

Dynamic Languages on the JVM / MLVM

Ben Evans

What makes a language dynamic?

- Typelessness of variables
- Complex scalar / “primitive” types
- Dispatch determined at runtime
- Modification of the method tables of a type at runtime
- Any others?

Why use the JVM / MLVM?

- **C is hard**
- The JVM is (almost) typeless - although Java is not
- “The implementation is the spec” is bad
- Remove duplication of effort
- Separation of concerns between language and platform
- Stability / Reduce bugs in core language platform features
- Interoperability / EBC (Enterprise Buzzword Compliance)
- Performance
- More unified approaches to deployment

JVM Constraints

- Object Lifecycle + GC
- Threading / Exception Handling
- Method Dispatch – the one sticking point for the JVM's type system
- Classloading – the only way to get new code into the JVM

Overcoming

- OO can model anything – including a dynlang
- Can put dynlang stack frames outside of the Java stack (eg Rhino)
- Dynamic dispatch via many Command Adapters (eg early Jruby)
- To be efficient / actually useful - have to leverage the platform
- Lifecycle, GC, Threading, Exceptions – dynlangs need to use as-is
- Method Dispatch, Classloading (+ some stack frame) – JVM changing to accommodate dynlangs.
- Invokedynamic (aka “Build Call Site”) - key component of dispatch which is changing for dynlangs
- “Autonomous methods” - changes to the way methods are packaged into classes – loosening the link between methods and classes

How Can Dynlangs Help Each Other?

- Jython, JRuby, Rhino (JS), Zero (PHP), Kawa (Scheme), p5vm - must be common ground to share
- Approaches to an AST
- Native call / JNI
- Common optimisation approaches
- Unicode implementation
- Interacting with JIT
- **Talk to Each Other More**

How Can Dynlangs Help The Java Platform?

- Dynamic Dispatch – loads of great use cases for invokedynamic coming out
- Closures and related functional techniques
- Need for better optimisation techniques and more aggressive JITing
- Spur for more innovation in Java itself
- Rise of Scala?

Thank You

- Mail me - I'd love to hear from more people about this
- ben@bpfh.net
- Or talk to me after...