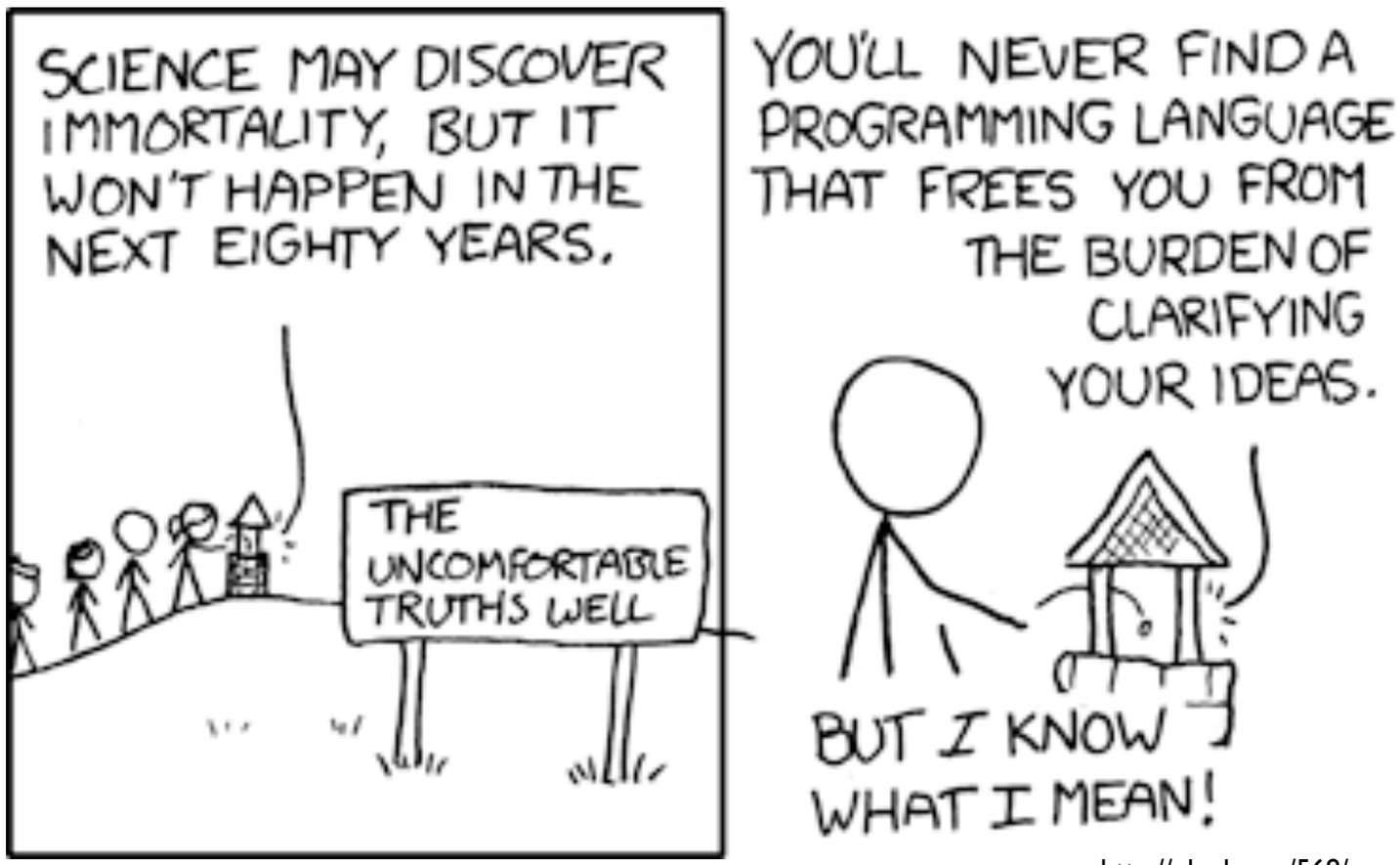


http://xkcd.com/568/

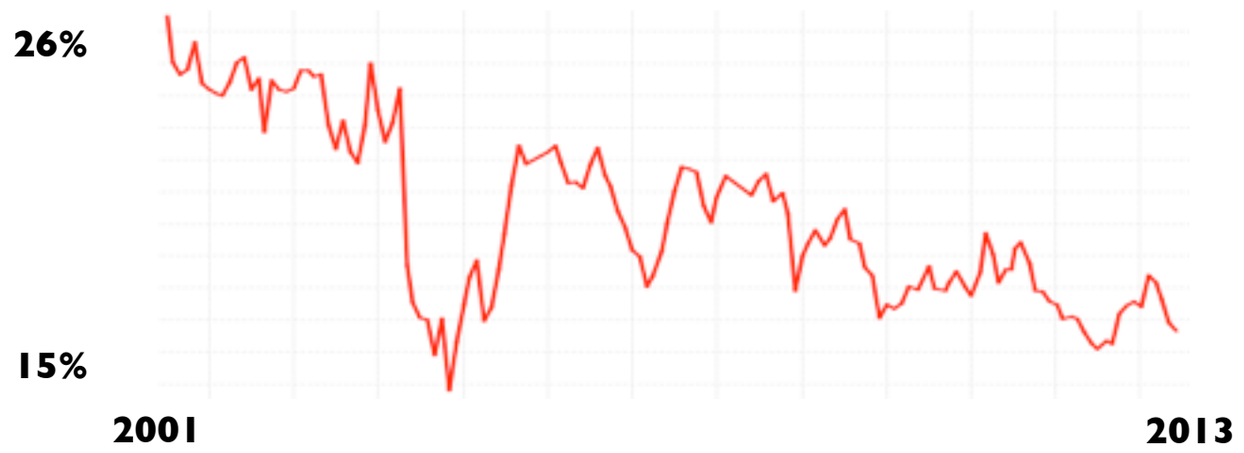
MASON External Language Support

Michael Schader

June 14th, 2013

Purpose

To increase the use of MASON among groups of programmers who cannot or do not wish to use Java.



TIOBE Index of Java Interest

Strategy

Make it easy to play with MASON in common, comfortable, high-productivity programming languages.

History

Original proof of concept with a non-Java language was with Kawa, a JVM Scheme variant.

3 Audiences

- **Language Connoisseurs**
  - Playing with what's interesting, elegant, experimental

  → ***Lisp (Clojure):*** Modern Lisp targeting the Java VM

- **Entry-Level Coders**
  - Practicing basic programming skills in intriguing domains

  → ***JavaScript (Rhino):*** The Language of the Web

- **Domain Specialists**
  - Trying to use MASON to solve problems in their own areas

  → ***Python (Jython):*** Clean, popular, widely applicable

Sample Problem

- **Conway's Game of Life**
  - Cellular automata in a 2D square grid
  - Best known in these parts as Tutorial 1
  - Heart of the algorithm is in *CA.step()*
  - Helper functions *lcount(x,y)* and *setCell(x,y)* kept time-critical work in native Java

Java

Hot Spots – Method	Self time [%]
sim.engine.IterativeRepeat.**step** ()	
sim.field.grid.IntGrid2D.**setTo** ()	

```
public class CA implements Steppable {
    ... // tempGrid decs go here
    public void step(SimState state) {
        ... // gridWidth, gridHeight decs go here
        Tutorial1 tut = (Tutorial1)state;
        tempGrid.setTo(tut.grid);
        int gridWidth = tempGrid.getWidth();
        int gridHeight = tempGrid.getHeight();
        for (int x=0; x<gridWidth; x++)
            for (int y=0; y<gridHeight; y++) {
                int count = lcount(x, y);
                if (count <= 2 || count >= 5) {
                    tut.grid.field[x][y] = 0;
                } else if (count == 3) {
                    tut.grid.field[x][y] = 1;
                }}}
```

4124 steps/sec (baseline)

