



Apache Harmony

Geir Magnusson Jr.

The Apache Software Foundation

<http://incubator.apache.org/harmony/>

TS-7722

Goal

Background, Motivations and Status

Agenda

Introduction and Background

Project Goals

Motivations

Modular Architecture

State of the FOSS Java Technology
Universe

Challenges

Summary

Agenda

Introduction and Background

Project Goals

Motivations

Modular Architecture

State of the FOSS Java Technology
Universe

Challenges

Summary

Introduction

Who are we?

- New project in the Apache Incubator
- Started May 18, 2005
- Individuals from many communities
 - Apache Software Foundation
 - Open source and free software communities
 - Academic Researchers
 - Commercial Researchers
 - Vendor employees

Introduction

What Is The Apache Software Foundation?

- 501(c)3 U.S. non-profit corporation
- Volunteers, funded by donations
- Charter—produce open source software under the Apache License
- What we do for our projects
 - Provide infrastructure
 - Provide legal oversight
 - Provide community oversight
- Everything else left to the projects

Introduction

Why do this at the Apache Software Foundation?

- Transparent community and development process
 - Level playing field for all
- Proven model for collaborative development
 - HTTPD, Apache Tomcat, Apache Geronimo
- Demonstrated support of open standards
- History of constructive JCPSM program engagement

Introduction

Apache incubator

- IP:
 - Ensure that contributions are made with permission of copyright owner for initial grant
- Community:
 - Form a transparent, meritocratic, diverse, collaborative community around the software
- We'll take the standard incubator process a step further for Harmony

Why? So people continue to trust our software!

Introduction

Starting harmony

- The time was right
- Broad interest
 - Approached many community members, corporations and individuals
- Sun is cautiously supportive

THANK YOU, SUN

Agenda

Introduction and Background

Project Goals

Motivations

State of the FOSS Java Technology
Universe

Modularity

Challenges

Summary

Goal 1: Community and Architecture

Invite an open, diverse community

- Develop a community-created specification for a modular implementation architecture
 - Open source/free software communities
 - Research
 - Academics
 - Individuals
 - Commercial vendors

Goal 1: Community and Architecture

Modular platform architecture

- What?
 - Well-defined set of components
 - Well-defined interfaces between them
- Why?
 - Freedom of choice for components
 - JIT, GC, Thread Manager, OS Glue
 - Freedom of license for components
 - Free software, open source and proprietary
- More later

Goal 2: J2SE™ Platform Implementation

Same functionality

- Develop a complete, compatible implementation of the Java 2 Platform, Standard Edition (J2SE) 5
- Same as J2SE 5 platform from Sun, IBM, BEA, Apple...
 - Tools—hard, but they exist (Eclipse compiler)
 - Java VM and JIT—harder—sophisticated computer science
 - Class library—hardest—massive amount of work

Goal 2: J2SE Platform Implementation

Same performance

- Must perform as well as the commercial versions
 - Speed
 - Memory
 - 100% Compatible
 - Pass the TCK

Why else would you use it?

Goal 2: J2SE Platform Implementation

Reuse what we can

- Bring together what exists. Avoid re-invention
- Candidates include:
 - GNU Classpath
 - JamVM, Jikes RVM, et al
 - Sun? IBM? BEA? Apple? Others?

Goals Recap

- Enable a community, and create an architecture for implementing the J2SE platform
- Implement the J2SE platform and certify with the TCK

Agenda

Introduction and Background

Project Goals

Motivations

State of the FOSS Java Technology
Universe

Modularity

Challenges

Summary

Java Technology Is Open Source

- We have a vibrant Java technology FOSS community
- JCP 2.5 program, 2002 JavaOneSM conference
 - Specs able to be implemented in open source
 - J2SE 5 platform aka “Tiger” first J2SE spec under 2.5
- Lots of success already:
 - J2EETM platform: Apache Geronimo, JBoss, JOnAS

Let's get to work

Motivations

The contrarians

The contrarians say:

We will fork Java technology

NO!

We will pollute with new technology

NO!

It's too hard—OSS can't do it

NO!

It's all about the license

Maybe...

What about Mustang?

Not FOSS

Motivations

The advocates/true believers

We want an open, collaborative community to implement the specification

- Commercial platform developers
- Academic and commercial researchers
- Independent developers
- Users

...under an open source license

Motivations

The advocates/true believers

Enable widespread adoption of Java technology without duplicate engineering

Want to port to a new platform?

- Just do it—the porting is what you're probably good at, not the Java VM and library. We provide that

Motivations

The advocates/true believers

Provide an acceptable open/free Java platform for the Linux and BSD communities

- Granted not a concern to many in this room
- Java technology can be a 1st-class environment for the FOSS community
- Linux desktop considering Mono—we think we have a better alternative...Java technology!

