



JRuby and the JVM

Christopher Mark Gore

cgore.com

Monday, May 8, AD 2017

**Ruby is my ~~most second~~ third favorite
programming language of all time.**

1. My own super-awesome programming language, Teepee
(but it's not that awesome just yet)

2. Common Lisp

3. Ruby

4. C

5. Clojure

...

999. Java

JRuby is Ruby, on the JVM.



Ruby on the JVM?

Why would we want Ruby on the JVM?

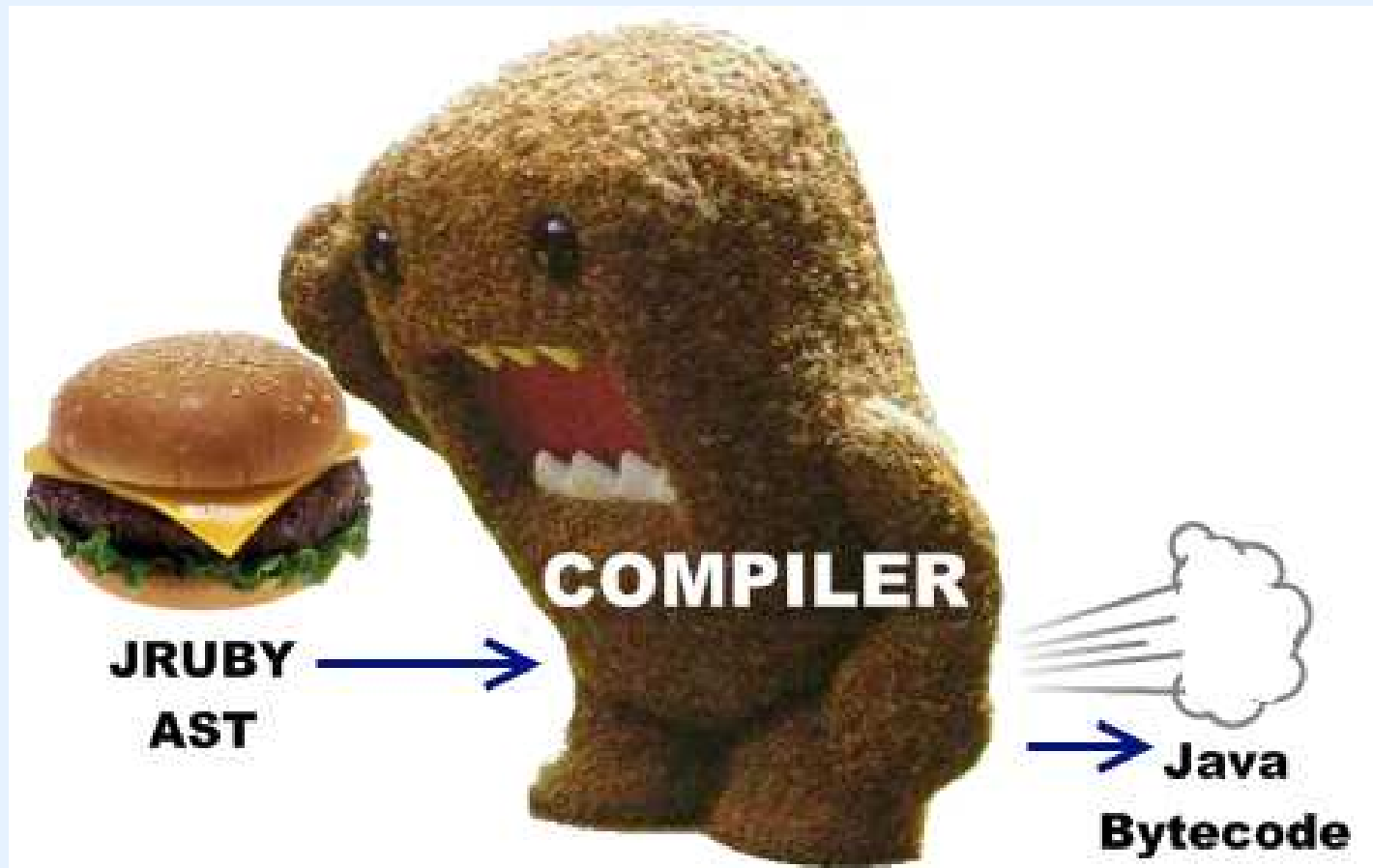


Because there's tons of great libraries for everything.