Kawa

Hot Spots – Method	Self time [%]
sim.engine.IterativeRepeat.**step** ()	
gnu.kawa.reflect.SlotGet.**getSlotValue** ()	
gnu.bytecode.Type.**make** ()	
gnu.expr.PrimProcedure.**apply** ()	

```
(define-simple-class <ca> (<Steppable>)
  ... ;; tempGrid dec goes here
  ((step state :: <sim.engine.SimState>) :: <void>
   (let (...) ;; gridWidth, gridHeight decs go here
     (temp-grid:setTo (<tutorial1>:.grid state))
     (let ((f :: <int[][]> temp-grid:field)
       (g :: <int[][]> (<tutorial1>:.grid state):field))
       (do ((x :: <int> 0 (+ x 1))) ((= x width))
         (do ((y :: <int> 0 (+ y 1))) ((= y height))
           (let ((count :: <int> (lcount x y)))
             (if (or (<= count 2) (>= count 5))
             (set! ((g x) y) 0)
             (if (= count 3)
                 (set! ((g x) y) 1)))))))))
```

2925 steps/sec (1.4x slower)

Clojure

Hot Spots – Method	Self time [%]
user$eval15$fn__16.**invoke** ()	
sim.field.grid.IntGrid2D.**setTo** ()	

```
(let [ca (proxy [Steppable] []
  (step [state]
    (.setTo tempGrid grid)
    (dotimes [x gridWidth]
      (dotimes [y gridHeight]
        (let [count (lcount tempGrid x y)]
          (if (or (<= count 2) (>= count 5))
            (set-cell grid x y 0)
            (if (= count 3)
              (set-cell grid x y 1)))))))])
```

1400 steps/sec (2.9x slower)

Rhino

Hot Spots - Method	Self time [%]
org.mozilla.javascript.NativeJavaObject.**coerceTypeImpl** ()	
org.mozilla.javascript.NativeJavaPackage.**getPkgProperty** ()	
org.mozilla.javascript.NativeJavaObject.**canConvert** ()	
org.mozilla.javascript.MemberBox.**invoke** ()	
org.mozilla.javascript.IdScriptableObject.**get** ()	

```
var ca = new Steppable({
  step: function(state) {
    tempGrid.setTo(grid);
    for (var x=0; x<gridWidth; x++) {
      for (var y=0; y<gridHeight; y++) {
        var count = lcount(tempGrid, x, y);
        if (count <= 2 || count >= 5) {
          setCell(grid, x, y, 0);
        } else if (count == 3) {
          setCell(grid, x, y, 1);
          }}}}});
```

74 steps/sec (55x slower)

Jython

Hot Spots - Method	Self time [%]
org.python.core.PyType.**fromClass** ()	
org.python.core.PyObjectDerived.**__tojava__** ()	
org.python.core.PyFrame.**getglobal** ()	