“Got Linux? You have Java technology!”

Motivations

The advocates/true believers

Java must be the preferred platform for the developing economies

- Right now it's not—due to license issues
- Strong drive for self-determinism
- Lots of talent!

Brazil: all government IT will be free or open source software

- Right now, that precludes Java technology as there's no FOSS implementation

What about others? India? China?

Motivations

Summary

A community-owned, open source implementation of the J2SE platform will:

- Allow general collaboration on what should be ubiquitous platform
- Help drive wider adoption of Java technology
 - To new and existing platforms
 - To the developing world
 - To the open source community

Agenda

Introduction and Background

Project Goals

Motivations

Modularity

State of the FOSS Java Technology
Universe

Challenges

Summary

Modularization

Where are we now?

- Commercial VMs
 - Not much visibility, but there clearly is some
- Research VMs
 - Not much in 1st generation
 - More in 2nd generation
- FOSS VMs
 - No/incidental
- Class libraries
 - No (10 year old legacy...)

Modularization

Why do we want it?

- Modular for plugability
 - Let us change performance profile
 - Client, server, real-time
 - Portability—different operating systems and processors
- Modular for language neutrality (maybe...)
 - Can we implement parts of a VM in different languages?
- Modular for participation
 - Contributors can work on discrete parts, or simultaneous alternatives

Modularization

How will we get there?

- Learn from current research
 - ORP and JikesRVM refactoring
- Look at what exists
 - OSGi
 - FOSS implementations in Eclipse and Oscar
 - JSR 232 looking at OSGi in J2ME™ platform space
- Participate in what is new
 - JSR 277
 - Targeted for J2SE 7 platform (we need it now!)
 - We can give real-time feedback as progresses

Agenda

Introduction and Background

Project Goals

Motivations

Modularity

State of the FOSS Java Technology Universe

Challenges

Summary

State of Open/Free Java Technology

Where are we now?

Some free software projects :

- Kaffe VM: “the Borg of Free VMs”
 - <http://www.kaffe.org/>
 - Ported everywhere, but performance lags commercial VMs
 - Rich community—lots of derivatives/forks
- GNU Classpath: Java class library
 - <http://www.gnu.org/software/classpath/>
 - Long running project, working towards J2SE 1.4 platform

State of Open/Free Java Technology

Where are we now?

Research projects:

- JikesRVM: “Java technology in Java language” research VM
 - Small bit of C, the rest in Java language
 - Popular platform for academic and commercial research
 - In heyday, faster than production C/C++-based VMs
 - <http://jikesrvm.sourceforge.net/>
- ORP: C/C++ research VM
 - Another research platform
 - Ongoing work in and around this VM
 - <http://orp.sourceforge.net/>

State of Open/Free Java Technology

Where are we now?

Interesting things :

- GCJ : Ahead of time compiler
 - Produces native binaries out of your java
 - Uses GNU Classpath
 - <http://gcc.gnu.org/java/>
- IKVM : run Java Technology bytecode on Mono
 - Downright studly and very fast execution
 - Uses GNU Classpath
 - <http://www.ikvm.net>

State of Open/Free Java Technology

Where are we now?

Interesting things :

- JavaLi: Brazilian project focused on producing a certified compatible FOSS J2SE platform implementation
 - Has massive government support
 - Part of the Harmony community
 - Helped bring 3 parts of the govt to the JCP program
 - <http://www.javali.org.br/>

