

Jigsaw Progress in Debian

What we did for GsoC and our Summer Vacation



Sylvestre Ledru, Tom Marble, Guillaume Mazoyer – DebConf11 Banja Luka

Overview

- What is Jigsaw
- GSoC status update
- New features in JDK 7 + JDK 8
 - Notes from the JVM Language Summit
- Next Steps

What is Jigsaw

- Currently a Mercurial forest off of OpenJDK trunk
- The JDK is modularized and versioned such that run time dependencies are resolved using metadata to insure no module is ever linked to more than one version of another
- Replaces the classpath with modulepath
- Lifts us out of jar hell



-
- ```

graph TD
 websvc --> management
 websvc --> logging
 compat --> management
 corba --> management
 corba --> jndi
 kerberos --> management
 kerberos --> jndi
 kerberos --> ssl
 cli-tools --> security-misc
 security-misc --> jndi
 jndi --> crypto
 jndi --> ssl
 crypto --> ssl
 instrument --> logging
 logging --> nio
 xml-dsig --> xml-transform
 xml-transform --> nio
 nio --> base
 scripting --> base
 resources --> base
 management --> client
 client --> xml-parse
 xml-parse --> base
 gui-tools --> client
 jdbbc --> jndi
 jdbbc --> rmi
 jndi --> rmi
 rmi --> ssl
 jx-annotation --> xml-parse
 jx-transaction --> rmi
 charsets --> base
 base --> base
 ssl --> base

```

# Why do you want Jigsaw?

- Smaller download/disk size
- Smaller memory footprint / faster startup time
- Handle module versioning that matches Debian versioning (coordinated behavior between JVM and dpkg).
- New features
- Smaller, simpler booting/porting?

# Why Jigsaw in Debian

- Modular decomposition of functionality
- Version syntax trouble w/ Debian Policy:
  - other systems wildly different: OSGi, maven, RPM
  - RPM doesn't have a spec!
  - Can't do resolution w/o coherent version syntax
- Upstream is looking to Debian
  - Current upstream packaging inadequate
  - Debian can influence JDK 8

# GSoC plan

- Building Jigsaw
- Packaging missing dependencies
- Writing examples
- Packaging Jigsaw
- Testing

# What we have done (1)

- Building Jigsaw on amd64
- Packaging dependencies to run upstream tests
  - jtharness (uploaded June 8)
  - jtreg (uploaded July 7)
- Running the tests to ensure it is working
  - 3484 tests, 29 failed tests (99.17%)
  - Failures in network hardcoding (etc.)
  - IcedTea and/or Upstream working on real failures

# What we have done (2)

- Git repository on Alioth
  - `git+ssh://git.debian.org/git/pkg-java/jigsaw.git`
- Applied Alan Bateman patch
  - To use exploded modules
  - Run tests again
- Writing examples of modules

# A classic hello world (1)

- Declaring a module:
  - File “module-info.java”

```
module com.greetings @ 0.1 {
 requires jdk.base;
 requires org.astro @ 1.2;
 class com.greetings.Hello;
}
```

# A classic hello world (2)

- Source tree can contains several modules

*src/modules/com.greetings/module-info.java*

*src/modules/com.greetings/com/greetings/Hello.java*

*src/modules/org.astro/module-info.java*

*src/modules/org.astro/org/astro/World.java*

- No more classpath, use modulepath

```
$ javac -d modules -modulepath modules \
-sourcepath src/modules \
`find src/modules -name '*.java'`
```

# A classic hello world (3)

- Creating our own module library

```
$ jmod create -L mlib
```

- Installing our modules in this library

```
$ jmod install modules \
org.astro com.greetings -L mlib
```

- Let's say “hello” to the world

```
$ java -L mlib -m com.greetings
Hello, World!
```

# Jigsaw in Debian Next Steps

- Packaging
  - `/usr/lib/jvm/java-8-openjdk/modules`
  - `/usr/share/java/modules`
  - review bootstrapping?
- Integrate patches and security fixes
  - IcedTea
  - Current OpenJDK packaging
  - Additional features: **tailc**
- Test
  - Functionality (jtreg)
  - Performance
- Push changes upstream

# Jigsaw and IcedTea

- IcedTea
  - A build harness for OpenJDK
  - Originally to enable Free Software builds w/o non-free binary plugs
  - Local patches (pulse audio, plugin-in)
  - Support for additional platforms via:  
Zero interpreter, Shark JIT
- Debian Jigsaw collaboration with IcedTea

# New Features in JDK 7

- Fork/join
- Nio2
- Class loader enhancements
- Try with resources
- Multi-catch
- Strings in switch
- JSR 292: invokedynamic

# JDK 7 **invokedynamic**

- Adds a function pointer (“method handle”) to connect classic Java code to byte codes generated by other language compilers: JRuby, Scala, Clojure, JavaScript, etc.
- Significantly reduces size of generated bytecode and invocation overhead
- Allows HotSpot optimization through alternate language code (e.g. inlining, escape analysis)

# openjdk-7 in Debian

- Currently in experimental
- Work by doko and drazzib
- Addressing issues on mips, mipsel
- Port ongoing to kFreeBSD
- Archive rebuild planned with openjdk-7
- Possibly a release goal for Wheezy

# New Features in JDK 8

- Jigsaw is one of the key release drivers
- The build system will be completely redone
- Additional mvlm patches?
  - **tailc** – tail call optimization
- Also from the JVM Language Summit
  - Community interest in a performance test harness
  - Review ProcessBuilder (for better job control)

# Discussion

- Jigsaw Quickstart Guide  
<http://openjdk.java.net/projects/jigsaw/doc/quickstart.html>
- JVM Language Summit 2011  
[http://www.wiki.jvmlangsummit.com/Main\\_Page](http://www.wiki.jvmlangsummit.com/Main_Page)
- Debian Java <[debian-java@lists.debian.org](mailto:debian-java@lists.debian.org)>
- Q/A

Copyright © 2011 Licensed [CC by-sa](#)

