

JRuby Now and Future

Charles Oliver Nutter

JRuby Guy

Sun Microsystems



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/>).

Agenda

- JRuby now
 - > Swing programming
 - > Tools and IDE support
 - > Graphics and applets
 - > Web applications
- JRuby future
 - > Ruby 1.9
 - > Performance
 - > Da Vinci Machine
 - > Other languages

Who Am I

- Charles Oliver Nutter
- Longtime Java developer (12+ yrs)
- Engineer at Sun Microsystems for 1 yr
- Full-time JRuby developer
- Also working on JVM dynlang support
- Wide range of past experience
 - > C, C++, C#, Python, Delphi, Lisp, Scheme
 - > Java EE and ME, WS

JRuby

- Java implementation of Ruby language
 - > “It's just Ruby!”
- Started in 2002, open source, many contributors
 - > Tom Enebo, Ola Bini, Marcin Mielzynsky, Nick Sieger, Vladimir Sizikov, MenTaLguY
- Aiming for compatibility with current Ruby version
 - > Ruby 1.8.6 patchlevel 111 (114 was just released)
- Improvements on Ruby
 - > Native threading, better performance, many libraries

JRuby 1.1.2 Now

- Maintenance release for 1.1
- Excellent improvements
 - > Drastically reduce raw startup time
 - > YAML symbol parsing >100x faster than 1.1.1
 - > Performance, threading, stack improvements for calls
 - > Several nested backref problems fixed
 - > Bad data race in regexp cache fixed
 - > Many BigDecimal issues fixed
 - > 95 bugs fixed since 1.1.1

Where is JRuby today?

- Tooling for IDEs
 - > JRuby's parser enables NetBeans, Eclipse, IntelliJ
- Swing GUI development
 - > Makes Swing much nicer to use, easier to handle
- Graphics
 - > Ruby + graphics = cool demos
- JRuby on Rails
 - > Better deployment options, better performance

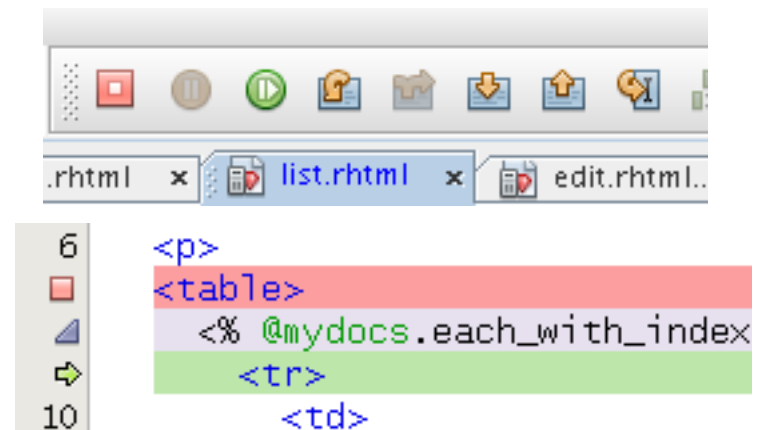
JRuby Enables Tooling

- JRuby's parser used by most Ruby IDEs
 - > NetBeans Ruby Support
 - > Eclipse RDT/RadRails/Aptana, DLTK, 3rd Rail
 - > IntelliJ (not parser, but other areas)
 - > Jedit
- NetBeans is the best right now

```
class Post < ActiveRecord::Migration
  def self.up
    create_table(name, options) ActiveRecord
    create_
  end
end
```

SchemaStatements.create_table(name, opt: ...)

Creates a new table There are two ways to work with it: use the block form or the regular form, like this:



```
6 <p>
  <table>
    <% @mydocs.each_with_index
  <tr>
    <td>
  10
```

Swing GUI Programming

- Swing API is very large, complex
 - > Ruby magic simplifies most of the tricky bits
- Java is a very verbose language
 - > Ruby makes Swing actually fun
- No consistent cross-platform GUI library for Ruby
 - > Swing works everywhere Java does (everywhere)
- No fire-and-forget execution
 - > No dependencies: any script works on any JRuby install

Swing Options

- Cheri - cheri.rubyforge.org
 - > Builder-like DSL syntax
- Profligacy - ihate.rubyforge.org/profligacy
 - > Rubified layout expression language
 - > Trivial event binding without listeners
- MonkeyBars - monkeybars.rubyforge.org
 - > Leverage GUI builders
 - > MVC structure

