



JRuby: Who What Now

Thomas Enebo

JRuby Guy

Sun Microsystems

Who am I? エネボ . トーマス

- Co-lead of JRuby project
- Longtime Java developer (12+ years)
- Ruby developer (6 years)
- Engineer at Sun Microsystems for >2 years
- Full-time JRuby developer



JRuby is Ruby

- You do not have to be a Java programmer to use it
- Really... "It's just Ruby!" (tm)
- Runs any pure-Ruby script
 - > Rails, Rubygems, Rake, Rspec, ...
- Compatible with Ruby 1.8.6
- OSS (GPL,CPL,LGPL)



JRuby

JRuby Compatibility

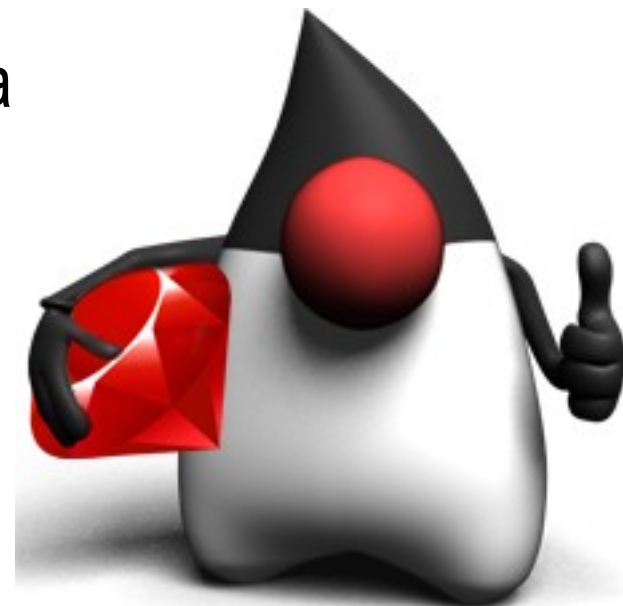
- Lots of Testing
 - > Internal test suite (~22,000 assertions)
 - > Ruby spec tests (~32,000 expectations and growing)
 - > Frequent CI runs against: rails, others...
- Communication with other impls
 - > Ruby-core/email, IRC, face-to-face
- Compatibility with 1.8.6 is “done”
 - > Anyone using 1.8.7?

JRuby Velocity

- JRuby 1.1 RC cycle Winter 2008
- 1.1.5 is current release
 - > 113 bugs fixed in 1.1.5, **1552** since last year
 - > 411 revisions in 1.1.5, **3141** since last year
- Performance is great, always improving
- New Stuff
 - > Ruby 1.9 support
 - > Improve Java interop.

JRuby 1.1.5 Released!

- Released two weeks ago
- Highlights:
 - > Constant cache, Class-level method cache
 - > Many RubySpec fixes
 - > 113 Issues resolved
 - > Reports of 5-20% improvement in Ra



It's not a bug it's a feature?

- Native-threaded
 - > Has not been a large compatibility problem
- No Native “C” Extensions
 - > “Java” Extensions for popular libraries
 - ActiveRecord-JDBC, YAML, RMagick4J, Mongrel,...
 - > Foreign Function Interface to the rescue?
- No Continuations
 - > Does any one use them?

Foreign Function Interface (FFI)

- Call C functions directly from Ruby
 - > Portable across impls, unlike extensions
 - > Most extensions just wrap a library
- Started in Rubinius, then joint API discussions
- JRuby added support in 1.1.4
- JRuby team released Ruby FFI gem
 - > `gem install ffi`

FFI Example

```
require 'ffi'

module POSIX
  extend FFI::Library
  ffi_lib 'c'

  attach_function :getuid, :getuid, [], :uint
  attach_function :getpid, :getpid, [], :uint
end

puts "Process #{POSIX.getpid}, user #{POSIX.getuid}"
```

JRuby 1.9 Support

- Bigger libraries done, more coming soon
- Parser support “landing any day now”
- M17N bits coming along
- 1.8.6 and 1.9.1 support in same binary (--1.9 flag)
- Hasn't taken very long (BasicObject in ~2 hours)
- Done by christmas...