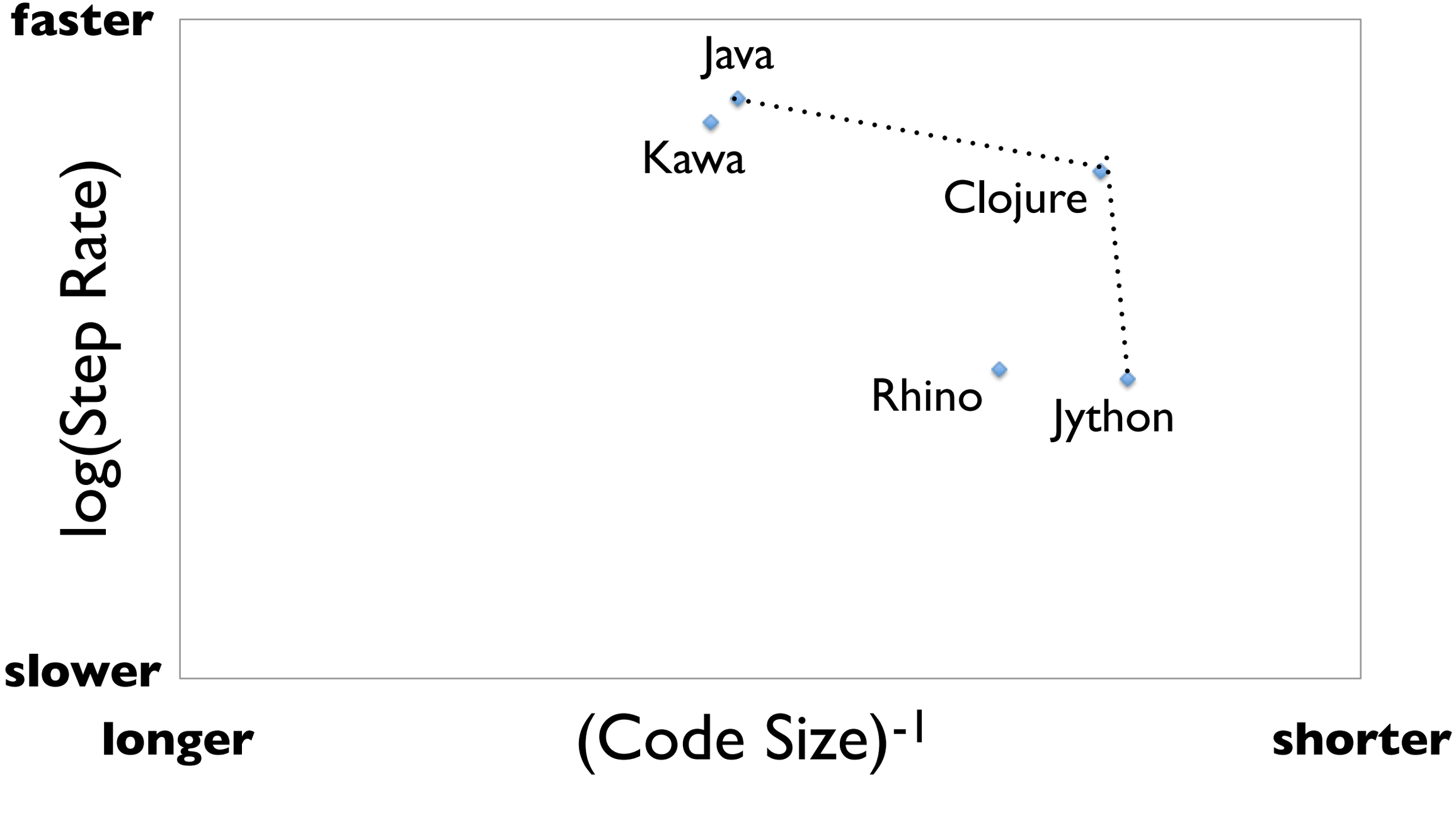
```
class Ca(Steppable):
    def step(self, state):
        tempGrid.setTo(grid)
        for x in range(gridWidth):
            for y in range(gridHeight):
                count = lcount(tempGrid, x, y)
                if count <= 2 or count >= 5:
                    setCell(grid, x, y, 0)
                elif count == 3:
                    setCell(grid, x, y, 1)
```

64 steps/sec (65x slower)

Results

	Steps/sec	Time Factor	Code Size
Java	4125	1.0	354
Kawa	2925	1.4	372
Clojure	1400	2.9	211
Rhino	74	55	238
Jython	64	65	205

