

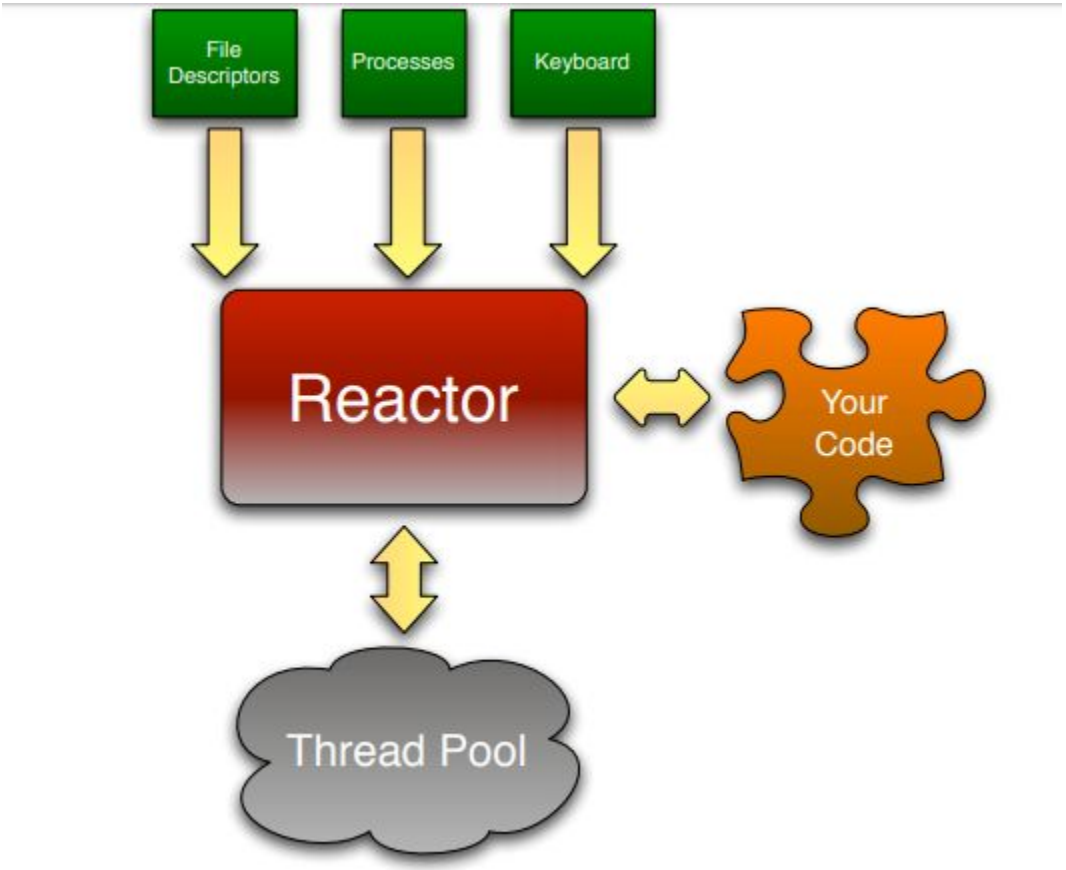
Reactor

THE DEVELOPER'S CONFERENCE

O manipulador de serviços demultiplexa as solicitações de entrada e

despacha-os de forma síncrona para os manipuladores de pedidos associados.

EventMachine



EM::Deferrable

THE DEVELOPER'S CONFERENCE

- Permite associar Callbacks e Errbacks
- É possível definir qualquer número de Callbacks e Errbacks
- Callbacks e Errbacks são executados na ordem que são atribuídos a uma instância de classe

EM::Deferrable

THE DEVELOPER'S CONFERENCE

```
require 'rubygems'
require 'eventmachine'

EM.run do
  df = EM::Protocols::HttpClient.request(:host => 'postrank.com',
       :request => '/')

  df.callback do |response|
    puts "Succeeded: #{response[:status]}"
    EM.stop
  end

  df.errback do |response|
    puts "ERROR: #{response[:status]}"
    EM.stop
  end
end
```