DEMO

Swing in Ruby

Graphics

- “Processing is an open source programming language and environment for people who want to program images, animation, and interactions.”
 - > Basically a cool Java library for 2D graphics
- Ruby-Processing wraps Processing with JRuby
 - > Cool, rubified 2D graphics environment for you
 - > Eye-candy demos for us
 - > Thanks to Jeremy Ashkenas for putting these together

DEMO
A Face for
Stephen Hawking

Web applications

- Classic Java web dev is too complicated
 - > Modern frameworks follow Rails' lead
- Over-flexible, over-configured
 - > Conventions trump repetition and configuration
- Rails deployment is still a pain
 - > You shouldn't need N processes!
- Rails performance should be better
 - > JRuby has potential to be much faster

JRuby on Rails on GlassFish

- Java app server deployment (WAR files)
 - > N apps
 - > N' concurrent requests
 - > N'' database connections
 - > 1 process
 - Plus full-site management, monitoring, profiling, debugging...
- Designed to scale up and out
 - > ...but at a cost of some agility
- Political standard
 - > Meet them halfway!

DEMO

JRuby on Rails

GlassFish Gem

- Agile development
 - > CLI support, dev time server, no build/deploy
- Simple to use
 - > One-shot execution, both prod and dev
- Scaling, both technical and logistical
 - > Single process for all apps and requests
- Ruby friendly
 - > gem install and go!

DEMO

GlassFish Gem

Production JRuby on Rails

- CSI's Disease Surveillance System
 - > www.govtech.com/gt/323349
- Oracle's Mix – digg-like social customer site
 - > mix.oracle.com
- Sun's MediaCast – file distribution portal
 - > mediacast.sun.com
- ThoughtWorks' Mingle – collaborative project mgmt
 - > mingle.thoughtworks.com
- More on the way!

JRuby Future: 1.1.3 Soon

- 30% faster interpreter
- Multiple compiled perf improvements
- Non-local flow perf fix (break, return, etc in block)
- Dozens more bug fixes
- Several Rails-related bottlenecks removed
- Shared or reduced memory AST (hopefully!)
- RubyGems 1.2 included

JRuby Future: Ruby 1.9

- JRuby will support both 1.8 and 1.9
 - > Flag to switch versions
 - > Maybe flags to switch individual features
- “Wait and see” approach
 - > Waiting for 1.9 adoption, demand
 - > 1.9 is still a work in progress
- 'jruby --1.9' turns on 1.9 features
 - > Only a few: Fiber, proc -> Proc.new, etc
 - > Eventually full 1.9 support

DEMO

JRuby 1.9 Support

JRuby Future: Performance

- JRuby competes on two fronts
 - > Against other Ruby impls
 - > Against other JVM languages
- JVM is a perfect host for languages
 - > Strong static-language backing for Java
 - > Strong dynamic-language backing from Smalltalk
- We can be fast now, faster in the future
 - > Every JVM release is 15-30% faster
 - > Want to run faster? Upgrade!

DEMO

JRuby Performance

JRuby Future: Da Vinci Machine

- <http://openjdk.java.net/projects/mlvm/>
- OpenJDK Multi-language VM
- Feature testbed for future JDKs
 - Anonymous classloading (prototype working)
 - Lightweight method handles (prototype almost done?)
 - Optimized dynamic invocation (waiting on handles)
 - Continuations (proof-of-concept working, prototype coming)
 - Tail call optimizations (under research)
 - Tuples
 - <YOUR FEATURE HERE>
- Crazy cool stuff!

JRuby Future: Other Languages

- Many common goals and challenges
 - > Why keep reinventing the wheel?
- We're all growing the platform
 - > We want the Java platform to succeed
 - > We're working hard to make sure it does
- We're all extending the platform
 - > Challenges are opportunities
 - > The platform isn't perfect, but it's Open
 - > It's up to you and us

JVM Languages Group

- Google for “jvm languages group”
- Implementers from most JVM languages
- Discussions on parsing, compiling, threads, more
- Sharing information, ideas
- Discussing future plans for languages and JVM
- Great fun to read, participate
 - > Even if you're not a language person!

JVM Language Runtime

- DLR-inspired, just getting started
- Gathering the best libraries together
- Provide a fast track for implementation
 - > You might make a language someday
- Solve the hard problems once
 - > You might hate making a language some day
- The pieces are already out there!

Thank you!

- JRuby - www.jruby.org
 - > wiki.jruby.org
- GlassFish - glassfish.dev.java.net
 - > `gem install glassfish`
 - > Looking for bug reports, feature requests!
- NetBeans - www.netbeans.org
 - > Try it out, send feedback!
- Ruby-processing (google it)
- Charlie's blog: headius.blogspot.com