Pareto Front



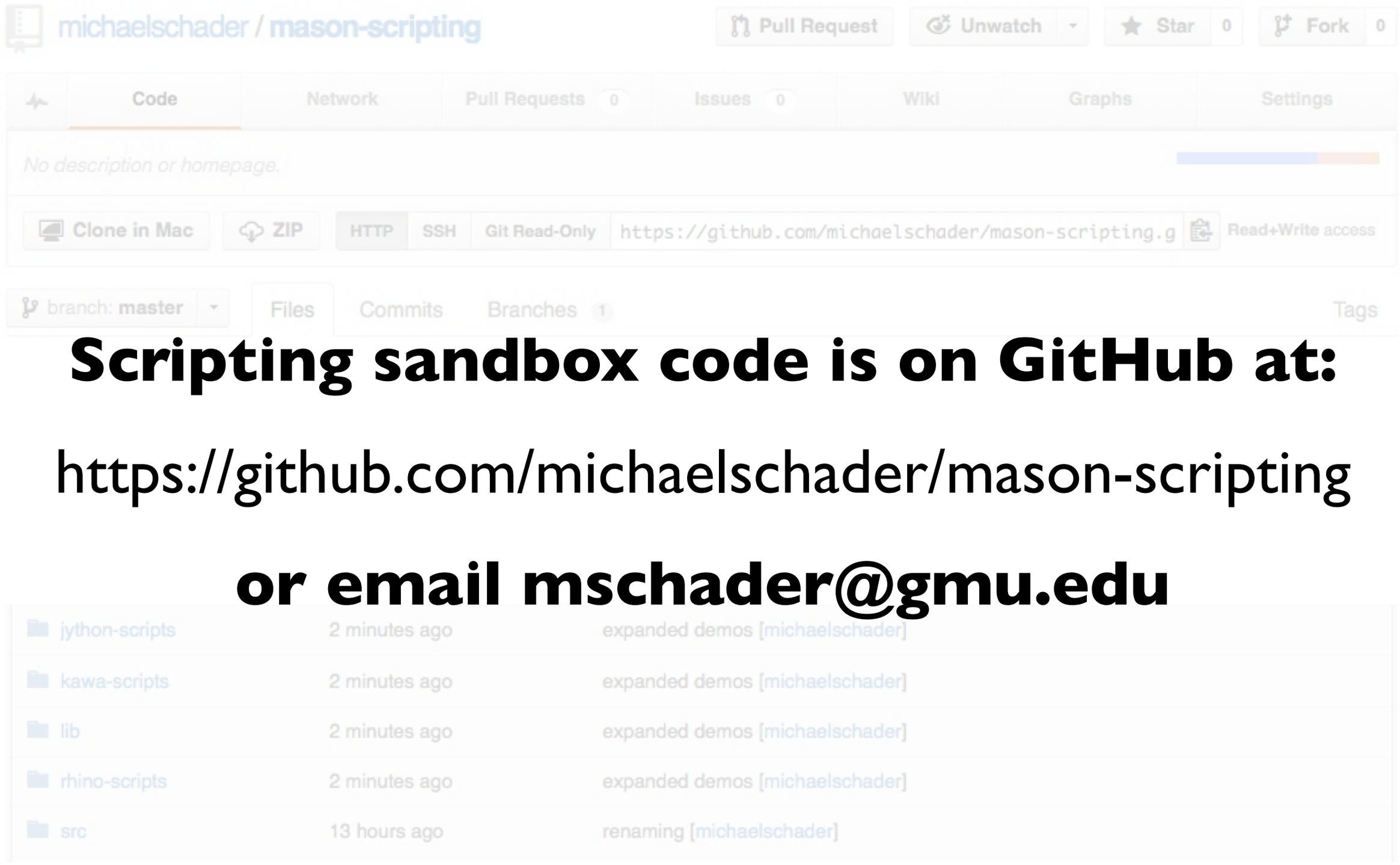
Future Work

- **Language-Specific Tutorials**
  - Step-by-step instructions for getting up and running with MASON in each of the JVM languages
    - Clojure version will showcase elegance
    - JavaScript version will emphasize ease and speed of getting started
    - Jython version will leverage common numerical computing idioms

Future Work

- **Domain-Focused Optimizations**
  - High-performance libraries for common operations in particular environments to ease development and improve performance
- **Agent Language Development**
  - A MASON domain-specific language that leverages libraries and macros to speed effective programming

Collaboration



Scripting sandbox code is on GitHub at:

https://github.com/michaelschader/mason-scripting

or email mschader@gmu.edu