(based on Patrick Thomson’s C4[3] presentation)

“The effective exploitation of the powers of abstraction must be regarded as one of the most vital activities of a computer programmer.”

- Edsger Dijkstra

Objective-C

We build the best desktop apps out there, but we do it in spite of ObjC.

- Patrick Thomson, C4[3]

1. Code Reuse

Singletons

1. **create a static, shared instance**
2. **initialize once and only once**
3. **add an** `sharedInstance` **accessor method**

… for each singleton class

```ruby
class Example
  include Singleton
  # your methods here
end
foo = Example.instance
```

2. Safety

We see C's unsafeness throughout Objective-C

3. Syntactic Abstraction

```objc
[NSArray arrayWithObjects: @"a", @"b", @"c", nil];
```

vs.

```objc
["a", "b", "c"]
```

```objc
[NSDictionary dictionaryWithObjectsAndKeys:
  @"Vienna", @"location", @"Cocoaheads", @"event", nil];
```

vs.

```ruby
{ "location" => "Vienna", "event" => "Cocoaheads" }
```

```
[foo compare:bar] == NSComparisonResultAscending
```

vs.

```
foo < bar
```

I want a language that lets Cocoa shine.

MacRuby

New Ruby Implementation

+

Ruby

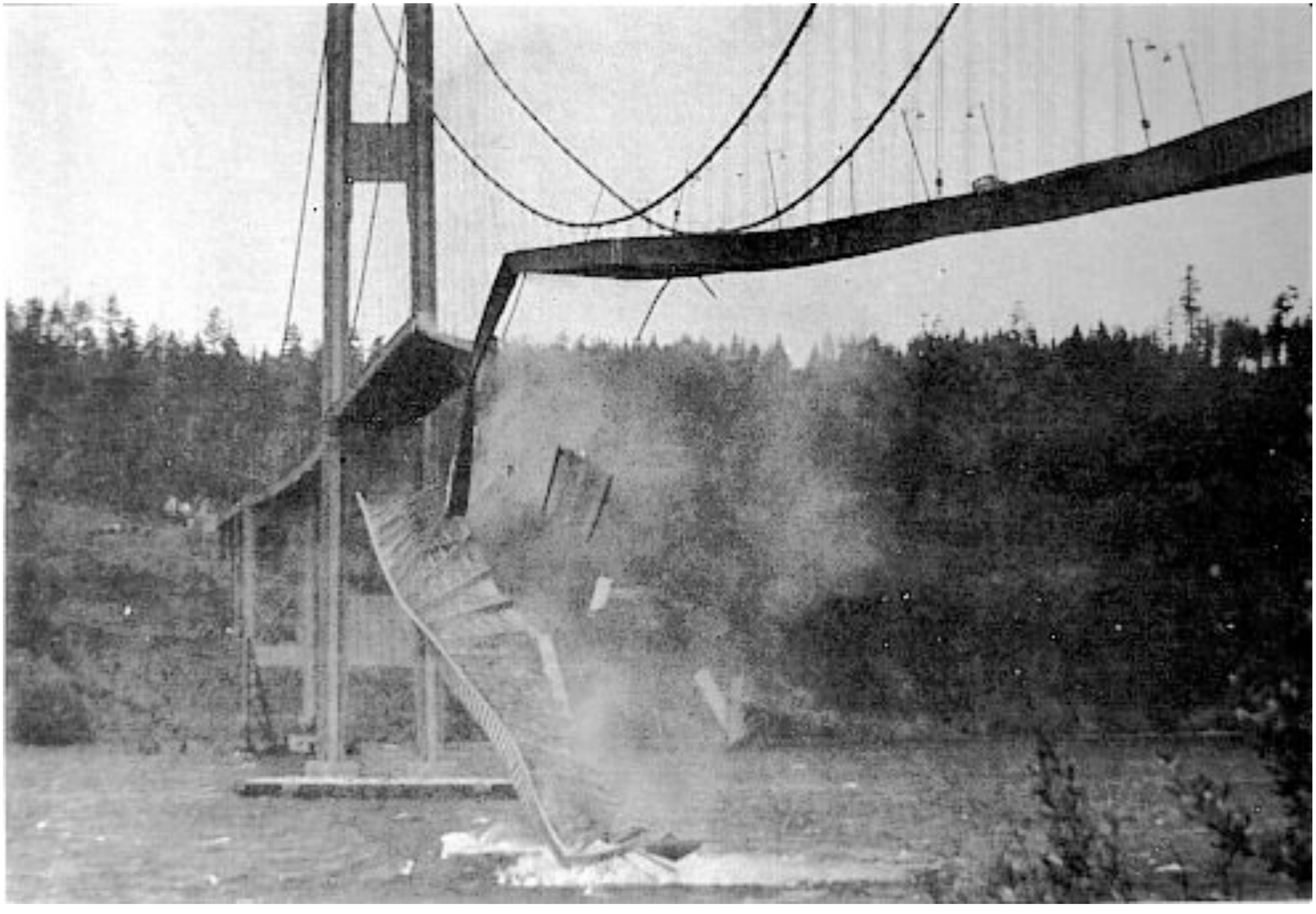
LLVM

```
>> "Cocoaheads".class
=> NSMutableString
>> [1, 2, 3].class
=> NSMutableArray
>> {:tollfree => "bridging"}.class
=> NSMutableDictionary
>> {:tollfree => "bridging"}.class.ancestors
=> [NSMutableDictionary, NSDictionary, Enumerable,
NSObject, Kernel]
>> "/usr/local/bin".pathComponents
=> ["/", "usr", "local", "bin"]
```

```
>> 3000.class
=> Fixnum
>> 3000.is_a? NSNumber
=> true
>> 3000.is_a? NSObject
=> true
>> 3000.class.is_a? NSObject
=> true
```

