A Quick Tour of JVM Languages

S. Fermigier, Open World Forum, Oct 2009
History

- Java 1.0 in January 1996
- 40 alt. languages implementations in 1997
- ...169 in 2004
- ...240 today (including research and toy languages)
- JVM Language Summits in 2008 and 2009
Why New Languages?

• New challenges have emerged: increased software complexity, agile dev methodologies, multi-core parallelism, DSLs, etc.

• Scientist and language enthousiasts want to try new language features or paradigms

• But the Java Language must evolve in a very conservative way to cater to a huge existing code base + skills investment
PROs of Targeting the JVM

• The JVM is very robust, multiplatform, highly optimized (HotSpot)
• Chance to leverage existing Java code base or libraries
• Wealth of tools for development (profiling, debug) and operations (monitoring)
CONs of Targeting the JVM

• JVM is (currently) not explicitly designed for certain categories of languages (ex: scripting, functional)

• Not suited for all types of systems because memory or CPU constraints (ex: embedded)

• For existing languages (ex: Jython), new implementation not compatible with existing native extensions
Currently Popular

• Jython (1997), an implementation of Python
• Rhino (1997), an implementation of JavaScript
• JRuby (2001), an implementation of Ruby
• Groovy (2003), a dynamic language
• Scala (2003), an object-oriented and functional programming language
• JavaFX Script (2005), a scripting language targeting the Rich Internet application domain
• Clojure (2007), a Lisp dialect
Common Features

• Dynamic typing (scripting): Jython, JRuby, Groovy

• Stronger type systems: Scala

• Functional programming style: Clojure, Scala (also: Jython, JRuby and Groovy)

• Parallelism: Scala Actors, Clojure STM
Da Vinci Project aka MLVM

- (Better) scripting support: JSR 292 =
  - new *InvokeDynamic* instruction at the JVM level
  - change classes and methods at runtime dynamically
- Continuations
- Tail-call elimination
- Interface injection
- Etc.
Next Speakers

- Guillaume Lafforge, Groovy Project Leader / SpringSource
- François Armand, Scala enthousiast
- Alexis Moussine-Pouchkine, Sun Microsystems