OpenJDK & What it means for the Java Developer

Dalibor Topić
Java F/OSS Ambassador
Sun Microsystems

http://twitter.com/robilad

Dalibor.Topic@Sun.com

Java

Programming Language

Virtual Machine

Cross-platform Programming Environment

Community Of Communities

Pretty successful

~1B LOC

of Open Source Code written in it (acc. to ohloh.net)

That's just the visible space

What else happened in the last 10 years?

Open Source

From fringe to mainstream

Open Innovation across organizational boundaries

Remember shipping containers?

Changed the world

Radically reduced transaction costs

For material goods

Standardized measures

Optimization opportunities

Open Source does the same

For software components

Standardized Legal Containers

Permeable development

Lowered barrier to participation

Collaborative User Innovation

What else happened?

Linux

From fringe to mainstream

Cloud, Cluster, Server, Netbook, TV, Phone, ...

Anyone can create a Linux-based software platform

shipping yard, fleet & port

in one

Organic growth

Cambrian explosion of Linux distros

Selective Pressure On Development Tools on Linux

Strongly Favors Open Source

Manifests itself around:

availability, integration, ease of use

Example:

sudo aptitude install openjdk-6-jdk

vs.

many minutes of manual work

Example:

sudo aptitude build-dep openjdk-6

vs.

many hours of manual work

Open Source
+
Java
+
Linux

OpenJDK

OpenJDK 7

JDK 7

Open Source

GPL v2

Classpath Exception

2006:

From closed to open

First Step:

Get the code out

Putting the effort in perspective

Mozilla

1.2M SLOCs

Eclipse

2.2M SLOCs

OpenJDK

3.5M SLOCs

Done within one year

Managing expectations

Pessimist extremists:

Java will be forked to death!

Well, no.

Didn't happen.

Optimist extremists:

I want a feature!
And I want it now!

Well, probably no.

You can have a go at it yourself, though.

Culture change

2007:

Cleaning up

Second Step:

100 % Open Source

Replacing encumbered third-party code

Removing structural barriers to innovation

Fully open source bootstrap

IcedTea

Gervill

Sound Synthesizer

64-Bit Plugin

Packaging

OpenJDK 6

2008:

Infrastructure & Adoption

Third Step:

Put it to use

Mercurial

External Committers

JDK 7 mainline

Sidestreams of development

Feature Projects

Like NIO2, MLVM & Jigsaw

Making JDK & JVM suitable for more problems

Porting Projects

Like Zero, Shark, BSD Port

Putting OpenJDK in more places

Gentoo, Debian, Fedora, Ubuntu, OpenSUSE, Mandriva, Simply MEPIS, Linux Mint

Red Hat Enterprise Linux, CentOS, Oracle Enterprise Linux

FreeBSD, OpenBSD, NetBSD, MacPorts

Poky Linux, Angstrom, OLPC XO

Removing social barriers to innovation

2009:
JDK 7 Milestones

Fourth step:

Create more opportunities to innovate

Code going from closed to open often has a tough learning curve

Private conversations

Insider knowledge

Easier to learn the ropes writing new code

Public knowledge

Searchable knowledge

In practice:

Modularize the code base

Reducing complexity

Removing intrinsic barriers to innovation

Where are we now?

Gradually growing, diverse Community

Individual Developers

BSD Port

Landon Fuller
Greg Lewis
Kurt Miller
Christos Zoulas

Academia

Type Annotations

Mahmood Ali
Michael Ernst

Corporate Contributors

Sun, Red Hat, Google, AMD

Total: 180

! @Sun.com: 47

25 Projects

Both Incremental Innovation

XRender Pipeline

And Disruptive Innovation

Dynamic language support

Larger Community

IcedTea

Jalimo

Cacao VM

Maxine VM

JNode OS

IKVM.NET

What about the language?

Size matters

Big changes harder then small ones

Project Coin

small language changes

Strings in switch

```
String s = ...
switch(s) {
  case "foo":
    processFoo(s);
    break;
}
```

Improved Type Inference for Generic Instance Creation

AKA: Diamond

```
// type less:
Map<String, List<String>> map = new HashMap<>();
```

Language support for JSR 292

```
Object x = InvokeDynamic.getMeSomething();

MethodHandle mh = ...;
mh.invoke();

int #"strange variable name" = 42;
System.out.println(#"strange variable name");
// prints 42
```

… and more

http://openjdk.java.net/projects/coin

Not so small changes

Type Annotations

JSR 308

Checker framework

@NonNull

@Nullable

javac -processor NullnessChecker MyFile.java

@ReadOnly

http://openjdk.java.net/projects/type-annotations

JSR 294

Improved modularity support in the language

Independent of a module system

Explicit Dependencies + Versioning + Accessibility

module-info.java

src/org/openjdk/SomeTool.java
src/module-info.java

```
module org.openjdk.SomeTool @ 1.0
{
requires module SomeLib @ 1.2;
requires module AnotherLib @ 2.1;
requires module jdk-swing @ 7.0;
class org.openjdk.SomeTool;
}
```

understood by javac

understood by java

$CLASSPATH is dead

Jigsaw

modularity for the JDK

Size matters

well-defined subsets

Just-enough JDK

simple & static

low level

native packaging

jpkg

Now:
deb

Soon:
RPM
SVR4
IPS

Add your format: jigsaw-dev@openjdk.dev.java.net

http://openjdk.java.net/projects/jigsaw/

JDK 7

8 milestones

M1 (finished)

Compressed OOPs

Garbage First Garbage Collector

M2 (finished)

NIO2

URLClassLoader.close()

M3 (finished)

InvokeDynamic

Stream Control Transmission Protocol

Sockets Direct Protocol

Unicode 5.1

ClassLoader Architecture Update

M4
(finished)

Forward-port 6u10 features

Type Annotations

M5
(ongoing)

Update the XML stack

Elliptic Curve Cryptography

Swing Updates

Concurrency & Collections Updates

Project Coin

http://openjdk.java.net/projects/jdk7/

Contribute

http://openjdk.java.net

Patches for JDK 7

http://bugs.openjdk.java.net

Stay up to date

http://planetjdk.org

Discuss development

discuss@openjdk.java.net

jdk7-dev@openjdk.java.net

Q & A

dalibor.topic@sun.com