

JRuby

State of the Union



Except where otherwise noted, the content of this presentation is licensed under the Creative Commons Attribution-Share Alike 3.0 United States License (<http://creativecommons.org/licenses/by-sa/3.0/us/>).

Engine Yard

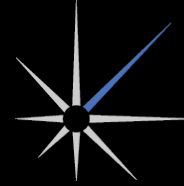


Engine Yard

- Amazing reception
 - JRuby support and cloud offerings
 - JRubyConf
 - Design work
 - Travel sponsorship
 - Moral support!

JRubyConf Sponsors

Gold Sponsors



Silver Sponsors



Who are we?

- Charles Oliver Nutter
- Thomas Enebo
- Nick Sieger
- Ola Bini

Who are you?

- Never tried JRuby?
- Reported JRuby bugs?
- Contributed patches to JRuby?
- Using JRuby in production?
 - JRuby on Rails?

Looking Back

- 2001: Jan Arne Petersen starts JRuby
- 2002: Tom Enebo joins the project
- 2004: Charles Nutter joins
- 2006: Ola Bini, Nick Sieger join
- 2009: Nine active committers

Milestones

- Winter 2006
 - IRB, Rake, RubyGems
- Spring 2006
 - Rails “works” in time for JavaOne
- Fall 2006
 - Sun Microsystems

Milestones

- Spring 2007: JRuby 1.0 released
 - Thoughtworks, Oracle begin using it
- Summer 2007: Compiler work begins
- Winter 2008: JRuby 1.1 released
 - Ruby execution faster than MRI
- Summer 2008: Rails “threadsafe” work

Milestones

- Fall 2008: Start of Java integration rework
- 2009
 - Three major releases
 - 1300+ issues resolved
 - Thousands of commits
 - Engine Yard and JRubyConf!

Lots of Friends

- Sun Microsystems (kenai.com and more)
- Oracle (mix.oracle.com and more)
- Thoughtworks (Mingle, customer apps)
- Collaborative Software Initiative
- LinkedIn (LinkedIn Polls)
- Relevance
- Many more!

We've come a long way!

Where are we now?

JRuby 1.4.0

- Released November 2
- 300+ bug fixes
- 1.8 support based on 1.8.7p174
- Improved (not complete) 1.9 support
- Installer + native .exe for Windows
- Java integration improvements

JRuby 1.4.0

- New embedding API “RedBridge”
- Bug-for-bug port of Syck (YAML)
- More memory reductions
- RubyGems, Rake, RSpec updates
- Focused perf fixes (IO, JI)

Note on Compatibility

- ~37,000 passing rubyspecs
- ~22,000 passing assertions
- Continuous integration
 - On commit: ~3 Java versions * 4 vendors
 - Nightly: 30 minute “full” run

Incompatibilities

- Missing some POSIX behavior (e.g. `fork()`)
- No continuations (`callcc`)
- Slower startup
- Cannot run native C extensions
 - Java-based equivalents (`ar-jdbc`, `yaml`, ...)
 - Foreign Function Interface (FFI)

Community

- 9 active committers
- Dozens of external contributors
- Countless users
 - Tipping point?
- Talks at every Ruby conference
 - ...and most Java conferences
- JRuby is part of Ruby now!

JRuby Everywhere!

- Cloud deployment
- Intranet applications
- Desktop/POS applications
- Graphics and games
- Mobile devices

More Friends!

- Gilt Group
- Oslo Gardermoen Airport (now live!)
- Allen Telescope Array
- PONS.eu
- Datek Wireless AS
- EdgeCase

The Future

- Pain Points
- Geeking Out

Pain Points: Windows

- What were we thinking?
 - Any Java folks on Windows?
 - Feedback on installer

Pain Points: Native Launchers+Installers

- `#!/usr/bin/env jruby --oh --crap`
- macports? .deb 1.1.3?

Pain Points: Horizontal JI

- Annotations, other Java 5 features
- Better type coercion
 - type-casted nulls?
- Improve ruby2java for “real” .classes

Pain Points: Vertical JI

- Easily integrate into Java projects
 - Hook up to existing DB, ORM
 - Integrate into testing, builds, process
 - Mixed language projects

Pain Points: Startup Time

- `stat()` is the root of all evil
 - 113k stats to load 3k files
- Setting up things that are never used
 - Too-eager reflection
 - `setAccessible()` unnecessarily
- Make Nailgun flawless

Pain Points: C Extensions

- What gems are important to you?
- FFI?
- Java-native extensions?
- Another solution?

