Jython

12.5.2006

Vesa-Matti Hartikainen
vesku@mac.com

Outline

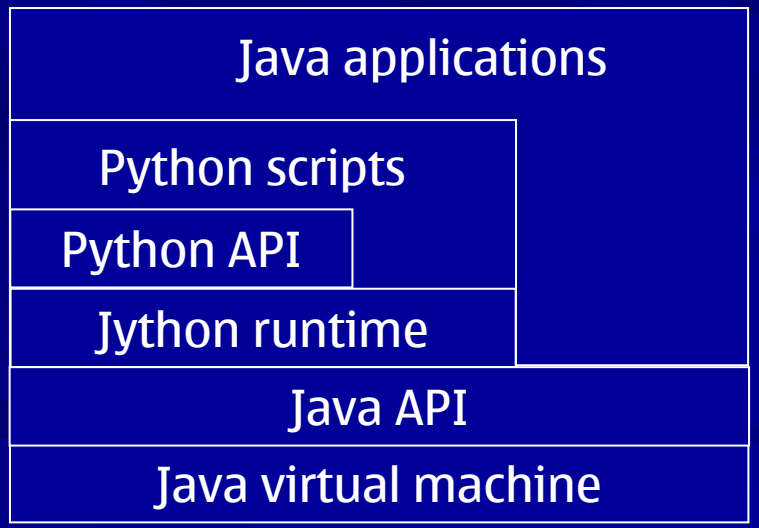
- Introduction
- Jython Features
- Internals
- Demo
- Analysis
- Summary

History

- Development started by Jim Hugunin in 1997
- Jim Hugunin needed a tool to replace combination of Python, C and Matlab code for computations while making his PhD thesis at MIT
- Jython was the answer

Jython combines three good things

- Python-language
- Java Virtual Machine
- Java API



Python

- Modern Scripting language
  - Like Groovy, Ruby, PHP, and JavaScript
- Dynamic typing
- Object-oriented
  - functions, methods, modules, and classes are all first class objects.
- Clear syntax, easy to read and easy to write

Jython features

- Jython runtime for running Python scripts
- Interactive shell
- Compiler for statically compiling Python scripts to classes
- Seamless integration to Java libraries
  - Ability to use Java classes from python code
  - Ability to Java to call Pyhton code
- Nearly all modules from standard Python are available to be used from Jython

Internals

- Pure Java implementation
  - Runs on wide variety of JVMs
- Python-compiler
  - Parser generated with JavaCC
  - Used for statical compilation
  - Used for dynamic compilation
- Interpretator
  - Used for embedded scripts
  - Used for shell
- Access from Python code to Java API relies on Java reflection
  - No additional glue code is needed when new Java APIs are introduced
  - First time Jython loads a JAR it reads the contents (may take some time)

Jython demo

Servlets

Benefits of using Jython (against Java)

- Most benefits come from using dynamic language
  - Tim Bray:

    "*it's faster to write software in dynamic languages, and the (real) benefits you get from an anally type-sensitive compiler can be had more cheaply with modern testing disciplines.*"

  - *More powerful object model, everything is an object*
  - *Simple and powerful language, shorter programs*
- Jython runtime provides additional benefits
  - Possibility to use existing Python code from Java
  - Interactive shell (can be used for example while learning functionality of new APIs)
  - Possibility to use Python as embedded scripting language in a Java application

Benefits of using Jython (compared to Python)

- Java APIs
  - Mature
  - Well designed
  - Javadocs
  - Servlets and JSP
  - There exists Java APIs for practically all needs
- Java Virtual Machine
  - High performance
  - Truly multithreaded
  - Real Garbage collection

So If Jython is great why isn't it used more?

- Developers need to be fluent in two languages
- Many people consider Python runtime fast enough
- Many people consider Java language is good enough
- Some Python features unavailable
  - Cannot access Python modules implemented in C
- JVM startup is slower than Pythons

Uses for Jython

- Web-development
  - Jython servlets
  - Jython JSP applications
  - Applets
- Testing automation
- Rapid application development

Summary

- Jython provides an efficient and relatively easy to use way of using Python as language on top of JVM
- Mature technology
  - Already used in commercial projects
- Jython suites well for web-development
- Also good for
  - Embedded scripting in Java apps
  - Testing automation
  - Prototyping
  - High-level integration of program components

References

- Jyhton homepage
  http://www.jython.org/
- Python homepage
  http://www.python.org/
- Tips for Scripting Java with Jython
  http://www.onjava.com/pub/a/onjava/2002/03/27/jython.html
- Charming Jython
  http://www-128.ibm.com/developerworks/java/library/j-jython.html
- Jim Hugunin – The story of Jython
  http://hugunin.net/story_of_jython.html
- Tim Bray's blog
  http://www.tbray.org/ongoing/When/200x/2004/05/03/Pedroni
- Jython Webapp Tutorial
  http://seanmcgrath.blogspot.com/JythonWebAppTutorialPart1.html