

Dynamic and Static

JRuby and Mirah

Me

- Charles Oliver Nutter
- headius@headius.com
- @headius
- JRuby co-lead
- Mirah creator

JRuby

- Dynamic typed
- Mutable classes
- Arbitrary precision integer arithmetic
- eval and closure/binding tricks

Dynamic Typing

- Mostly not a problem for JVM
 - Method-to-method is important
 - Per-method classes work great
- However, lots of classes = tricky
- Perhaps we can dispatch directly!

Mixed Mode!

- Interpreter enables runtime profiling
- Mixed mode on top of mixed mode...
 - ...hmmm
- Runtime optimization all over again!

What Can We Do?

- Exact target method called
 - Perhaps including arity, signature?
 - Java invocation without reflection?
- Local variables
 - Type and constant propagation

What “Can’t” We Do?

- OSR-like stack restoration
 - Optimistic optimizations
 - Integer overflow
- Branch back to interpreter
- Manual inlining with reasonable fallback

JRuby Dynopt

- “Last method” static optimization
 - Take the last-seen call and add a fast path
 - Guard based on type “serial number”
- “Implicits” for key core types
 - Fixnums, Floats, etc...long or double
- Recursive calls

JRuby Dynopt

Future Directions

- Solve the “can’t do” cases
 - Easy, right? :)
- Optional static types to coarsen guards
- Move to method handles for everything
 - May still need direct invocations?
 - JSR-292 Backport as fast as brute force?

Indy and Method Handles

- One pattern for all dispatch
- Much can be done statically
- See Remi's talk

Mirah

- Statically-typed, object-oriented
- Ruby-inspired syntax
- JVM or Java source backend
- How far can we go?

What If This...

```
public class Foo {  
    private int a;  
  
    public Foo(int a) {  
        this.a = a;  
    }  
  
    public void show() {  
        System.out.println(a);  
    }  
}
```

...Could Be This

```
class Foo
  def initialize(a)
    @a = a
  end

  def show
    puts @a
  end
end
```

Mirah

```
class Foo
  def initialize(a:int)
    @a = a
  end

  def show
    puts @a
  end
end
```

Mirah

- A nicer way to write Java
- Ruby syntax with modifications
- Feels like Ruby
- Compiles to Java/JVM
- No runtime library

Features From Ruby

- Optional arguments ✓
- Internal iteration ✓
- Closures ✓
- Literals ✓
- String interpolation ✓
- Mixins, “open” classes (soon)

Ruby

```
puts "Hello, world!"
```

Mirah

```
puts "Hello, world!"
```

Ruby

```
public static __file__(Lorg/jruby/runtime/ThreadContext;Lorg/jruby/
runtime/builtin/IRubyObject;[Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/
Block;)Lorg/jruby/runtime/builtin/IRubyObject;
  @Lorg/jruby/anno/JRubyMethod;(name="__file__", frame=true, required=0, optional=0,
rest=-2)
  L0
    LINENUMBER 1 L0
    ALOAD 1
    ICONST_0
    INVOKESTATIC ruby/__dash_e__.setPosition (Lorg/jruby/runtime/ThreadContext;I)V
    ALOAD 0
    INVOKEVIRTUAL ruby/__dash_e__.getCallSite0 ()Lorg/jruby/runtime/CallSite;
    ALOAD 1
    ALOAD 2
    ALOAD 2
    ALOAD 0
    ALOAD 1
    GETFIELD org/jruby/runtime/ThreadContext.runtime : Lorg/jruby/Ruby;
    INVOKEVIRTUAL ruby/__dash_e__.getString0 (Lorg/jruby/Ruby;)Lorg/jruby/RubyString;
    INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/
jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/
runtime/builtin/IRubyObject;)Lorg/jruby/runtime/builtin/IRubyObject;
    ARETURN
```

Mirah

```
public static void main(java.lang.String[]);
```

Code:

```
  0: getstatic  #12; //Field java/lang/System.out:Ljava/
io/PrintStream;
  3: ldc      #14; //String Hello, world!
  5: invokevirtual#20; //Method java/io/
PrintStream.println:(Ljava/lang/String;)V
  8: return
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
        java.io.PrintStream temp$1 = java.lang.System.out;
        temp$1.println("Hello, world!");
    }
}
```