JRuby can also...

- Access Java as if it were Ruby

```
array = java.util.ArrayList.new
```

```
array << "aa" << "b"
```

```
puts array
```

[aa, b]

```
puts array.sort! { |a,b| a.length <=> b.length }
```

[b, aa]

- JRuby scripting makes Java “fun” again

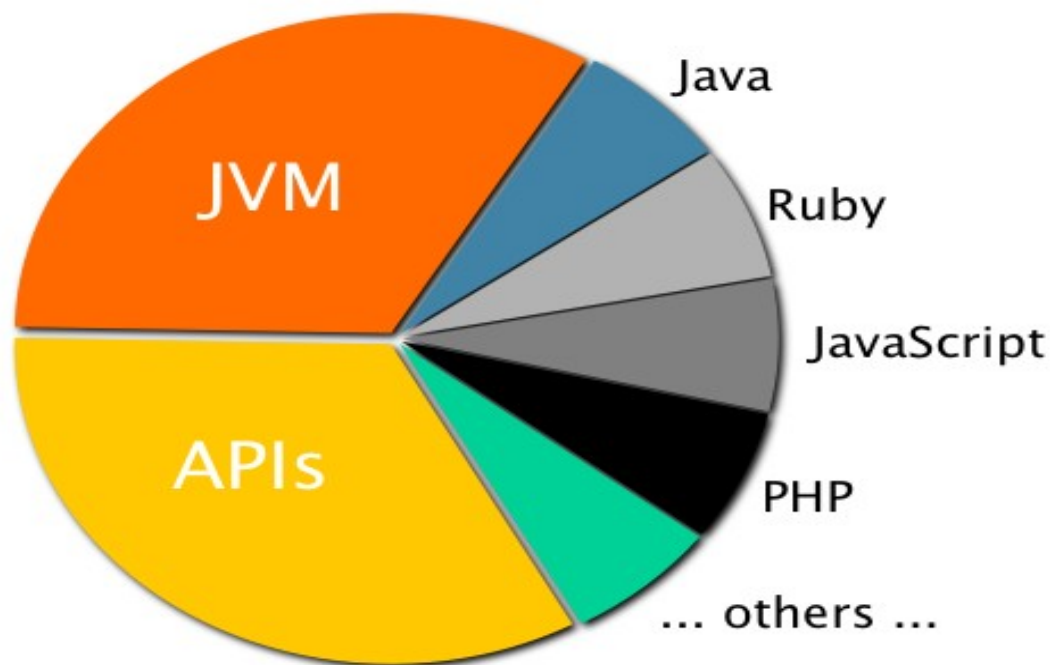
Demo



JMonkeyEngine Java Scripting Demo (3D Scenegraph)

Why Java?!?!!!

- Java Platform is a solid base



Why Java?!?!!!

- Fantastic Virtual Machine
 - > Tuned for over a decade by an army
 - 500 man years so far
 - > Runs on virtually all OS/hardware combos
 - > Dynamic optimizations (Hotspot)
 - > Keeps getting faster:

	Java 5	Java 6	
Rexml	10.9s	7.41s	%32
Hpricot	4.06s	2.59s	%36

The Magic of Hotspot

```
Vector v = new Vector(3); // Thread-safe list
```

```
....
```

```
reset(v); // Called many times
```

```
....
```

```
void reset(Vector v) {  
    for (int i = 0; i < v.size(); i++) {  
        v.set(i) = 0;  
    }  
}
```

Hotspot: Method inlining

- Inline

```
void reset(Vector v) {  
    fast_guard(v) {  
        for (int i = 0; i < lock { arr.length }; i++) {  
            lock { arr[i] = 0; }  
        }  
    }  
}
```

Hotspot: Simple Optz

- Unroll Loop

```
void reset(Vector v) {  
    fast_guard(v) {  
        lock { arr[0] = 0; }  
        lock { arr[1] = 0; }  
        lock { arr[2] = 0; }  
    }  
}
```

Hotspot: Lock Coarsening

- Coarsen Locks

```
void reset(Vector v) {  
    fast_guard(v) {  
        lock {  
            arr[0] = 0;  
            arr[1] = 0;  
            arr[2] = 0;  
        }  
    }  
}
```

Hotspot: ArrayCopyStub

- Contiguous Array replaced by Array Copy Stub

```
void reset(Vector v) {  
    fast_guard(v) {  
        lock {  
            arrayCopyStub(v, [0,0,0])  
        }  
    }  
}
```

Why Java?!?!!!