Pain Points: ActiveRecord-JDBC

- MySQL adapter is not yet perfect
- The rest are even less perfect
- Performance issues?

Geeking Out

- Cool tech coming soon...

Geeking Out: Java 7

- invokedynamic
- dyn-lang friendly JVM settings
- escape analysis

Geeking Out: New IR+Compiler

- “Compiler guy” making next-gen compiler
- Ruby-level optimizations
 - Ruby method-inlining
 - Type-propagation
 - Ruby block-inlining

Geeking Out: Duby?

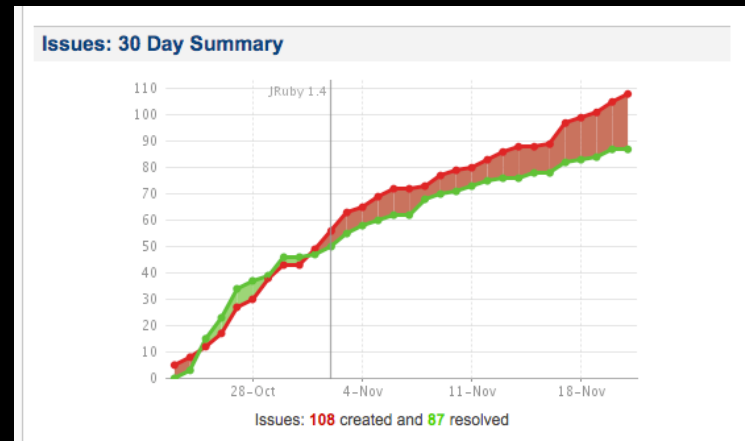
- Java-performance with Ruby-syntax
 - Reduce semantic gap for contributors
 - One-off Java as Duby code
 - Learn ways to improve JRuby

Geeking Out: Library Optz

- [1,2,3].all? PWNAGE
 - Array Enumerable ops 20-30% faster
 - Less GC/memory overhead

HELP!!!

- “HELP!!!” -Yugui



Help!!!:

Today's Heroes

- Vladimir's Back!
 - Great source of issues
 - Greater source of fixes
- Hiro, David, Logan, Wayne, Yoko, Marcin, Jim, James....
- <YOUR NAME HERE>

Evangelism!!!

- Presentations
- JRails tutorials at conferences
- Screencasts
- Blog
- Tweets

Documentation!!!

- RedBridge as model
 - Examples, tutorials, walkthroughs

```
• Core
01. package vanilla;
02.
03. import org.jruby.embed.ScriptingContainer;
04.
05. public class HelloWorld {
06.
07.     private HelloWorld() {
08.         ScriptingContainer container = new ScriptingContainer();
09.         container.runScriptlet("puts \"Hello World!\");
10.     }
11.
12.     public static void main(String[] args) {
13.         new HelloWorld();
14.     }
15. }
```

Feedback!!!

- Test gems, report results
 - Native libraries
 - Multi-threading
 - Test/spec failures
 - Performance

RubySpec!!!

- More specs == more confidence
 - We all can speak Ruby
 - Help define what Ruby is
- Ruby 1.9 support
 - Weak specs == weak support

Hacking!!!

- Help us fix pain points
- Scratch your own itch
- Vertical integration
 - Any Hibernate experts?

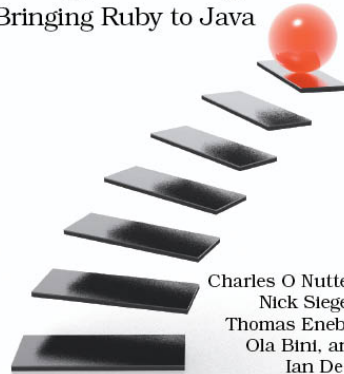
The future looks bright!

Oh, one more thing...

The
Pragmatic
Programmers

Using JRuby

Bringing Ruby to Java



Charles O Nutter,
Nick Sieger,
Thomas Enebo,
Ola Bini, and
Ian Dees

Edited by Jacquelyn Carter

The Facets  of Ruby Series

Enjoy the show!