

Why JRuby, Why?!

Brett Giles

YYCRuby Meetup

2014-11

Ruby on Java!

- It *is* just ruby. Write it. Run it.
- but... you *can* call Java.
- and... you *can't* call C.

Why call Java?

- Threads!
- WARs!
- Big conservative companies!
- Swing!

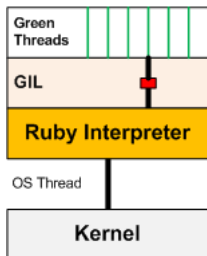
Why *not* call Java?

- Context switching (yours - not the computer's).
- JVM startup is a pain (unTDD-like).
- You really really *really* need the C interface.

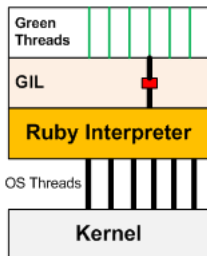
Threads

The dreaded Global Interpreter Lock!

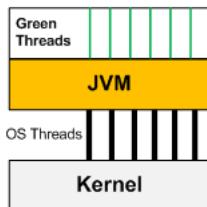
Ruby 1.8.7



Ruby 1.9



JRuby



But... Rubinius does not have a GIL either....

Diagram from <https://www.igvita.com/2008/11/13/concurrency-is-a-myth-in-ruby/>

Thread code - Ruby

```
num_iterations.times do |iter|  
  threads = []  
  t_0 = Time.now  
  (1..num_threads).each do |i|  
    threads << Thread.new(i) do  
      count = 0  
      1_000_000.times { count += 1 }  
    end  
  end  
  threads.each(&:join)  
end
```

Thread code - JRuby - Pg 1

```
class CountMillion
  include Callable
  def call
    count = 0
    1_000_000.times { count += 1 }
  end
end

executor = ThreadPoolExecutor.new(
  num_threads, # core_pool_treads
  num_threads, # max_pool_threads
  60, # keep_alive_time
  TimeUnit::SECONDS,
  LinkedBlockingQueue.new)
```

Thread code - JRuby - Pg 2

```
num_iterations.times do |i|  
  tasks = []  
  
  t_0 = Time.now  
  num_threads.times do  
    task = FutureTask.new(CountMillion.new)  
    executor.execute(task)  
    tasks << task  
  end  
  
  # Wait for all threads to complete  
  tasks.each(&:get)  
end
```