Everything is a NSObject

MacRuby != RubyCocoa



```
obj.setValue_forKey_(val, key)
```

provides real solutions to real-world problems

Global Interpreter Locks

Only one system thread can run on the interpreter at any given time

~~Multithreading~~

MacRuby has no GIL!

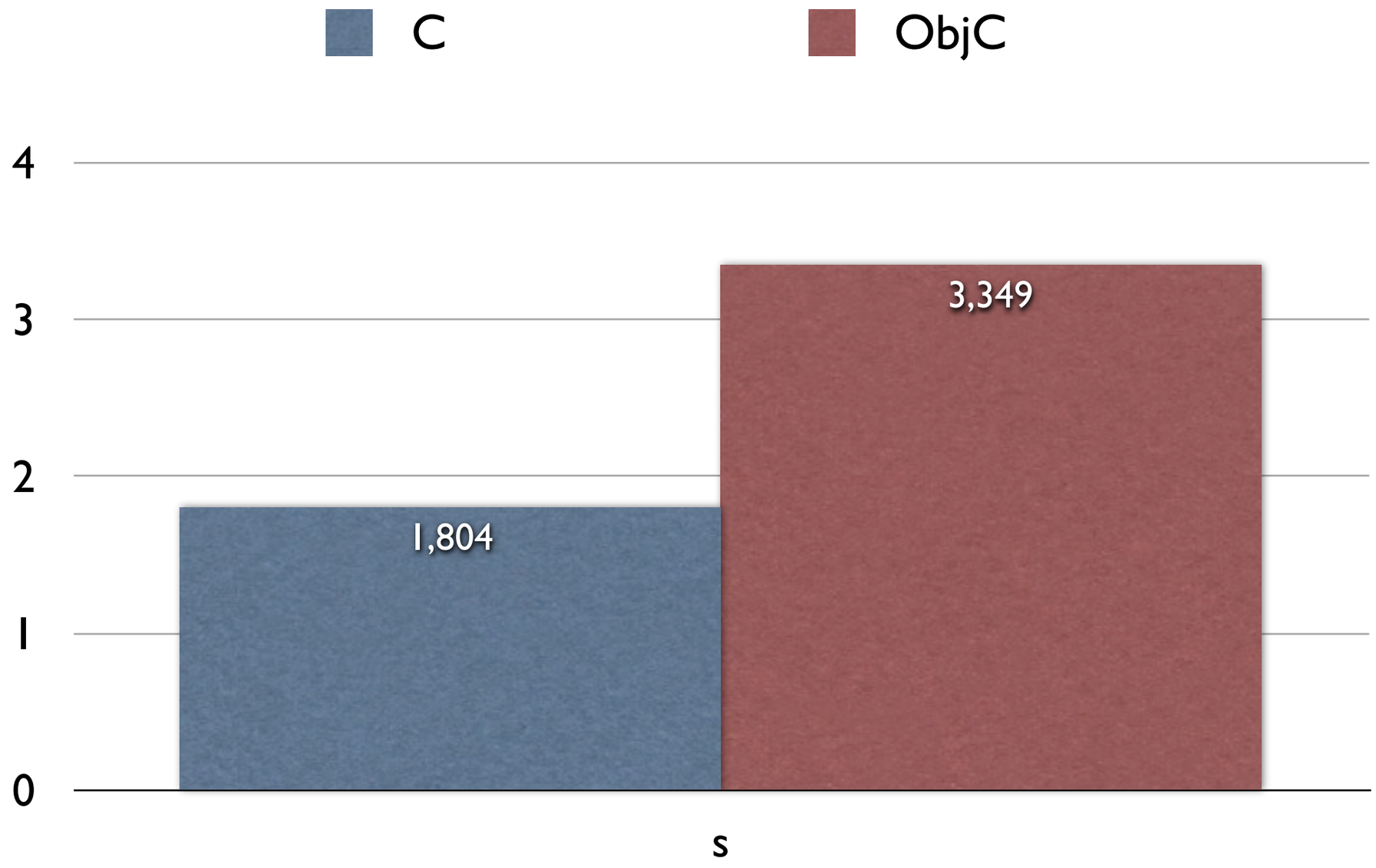
Fast

```c
#include <stdio.h>

static long long fib(long long n)
{
    if (n < 3) {
        return 1;
    } else {
        return fib(n - 1) + fib(n - 2);
    }
}

int main()
{
    printf("fib(40) = %lld\n", fib(40));
    return 0;
}
```