- Fantastic Garbage Collectors
 - > Tuned over a decade by another army of engineers
 - > Generational (short-lived object collect very quickly)
 - > Compacting (say good-bye to memory fragmentation)
 - > Concurrent
 - > Many tunables and choices

Why Java?!?!!!

- Native Threads
 - > Ground-up architecture around native threads
- Tools
 - > Profilers, Debugging, IDEs (Netbeans)
 - > Monitoring (JMX, DTrace)
- Libraries
 - > Anything you can think of is already a Jar file

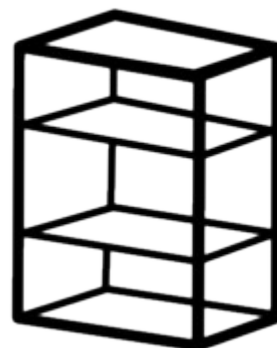
JRuby on ~~Rails~~ Web

JRuby on Rails Web

- Java is Everywhere
 - > Every OS/Hardware platform you can think of
 - > Probably already on a server near you
- Less political resistance
 - > “JRuby is just another Jar file”
 - > No need to install additional software on your servers
- Wider database support
- Decent performance

JRuby on Rails Web (Warbler)

- Java-style deployment to Application Server
- Bundles a Rails/Merb/? application into a WAR file
- Hand WAR file to production staff
- Rails App deployed!



rack
powers web applications

http://wiki.jruby.org/wiki/JRuby_Rack

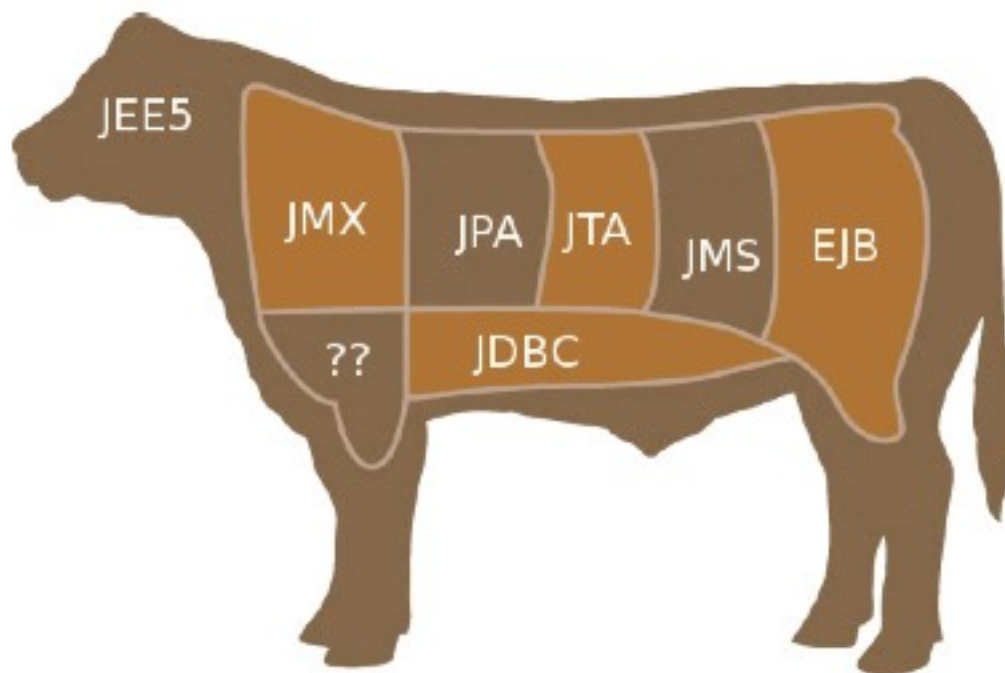
JRuby on Rails Web (GlassFish gem)

