



JRuby – The enterprise ruby

Stefan Magnus Landrø

RubyFools Oslo – 3. April 2008

Agenda

- Introduction to JRuby
- Java Integration
- JRuby on Rails Deployment
- Real life JRuby on Rails
- Q&A

Introduction - History

- JRuby is a Java implementation of the Ruby interpreter
 - Open Source project (CPL/GPL/LGPL)
- Originally created by Jan Arne Petersen, in 2001
 - Direct port of the Ruby 1.6 C code
- Currently 4 core developers
 - Supporting ruby 1.8 semantics
 - Since JRuby 1.1 RC support for Just-In-Time compilation
- JRuby on Rails
 - Support for Rails since version 0.9 (May 2006)
 - Becoming better and better every day (especially due to Sun's efforts)

Introduction - Performance

- Interpreted mode
 - Slower than the C Ruby Reference implementation
 - Rails: 50% to 70% slower
- Just-in-time compilation mode (1.1 RC)
 - Mostly faster than C Ruby 1.8
 - Still slower than C Ruby 1.9
 - Will approach or surpass performance of MRI in release version

Introduction – What's cool about JRuby?

- Call ruby code from java (using JSR-223 alias Java Scripting)
 - Leverage language features found in ruby inside java programs
 - Ruby plugins in a java application
 - Scripting java applications
- Call java code from ruby code (using the jruby interpreter)
 - Make use of company's investment in java-code
 - Make use of open source java libraries when better/faster than ruby alternatives
 - Make use of commercial java libraries
- Deploy JRuby on Rails applications in a standard J2ee servlet container
 - Make use of existing company infrastructure (monitoring, management)

Java Integration

Java Integration – Calling ruby from java

```
import javax.script.ScriptContext;
import javax.script.ScriptEngine;
import javax.script.ScriptEngineManager;
import javax.script.ScriptException;
{...}
ScriptEngineManager m = new ScriptEngineManager();
ScriptEngine rubyEngine = m.getEngineByName("jruby");
ScriptContext context = engine.getContext();

context.setAttribute("label", new Integer(4), ScriptContext.ENGINE_SCOPE);

try{
    rubyEngine.eval("puts 2 + $label", context);
} catch (ScriptException e) {
    e.printStackTrace();
}
```

Java Integration – Accessing java classes

- Mapping of java classes in JRuby
 - Java: ***org.company.MyClass***
 - JRuby: ***Java::OrgCompany::MyClass***
- ***include Java / require 'java'***
 - directive gives access to standard java libraries
- ***import <fully.qualified.JavaClassName>***
 - directive gives access to using class directly
 - works just like in Java
- ***include_package***
 - import complete java package within a jruby module

Java Integration – Extending , adding, implementing

- Extending java classes in JRuby
 - Same syntax as for ruby classes
- Adding behaviour to java classes in JRuby (very powerful!)
 - Same syntax as for ruby classes
- Implementing java interfaces JRuby
 - Use **include** directive just like with mix-ins
- Using java classes that are not part of the standard library
 - Use **require <my.jar>**

Java Integration – Rubyfy it!

- Java accessors get rubyfied in JRuby
 - Java: ***instance.getSize()*** JRuby: ***instance.size***
 - Java: ***instance.setSize(12)*** JRuby: ***instance.size = 12***
- Method name translation
 - Java: ***instance.myTerribleMethodName()***
 - JRuby: ***instance.my_terrible_method_name***
- Extensions to java classes
 - java.util.Map and java.util.Collection ***include Enumerable*** and ***each***
 - And more features that make sense to a ruby programmer

JRuby on Rails Deployment

JRuby on Rails Deployment

- Can use standard C ruby setup (pack of Mongrels etc)
- More attractive to use Java application server/servlet container
 - Reuse existing infrasturcture (clustering, management, monitoring etc)
 - Preferred by bigger companies
- Currently three alternatives
 - Glassfish gem (Sun Application Server: Glassfish V3 – not quite stable yet)
 - Goldspike by JRuby Team
 - Warbler by Nick Sieger
 - Uses Goldspike servlet internally, but uses different packaging and is more stable
 - Several other advantages (See Nick Sieger's blog-post from 4. September 2007)

JRuby on Rails Deployment - Warbler

- Create standard rails application
 - *jruby -S rails rubyfools --database=mysql*
 - *jruby script/generate scaffold post title:string body:text published:boolean*
 - *jruby -S rake db:migrate RAILS_ENV=production*
 - *jruby script\server -eproduction (WEBrick)*
- Add warbler to your installation
 - *jruby -S gem install warbler* **SKIP!**
 - *Modify application.rb* **BUG?**
 - *jruby -S warble [war]*
 - *jruby -S warble war:clean*
- Deploy it to your favourite java application server (e.g. Apache Tomcat)

JRuby on Rails Deployment - Warbler

- Possible to customize lots of things in *warble.rb* (*jruby -S warble config*)
 - Number of JRuby runtimes
 - Gems to be included
 - JNDI database pool

Real life JRuby

Real life JRuby

- 4-month proof of concept project – 4 programmers
- No previous JRuby knowledge – very easy to learn
- Integration with company internal java-based software was very easy
- Netbeans v 6 adds good support for JRuby, Ruby and Rails development
- Problems
 - Serializing java objects in JRuby
 - Solution: First serialize to a java String
 - DRb didn't work properly under JRuby
 - Solution: quit using it
 - JSR 223 JRuby engine implementation is buggy (no tests)
 - Solution: fix it if you have to use it

Conslusion

**JRuby
makes
ruby
enterprise
ready**

Conclusion

■ +++

- Reusing sourcecode written in Java works perfectly
- Deploying to a Java-based application server works flawlessly

■ ---

- Most C-based gems have not been ported to JRuby yet
- Performance is behind MRI

References

- Practical **JRuby on Rails** – Web 2.0 Projects by **Ola Bini**
- JRuby Wiki – **<http://wiki.jruby.org>**
- **user@jruby.codehaus.org** (10-20 posts per day)
- Warbler
 - <http://caldersphere.rubyforge.org/warbler/>
 - <http://blog.nicksieger.com/articles/2007/09/04/warbler-a-little-birdie-to-introduce-your-rails-app-to-java>

Questions & Answers

Q & A