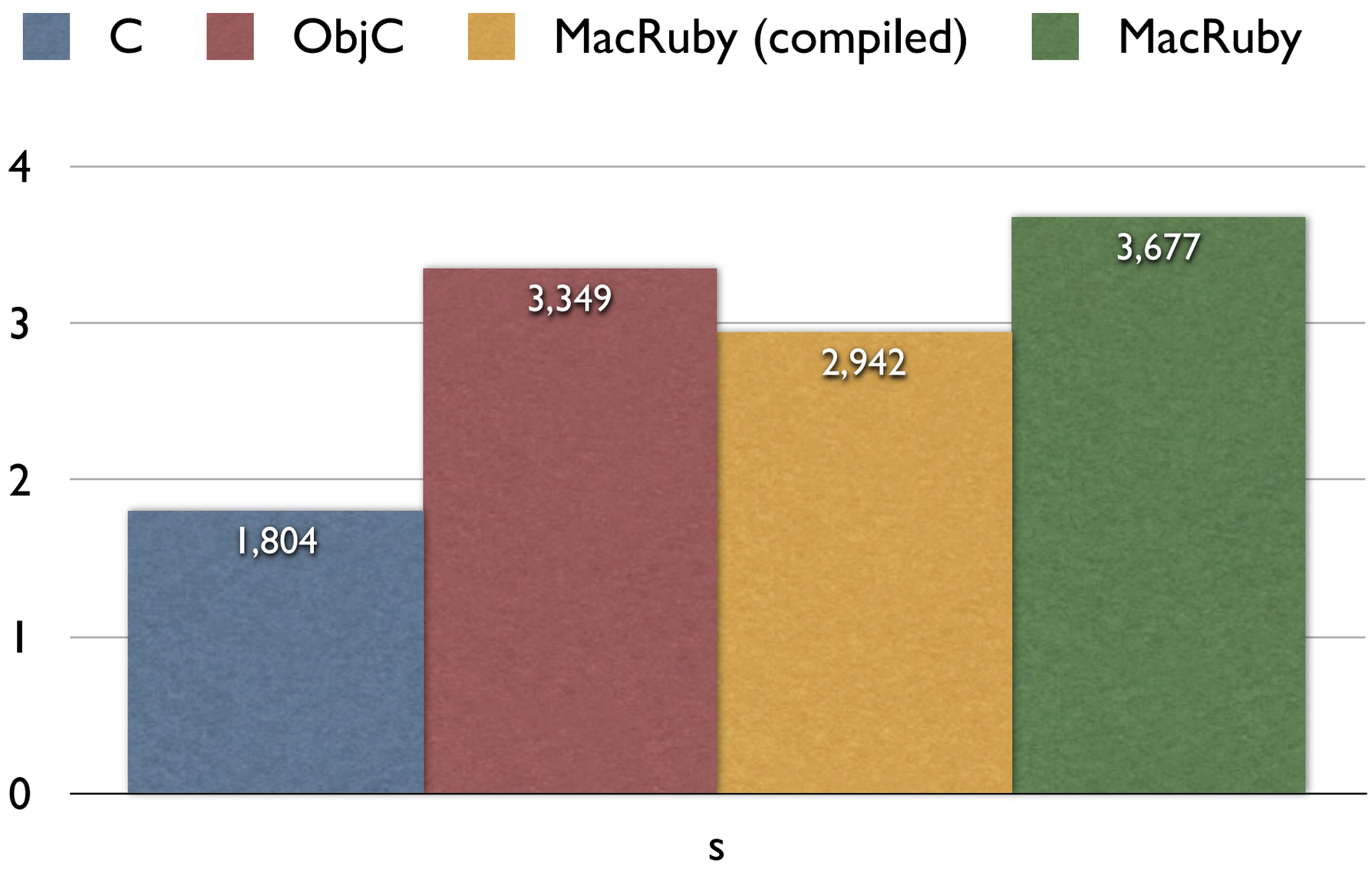
```objc
@implementation Fib

- (long long)fib:(long long)n
{
    if (n < 3) {
        return 1;
    } else {
        return [self fib:n - 1] + [self fib:n - 2];
    }
}

@end
```