- Entire App Server in a 4Mb gem
- Ruby-style deployment to GF Application Server
 - > gem install glassfish
 - > glassfish <Rails/Merb app dir>
- Configurable number of listeners all in one process
- Under active development



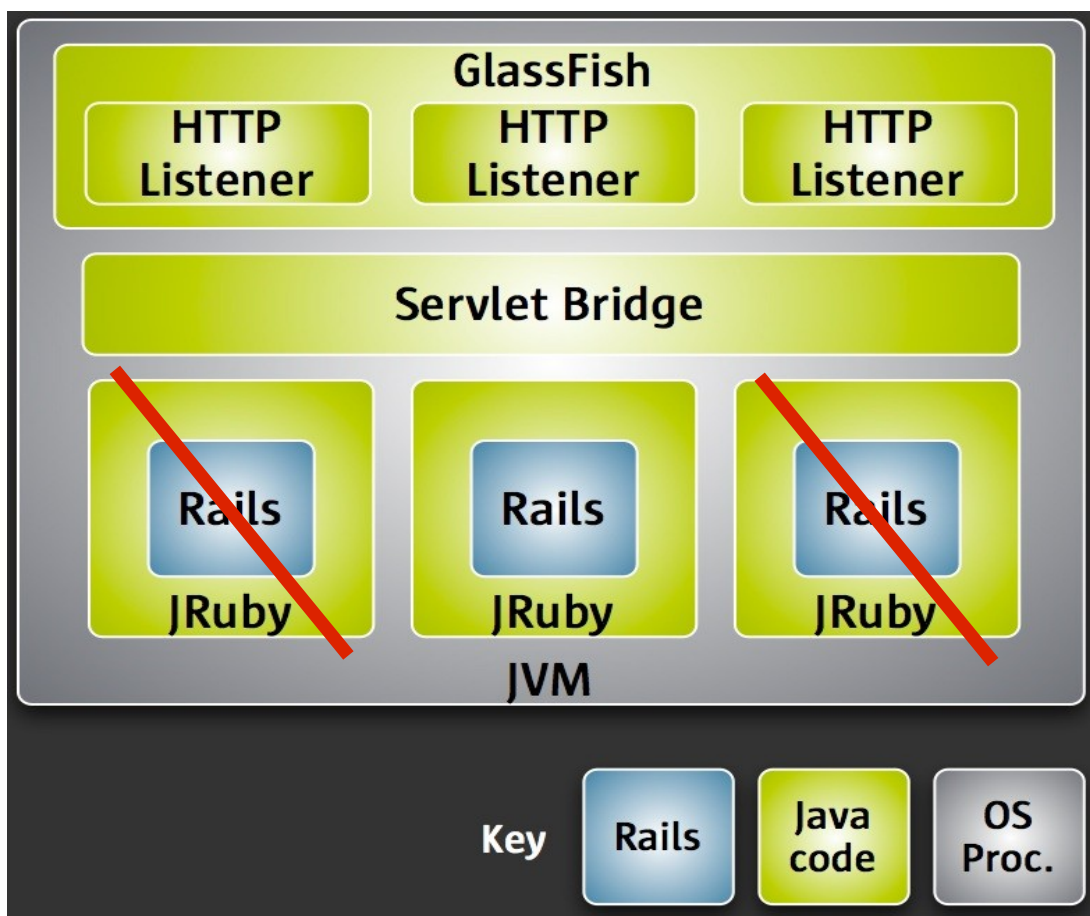
JRuby on Rails Web (Parts is Parts)

- Script Java technology you need into your Rails app
 - > Directly call “Legacy” Java APIs
 - > Get additional choices



Rails 2.2 “Holy Grail” (Thread-Safe?)