EM::Deferrable

THE DEVELOPER'S CONFERENCE

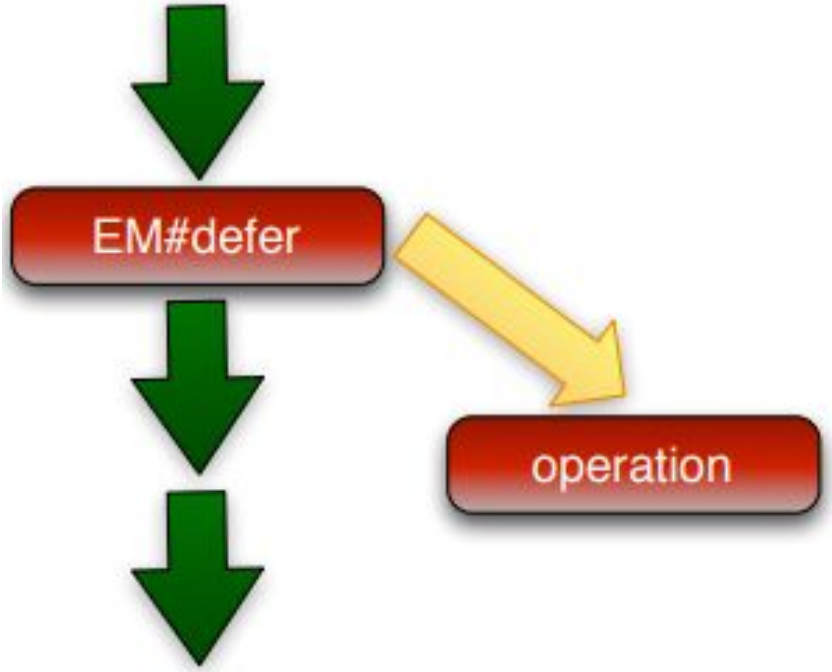
- Permite associar Callbacks e Errbacks
- É possível definir qualquer número de Callbacks e Errbacks
- Callbacks e Errbacks são executados na ordem que são atribuídos a uma instância de classe

EM.defer

THE DEVELOPER'S CONFERENCE

- Usado para operações demoradas
  - Acesso API externa
  - Requisições demoradas
  - Esperas em geral
- Default 20 threads
- Mecanismo para concorrência leve

EM.defer



Começando

THE DEVELOPER'S CONFERENCE

```
require 'eventmachine'

EM.run do
    EM.add_timer(1) { EM.stop }
end
```

Timers

THE DEVELOPER'S CONFERENCE

```
require 'eventmachine'

EM.run do
  EM.add_timer(5) do
    puts "BOOM"
    EM.stop_event_loop
  end

  EM.add_periodic_timer(1) do
    puts "Tick ... "
  end
end
```

Timers

THE DEVELOPER'S CONFERENCE

```
diegogarcia$ ruby timer.rb
Tick...
Tick...
Tick...
Tick...
BOOM
```

THE DEVELOPER'S CONFERENCE

KEEP CALM AND LET'S PLAY A GAME

Obrigado!
Dúvidas?

THE DEVELOPER'S CONFERENCE

linkedin.com/in/**diego-cardoso-garcia**

dcg.cardoso@gmail.com

@diego_cgarcia

Fonte

THE DEVELOPER'S CONFERENCE

https://engineering.universe.com/introduction-to-concurrency-models-with-ruby-part-i-550d0dbb970

https://diogommartins.wordpress.com/2017/04/07/concorrencia-e-paralelismo-threads-multiplos-processos-e-asyncio-parte-1/

https://en.wikipedia.org/wiki/EventMachine

https://zacharyvoase.com/2010/01/02/nice-redis-em/

https://walde.co/2016/11/27/node-js-o-que-e-esse-event-loop-afinal/

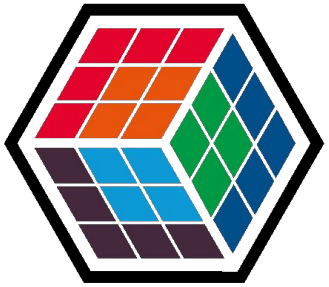
https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_orientada_a_eventos

https://www.ruby-toolbox.com/categories/Concurrent_Processing

https://www.igvita.com/2008/05/27/ruby-eventmachine-the-speed-demon/

https://everburning.com/images/2009/02/eventmachine_presentation.pdf

https://everburning.com/images/2009/02/eventmachine_introduction_10.pdf



THE DEVELOPER'S CONFERENCE