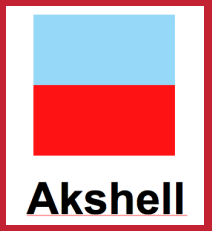
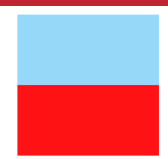


RingoJS

Server-side JavaScript on the Java VM

Hannes Wallnöfer

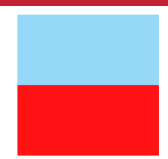


Why the JVM?

- Very stable server side runtime
- Super fast memory allocation, garbage collection
- Fast, scalable networking
- Native multi-threading
- Lots of high quality open source libraries

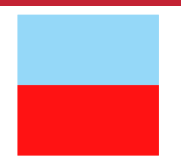
Not so great...

- Big memory footprint



Why JavaScript?

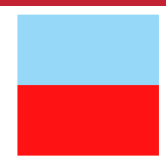
- Simple, well known language
  - Very dynamic: no typing, no classes, prototypal inheritance
  - Objects are just hash tables
- Great for functional programming
  - Closures
  - Functions are first class objects
- Has some quirks
- Lacks concept for structuring code
- Lacks standard libraries for I/O, file system, processes, etc.



Mozilla Rhino

A JavaScript engine written in Java.

- First developed by Netscape, now a Mozilla project
- Smooth Java integration:
  - Full access to Java libraries
  - Implement Java classes in JS
- Supports most of ECMAScript 5 and JavaScript 1.8



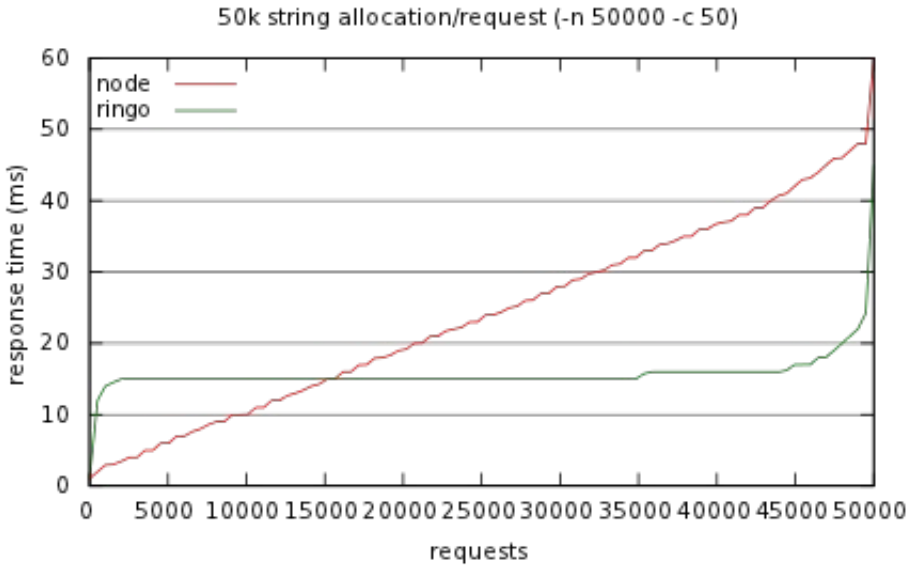
Rhino Performance

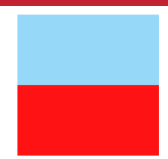
Used to be fast, by today's standards it's too slow

- Java 1.7 `invokedynamic` to the rescue
  - Makes dynamic method invocation as fast and optimizable as ordinary Java method invocation
- We're also working on adopting other V8 optimizations
  - Hidden classes
- Watch us! http://github.com/mozilla/rhino

Rhino performance in practice

- Still fast thanks to JVM
- Beats Node.js in some benchmarks





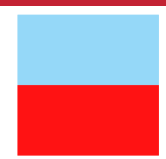
Enter RingoJS

RingoJS provides:

- Module loader based on CommonJS Modules 1.1
  - `var fs = require('fs');`
  - `exports.foo = 'bar';`
- Package support (loosely) based on CommonJS Packages 1.0
  - a directory containing a package.json file

Enter RingoJS (continued)

- Testing framework (test runner, assert)
- I/O, filesystem access
- HTTP client and server
- Command line infrastructure (shell, argument parser, subprocesses)
- Logging (based on SLF4J)
- Development tools (debugger, jsdoc, profiler)
- Scheduler (setTimeout, setInterval)
- Utils (objects, arrays, dates, numbers...)

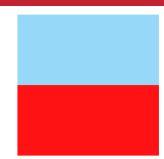


Building RingoJS Web Apps

- Older versions of RingoJS (0.7 and earlier) tried to provide full web framework (templating, database abstraction)
- With 0.8, we're retargeting on core functionality
  - This includes low-level HTTP infrastructure
  - But not an actual high-level framework
  - Expect frameworks to grow out of ecosystem

Deploying RingoJS Web Apps

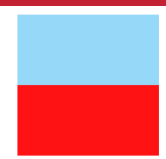
- Debian/Ubuntu packages
  - daemon script calls init() with root uid, start() with normal uid so it can listen on privileged ports.
- Google App Engine
  - free quota is enough to run small sites



Stick: a RingoJS web app framework

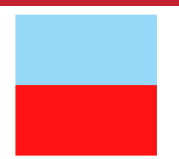
http://github.com/hns/stick

- Request Routing
- Static Files
- Parameter and File Upload Parsing
- Sessions
- Templating
- Error and 404 Handling
- Gzip Compression
- ETag based conditional GET



Asynchronous Web Apps

- Asynchronous response
  - detach current thread from request
- Promise based response
  - resume when promise is fulfilled
- WebSocket support out of the box



Not just for web apps!

- Command line applications
  - argument parser
  - launch subprocesses
- GUI applications
  - based on Swing or any other Java GUI toolkit

Questions?

http://ringojs.org/
http://github.com/ringo/ringojs
Twitter: @ringojs, @hannesw