Agenda

Introduction and Background

Project Goals

Motivations

Modularity

State of the FOSS Java Technology
Universe

Challenges

Summary

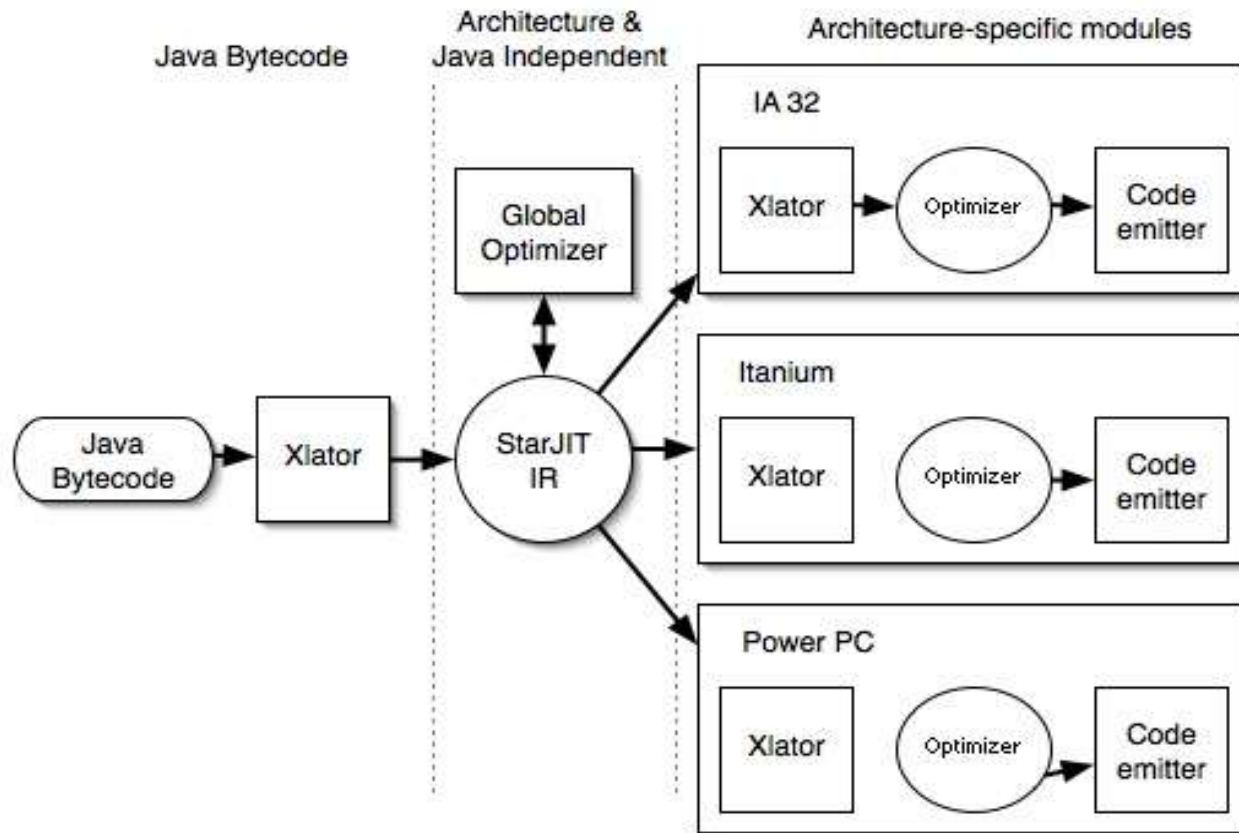
Challenges

Technical

- Language-neutral architecture
 - At least accommodate C/C++ and Java language
- Meaningful modularity?
 - Needs to support broad number of approaches
- Must support portability
 - Must be portable beyond just OS hooks and using an interpreter
 - Example—modular JIT?

Challenges

Technical



Derived from : Intel Technology Journal, "The StarJIT Compiler", Vol 7, Issue 1, 2003D

Challenges

Legal

Goal: protect the IP rights of all community members

- The problem of “taint”
 - Putting the codebase (and thus users) at risk due to some kind of IP inclusion without IP owners permission
- Where and how can this affect us?
 - Individual contributors
 - Many probably have been exposed to class library source
 - Some could have worked on other VM or class library implementations
 - Bulk contributions
 - Large re-purposed contributions must be clean and unencumbered IP

Challenges

Legal

Can tainted people participate?

- Not in technology areas in which they are tainted
- Yes anywhere else in the codebase
 - Document what people have been exposed to
 - We'll segment our repositories

We want to work with Sun to clearly define and reduce number of people considered tainted

Agenda

Introduction and Background

Project Goals

Motivations

State of the FOSS Java Universe

Modularity

Challenges

Summary

Summary

1. Build a community, and community create an implementation architecture
2. Implement that under the Apache License
3. Pass the TCK
 - An idea whose time has come
 - Still early days
 - Good for Java technology, good for the Java technology community

Get involved!

For More Information

- Project website:
 - <http://incubator.apache.org/harmony/>
- Project mail list:
 - Harmony-dev-subscribe@incubator.apache.org

Q&A

Q&A

Submit Session Evaluations for Prizes!

Your opinions are important to Sun

- You can win a \$75.00 gift certificate to the on-site Retail Store by telling Sun what you think!
- Turn in completed forms to enter the daily drawing
- Each evaluation must be turned in the same day as the session presentation
- Five winners will be chosen each day (Sun will send the winners e-mail)
- Drop-off locations: give to the room monitors or use any of the three drop-off stations in the North and South Halls

Note: Winners on Thursday, 6/30, will receive and can redeem certificates via e-mail.



Apache Harmony

Geir Magnusson Jr.

The Apache Software Foundation

<http://incubator.apache.org/harmony/>

TS-7722