Ruby

```
def fib(a)
  if a < 2
    a
  else
    fib(a - 1) + fib(a - 2)
  end
end
```

Mirah

```
def fib(a:int)
  if a < 2
    a
  else
    fib(a - 1) + fib(a - 2)
  end
end
```

Ruby

```
public static method__0$RUBY$fib(Lruby/__dash_e__;Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/Block;)Lorg/jruby/runtime/builtin/IRubyObject;
@Lorg/jruby/anno/JRubyMethod;(name="fib", frame=true, required=1, optional=0, rest=-1)
  ALOAD 3
  ASTORE 9
  L0
  LINENUMBER 1 L0
  ALOAD 1
  ICONST_0
  INVOKESTATIC ruby/__dash_e__.setPosition (Lorg/jruby/runtime/ThreadContext;I)V
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite0 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 9
  LDC 2
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;J)Lorg/jruby/runtime/builtin/IRubyObject;
  INVOKEINTERFACE org/jruby/runtime/builtin/IRubyObject.isTrue ()Z
  IFEQ L1
  ALOAD 9
  GOTO L2
  L1
  FRAME FULL [ruby/__dash_e__ org/jruby/runtime/ThreadContext org/jruby/runtime/builtin/IRubyObject org/jruby/runtime/builtin/IRubyObject org/jruby/runtime/Block
T T T org/jruby/runtime/builtin/IRubyObject] []
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite1 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite2 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 2
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite3 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 9
  LDC 1
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;J)Lorg/jruby/runtime/builtin/IRubyObject;
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;)Lorg/jruby/runtime/builtin/IRubyObject;
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite4 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 2
  ALOAD 0
  INVOKEVIRTUAL ruby/__dash_e__.getCallSite5 ()Lorg/jruby/runtime/CallSite;
  ALOAD 1
  ALOAD 2
  ALOAD 9
  LDC 2
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;J)Lorg/jruby/runtime/builtin/IRubyObject;
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;)Lorg/jruby/runtime/builtin/IRubyObject;
  INVOKEVIRTUAL org/jruby/runtime/CallSite.call (Lorg/jruby/runtime/ThreadContext;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;Lorg/jruby/runtime/builtin/IRubyObject;)Lorg/jruby/runtime/builtin/IRubyObject;
  L2
  FRAME SAME1 org/jruby/runtime/builtin/IRubyObject
  ARETURN
```

Mirah

```
public static int fib(int);
```

Code:

```
0:  iload_0
1:  iconst_2
2:  if_icmplt 9
5:  iconst_0
6:  goto 10
9:  iconst_1
10: ifeq 17
13: iload_0
14: goto 30
17: iload_0
18: iconst_1
19: isub
20: invokestatic #10; //Method fib:(I)I
23: iload_0
24: iconst_2
25: isub
26: invokestatic #10; //Method fib:(I)I
29: iadd
30: ireturn
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
    }
    public static int fib(int a) {
        return (a < 2) ? (a) : ((DashE.fib((a - 1)) +
            DashE.fib((a - 2))));
    }
}
```

Ruby

```
def foo(a = 1, b = 2)
  puts a + b
end
```

Mirah

```
def foo(a:int = 1, b:int = 2)
  puts a + b
end
```

Mirah

```
public static java.io.PrintStream foo(int a, int b) {
    java.io.PrintStream temp$1 =
        java.lang.System.out;
    temp$1.println((a + b));
    return temp$1;
}
public static java.io.PrintStream foo() {
    return foo(1);
}
public static java.io.PrintStream foo(int a) {
    return foo(a, 2);
}
```

Ruby

```
a = [5, 4, 3, 2, 1]
a.each do |x|
  puts x
end
```

Mirah

```
a = [5,4,3,2,1]
a.each do |x|
  puts x
end
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
        java.util.List a =
            java.util.Collections.unmodifiableList(
                java.util.Arrays.asList(1, 2, 3, 4, 5));
        java.util.Iterator __xform_tmp_1 = a.iterator();
        label1:
        while (__xform_tmp_1.hasNext()) {
            java.lang.Object x = __xform_tmp_1.next();
            label2:
            {
                java.io.PrintStream temp$3 = java.lang.System.out;
                temp$3.println(x);
            }
        }
    }
}
```

