

JRuby

- What is JRuby?
- Why Java/JVM?
- What can it do?
- What can't it do?
- Getting started

What is JRuby

- JRuby is a 100% pure-Java implementation of the Ruby programming language
- Compatible with MRI 1.8.6
- Nearing compatibility with 1.9.1
- Currently under development funded by Sun

Why Java/JVM?

- Why not?
- Cross platform
- Very mature VM implementation

What can it do?

- Nearly everything MRI Ruby can do – can be used as a replacement for most projects
- Interact with Java classes directly – make use of a very large collection of existing Java libraries and Java's runtime library.
- Allow Java classes to interact directly with Ruby classes – think how you can make your apps enterprisey
- Compile – (good) leverage present and future speed improvements with JIT compilation and (bad) obfuscate code.

What can it do?

- Native threads and thread pooling
- JDBC drivers for database connections – less hassle when connecting to, for example, MSSQL from Unix

What can't it do?

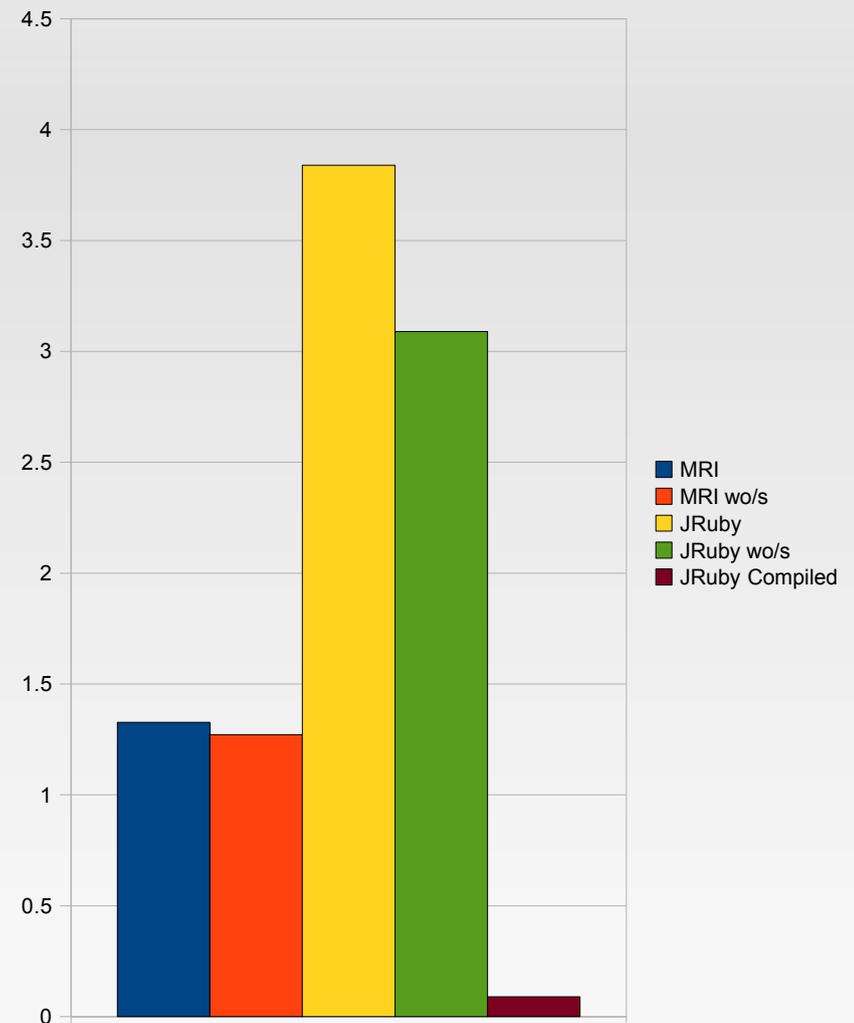
- Start very quickly
- Use C extensions – Many gems have JRuby replacements but some don't so functionality needs rewriting
- Access ObjectSpace

Getting started

- Very easy to get started – all you need is Java(!)
- Linux – `apt-get jruby` || `yum install jruby`
- OS X – `port install jruby`
- Windows – Download JRuby binary
- Or alternatively to the above, Netbeans the Sun IDE platform comes in a ruby edition with JRuby and Rails ready to go.

Performance – Simple benchmark

- Not a real world test
- Fib to 100,000 places
- MRI 1.8.7 = 1.327s
- JRuby w/s = 3.839s
- MRI 1.8.7 wo/s = 1.272s
- JRuby wo/s = 3.086s
- JRuby compiled = 0.092s



Performance – Rails benchmark

- Rails application with JIT
- Fib to 100,000
- Mongrel in dev mode
- MRI: 2840ms average
- JRuby: 4185ms 2936ms 1841ms 1499ms

Performance

Expect JRuby performance to increase greatly
with Java 1.7

Deployment with JRuby

- Why should I care?
- What is Glassfish?
- How do I deploy to it?
- Capistrano



Why should I care?

- Two obstacles to Ruby deployment in "standard" IT environment
- Installing new unapproved software
- Microsoft Windows
- JRuby solves both these problems. Java applications are well supported. Java applications are relatively easy to get running on the windows platform.
- JRuby deployment has **everything** required, including itself

What is Glassfish?

- Why am I talking about these fish? Because it's what we're using.
- Open source application server
- Supports many Java acronyms
- Has a pretty admin interface
- Sun supported
- Approaching the performance of Apache for serving static content

How do I deploy to Glassfish?

- Package Rails app as a war file using warbler gem
- Click deploy my application
- Upload war file
- Application is running
- (curse while you can't figure out how to run rake tasks on production to get the database up and running)