



Introduction to JRuby

Teera Kanokkanjanarat

Teera.K@Sun.Com

<http://blogs.sun.com/teera>



Agenda



Intro to JRuby

What's in it for me?

Java Integration

Performance Discussion

Future of JRuby

Intro to JRuby

- Java implementation of Ruby interpreter
- Full support from Sun
- Providing alternative implementation
- Ruby + JVM
- Current: JRuby 1.0.1 release
- Compatibility: Ruby 1.8.5

Misconception about JRuby

- JRuby is NOT:
 - > an attempt to pollute Ruby
 - > an attempt to alter/add incompatible feature
 - > silver bullet to all Ruby issues
 - > an “admit it Java sucks”
 - > slow...

Design Goals

- Original:
 - > Porting from C Ruby (1.6)
 - > Bringing beauty of Ruby to Java world
- Now:
 - > Taking advantage of Java ecosystem
 - > Optimization for performance

For Ruby Developer

JRuby ? I'm already up and running fine.
..Why another implementation?

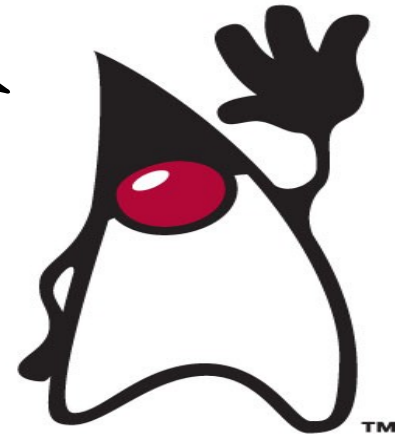


Why JRuby

- Java class and libraries in Ruby
- Ruby code from Java side
- Native Unicode support
- Scalability with native threading (Ruby Thread <-> Java Thread)
- Performance
- Making the way to the enterprise

For Java Developer

Huh..Ruby ? What about it ?
Sounds like another buzz word to me..

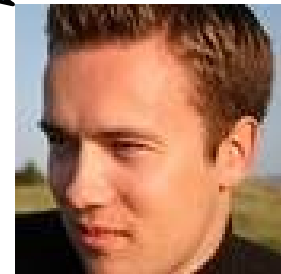


Why JRuby (2)

- Fun, beautiful, powerful language!
- Features missing from Java
 - > Closure (blocks, mixin)
 - > Open classes, metaprogramming
 - > Literals for array, hash, regex
 - > Duck-typing
- Integration (between many libraries)
- Domain Specific Language (DSL)

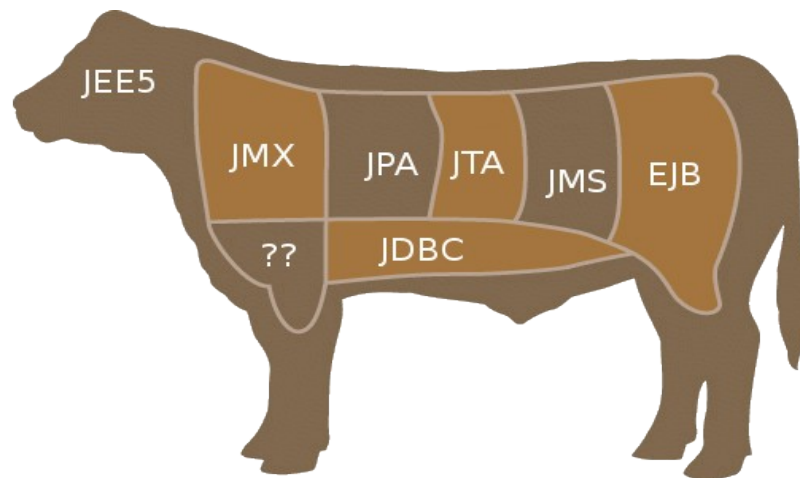
For Rails Developer

Java ? Are you kidding ?
That's so not cool !



Why JRuby on Rails (3)

- Agile & dynamic development
- Leveraging Java App Server & libraries
- Broader database support
- Support Rails and many (most) gems



Code Sample



Java When You Want It

```
include 'java'
```

```
import 'java.lang.ArrayList'
```

```
import 'javax.swing.JFrame'
```

```
list = ArrayList.new
```

```
frame = JFrame.new("Ruby SWINGS!")
```

```
list << frame
```

```
list.each {|f| f.set_size(200,200) }
```

Rubified Java (with JSR 223)

```
import javax.script.*;
public class EvalScript {
    public static void main(String[] args) throws Exception {
        ScriptEngineManager factory = new ScriptEngineManager();
        // Create a JRuby engine.
        ScriptEngine engine = factory.getEngineByName("jruby");
        // Evaluate JRuby code from string.
        try {
            engine.eval("puts('Hello')");
        } catch (ScriptException exception) {
            exception.printStackTrace();
        }
    }
}
```

JDBC Configuration

development:

adapter: jdbc

url: jdbc:mysql://localhost/testapp_dev

username: testapp

password: testapp

driver: com.mysql.jdbc.Driver

development:

adapter: jdbc

jndi: java:comp/env/jdbc/MyAppPool

driver: mysql

Deployment

- Java EE Deployment (taking advantage of Db connection pooling)
 - > WAR File deployment (gold spike, tomcat, jetty, glassfish, weblogic)
 - > Direct deployment (via connector)
 - > Glassfish V3 Rails module
- Rails-like integration with JMX, EJB, JMS, JTA

Performance

- Interpret JRuby is still slower than C Ruby
- Garbage collection
- Dynamic optimization
- Mix-mode virtual machine: compiled JRuby
 - > New! JRuby Compiler
- Mode:
 - > Interpret
 - > Just-in-time (JIT)
 - > Ahead-of-time (AOT)

Issues

- Database support
 - > ActiveRecord-JDBC still needs more work
- Native Extension support
 - > Mongrel: done
 - > Hpricot: done
 - > Rmagick: in progress
- Command line performance

demo



JRuby Future

- Full support on Glassfish V3
- JRuby compiler optimization
- JRuby <-> Java code
- More db and gems support
- Support for Ruby 2.0

<http://jruby.codehaus.org/>





Introduction to JRuby

THANK YOU!

Teera Kanokkanjanarat

Teera.K@Sun.Com

<http://blogs.sun.com/teera>