Ruby

```
t = Thread.new do  
  puts "in thread"  
end
```

Mirah

```
t = Thread.new do  
  puts "in thread"  
end
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
        DashE.__xform_tmp_1 $binding = new DashE.__xform_tmp_1();
        $binding.x = "in thread";
        java.lang.Thread t = new java.lang.Thread(new DashE.__xform_tmp_2($binding));
    }
    public static class __xform_tmp_1 extends java.lang.Object {
        java.lang.String x;
    }
    public static class __xform_tmp_2 extends java.lang.Object implements
        java.lang.Runnable {
        private DashE.__xform_tmp_1 binding;
        public __xform_tmp_2(DashE.__xform_tmp_1 binding) {
            this.binding = binding;
        }
        public void run() {
            DashE.__xform_tmp_1 $binding = this.binding;
            java.io.PrintStream temp$1 = java.lang.System.out;
            temp$1.println($binding.x);
        }
    }
}
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
        DashE.__xform_tmp_1 $binding = new DashE.__xform_tmp_1();
        $binding.x = "in thread";
        java.lang.Thread t = new java.lang.Thread(new DashE.__xform_tmp_2($binding));
    }
    public static class __xform_tmp_1 extends java.lang.Object {
        java.lang.String x;
    }
    public static class __xform_tmp_2 extends java.lang.Object implements
        java.lang.Runnable {
        private DashE.__xform_tmp_1 binding;
        public __xform_tmp_2(DashE.__xform_tmp_1 binding) {
            this.binding = binding;
        }
        public void run() {
            DashE.__xform_tmp_1 $binding = this.binding;
            java.io.PrintStream temp$1 = java.lang.System.out;
            temp$1.println($binding.x);
        }
    }
}
```

Mirah

```
// Generated from DashE
public class DashE extends java.lang.Object {
    public static void main(java.lang.String[] argv) {
        DashE.__xform_tmp_1 $binding = new DashE.__xform_tmp_1();
        $binding.x = "in thread";
        java.lang.Thread t = new java.lang.Thread(new DashE.__xform_tmp_2($binding));
    }
    public static class __xform_tmp_1 extends java.lang.Object {
        java.lang.String x;
    }
    public static class __xform_tmp_2 extends java.lang.Object implements
        java.lang.Runnable {
        private DashE.__xform_tmp_1 binding;
        public __xform_tmp_2(DashE.__xform_tmp_1 binding) {
            this.binding = binding;
        }
        public void run() {
            DashE.__xform_tmp_1 $binding = this.binding;
            java.io.PrintStream temp$1 = java.lang.System.out;
            temp$1.println($binding.x);
        }
    }
}
```

Dynamic Invocation

- Duby and Surinx merged!
- “dynamic” type as in CLR/DLR
- Dynamic invocations
 - Return “dynamic” or Object
 - Method selection at runtime
- Using Attila’s dynalang indy work

Mirah

```
def foo(a:dynamic)  
  puts a.getClass  
end
```

```
foo( 'hello' )
```

Mirah

```
public static java.io.PrintStream foo
(java.lang.Object);
    Code:
        0: getstatic      #14          // Field
java/lang/System.out:Ljava/io/PrintStream;
        3: dup
        4: aload_0
        5: invokedynamic #17,  0          //
NameAndType "dyn:callPropWithThis:getClass":(Ljava/
lang/Object;)Ljava/lang/Object;
        10: invokevirtual #35          // Method
java/io/PrintStream.println:(Ljava/lang/Object;)V
        13: areturn
```

Problems...

- Dynamic invocation happens at runtime
- Compile-time logic doesn't apply
- Multi-method selection needs a library
 - ...wishing it were built into Java 7...

Workshop

- Discuss these problems
- Explore options to solve them
- Learn about VM-level options