1

I found this image on the internet that describes very accurately how JRuby works.



Getting Started

1. Download it from `http://jruby.org/download`
...Or `brew install jruby` on OS X
...Or `rvm install jruby` if you use RVM
2. Run jruby from your shell
3. Make code!

CON: The JVM takes forever to start up

```
chris@nephesh ~ $ time jruby -e "puts 'hi'"
```

```
hi
```

```
real 0m1.761s
```

```
user 0m4.800s
```

```
sys 0m0.235s
```

```
chris@nephesh ~ $ time ruby -e "puts 'hi'"
```

```
hi
```

```
real 0m0.595s
```

```
user 0m0.054s
```

```
sys 0m0.050s
```


CON: Until the JIT kicks in it's actually considerably slower than MRI.

CON: Even after the JIT kicks in, it's not really that much faster.

CON: THE JVM WANTS ALL OF YOUR RAM AND IT WANTS IT NOW.

%CPU	%MEM	VSZ	RSS	COMMAND
0.0	0.1	2475044	9300	ruby -e sleep 60

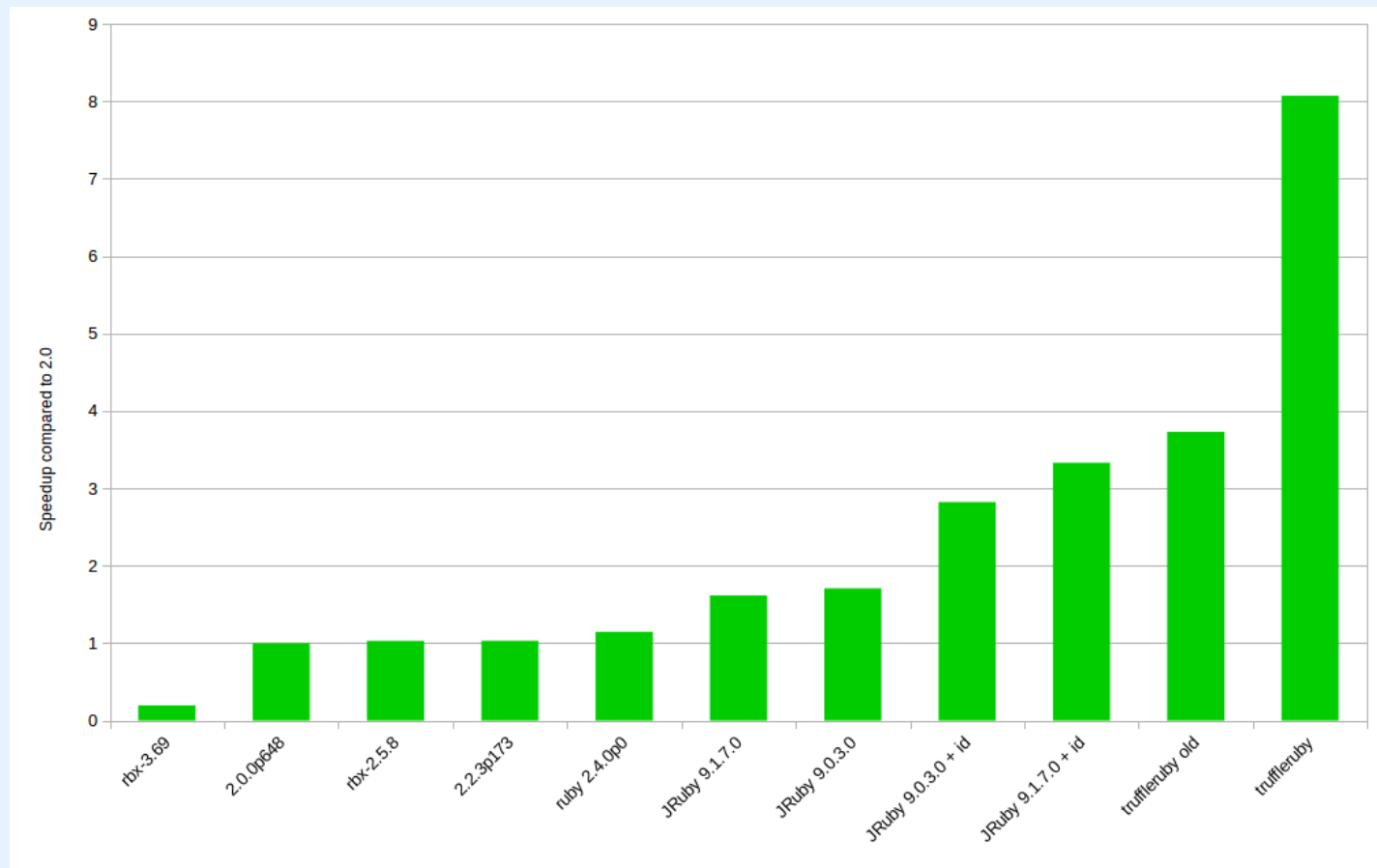
%CPU	%MEM	VSZ	RSS	COMMAND
0.0	1.0	8325372	164520	... org.jruby.Main -e sleep 60