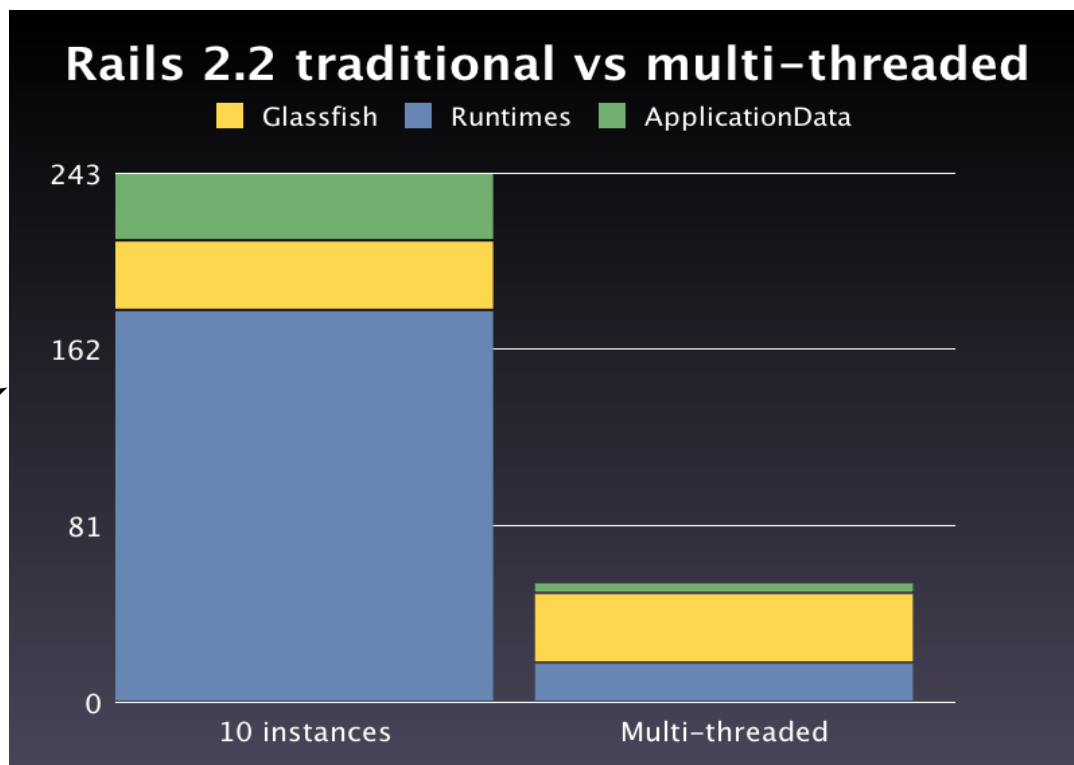
- One Rails instance can satisfy multiple listeners



The “Lame” Test

- Simple 1 controller, 1 model, 1 view app
- Send 1000 reqs with concurrency 10 (ab)
- GlassFish v2

Generated with
JRuby + gruff + RMagick4j



Users Speak

- Customers already have Java App Servers
- Java libraries fill a void in Ruby (Swing)
- Cross Platform (especially Windows)
- Code obfuscation (AOT)
- Odd platform (AS400)
- Speed

Thank you!

- JRuby - www.jruby.org, wiki.jruby.org
 - > http://wiki.jruby.org/wiki/Success_Stories
- GlassFish - glassfish.dev.java.net
 - > `gem install glassfish`
 - > Looking for bug reports, feature requests!
- Tom's blg: www.bloglines.com/blog/ThomasEEnebo
- thomas.enebo@sun.com
- Twitter: [tom_enebo](https://twitter.com/tom_enebo)

I went too fast Demo 1

- Ruby 1.9

I went way too fast Demo 2

- JMonkeyEngine demo

Wow...one more demo!!!!

- `image_voodoo` demo

JRuby and Swing!

- Rubyists are creating many Swing libraries in JRuby
 - > Cheri, Profligacy, Rubeus, Monkeybars, Swiby, Limelight
- Why are they doing this?
 - > No OS library dependency
 - > Cross-platform on all OS's
 - > Swing is a full-featured framework
- Why more than one?
 - > Different styles
 - RAD-based (Matisse GUI builder), Builder/DSL-based, Cleanup Swing syntax in Ruby



Jruby and Swing (Profligacy)

- Twitbucket demo app partial snippet:

```
layout = "[picture|label|filter]"
controls = JEL.new(JPanel, layout) do |c,i|
  c.picture = JButton.new ""
  c.label = JLabel.new "Filter"
  c.filter = JTextField.new 16
end.build
```