VSZ: virtual memory size, all memory that the process can access, including memory that is swapped out and memory that is from shared libraries.

RSS: resident set size, how much memory is allocated to that process and is in RAM.

No (practical) C extension support, just FFI and Java stuff.

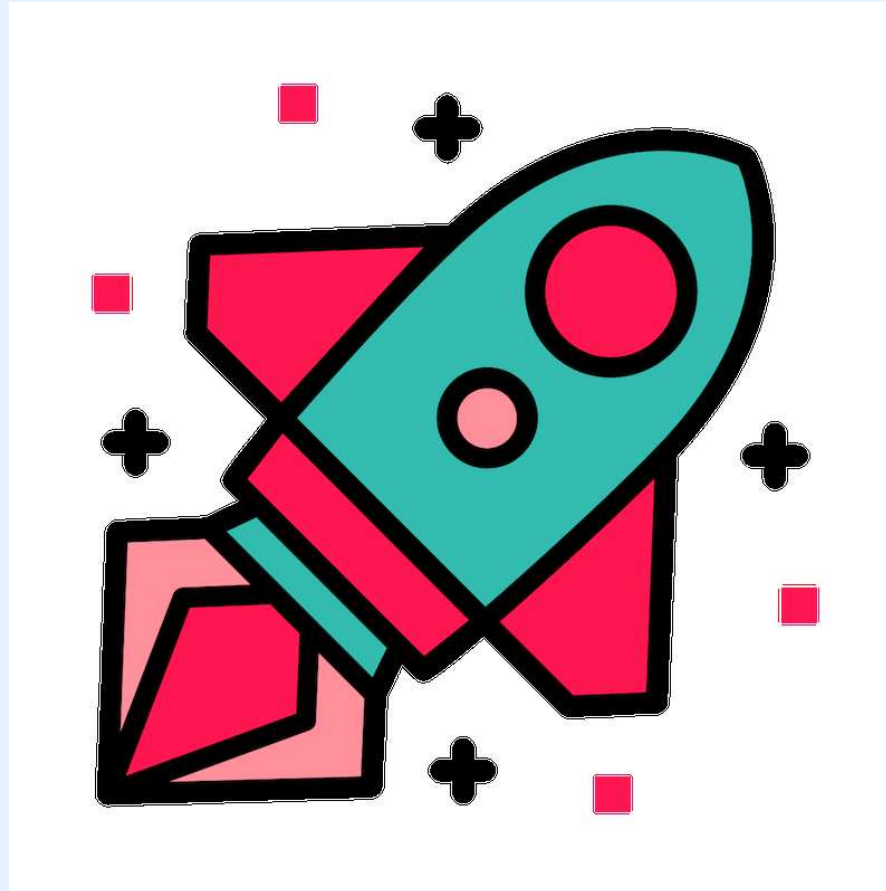
PRO: It's often faster than MRI.



<https://pragtob.wordpress.com>

[/2017/01/24/benchmarking-a-go-ai-in-ruby-cruby-vs-rubinius-vs-jruby-vs-truffle-a-year-later/](https://pragtob.wordpress.com/2017/01/24/benchmarking-a-go-ai-in-ruby-cruby-vs-rubinius-vs-jruby-vs-truffle-a-year-later/)

...Although, nowhere near as fast as TruffleRuby apparently, also on the JVM.



<http://chrisseaton.com/rubytruffle/>

So let's just use TruffleRuby, right?

- No OpenSSL support
- No Nokogiri
- No ActiveRecord device drivers
- Only some of the Rails test suite passes

Maybe in a few years?

Questions?