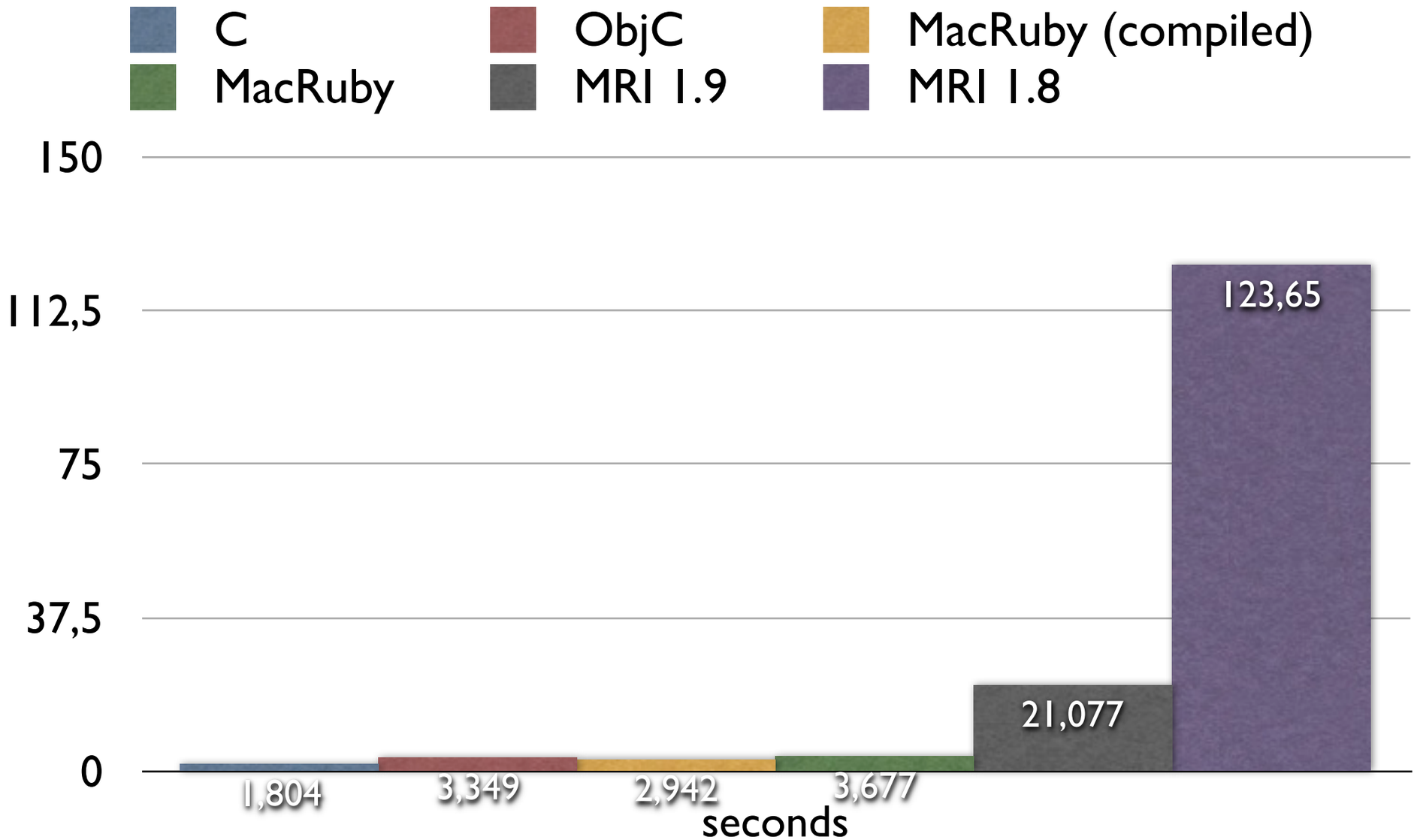
…

```objc
int main (int argc, const char * argv[]) {
    Fib *fib = [Fib new];

    // insert code here...
    NSLog(@"fib(40) = %lld", [fib fib:40]);

    return 0;
}
```


C
ObjC
4
3
2
1
0
1,804
3,349
s

```
def fib(n)
  if n < 3
    1
  else
    fib(n-1) + fib(n-2)
  end
end

puts "fib(40) = #{fib(40)}"
```


C
ObjC
MacRuby (compiled)
MacRuby
4
3
2
1
0
1,804
3,349
2,942
3,677
s


C
ObjC
MacRuby (compiled)
MacRuby
MRI 1.9
MRI 1.8
112,5
75
37,5
0
1,804
3,349
2,942
3,677
21,077
123,65
seconds

```
obj.setValue_forKey_(val, key)
```

```
[obj setValue:val forKey:key]
```

```
obj.setValue(val, forKey:key)
```

```
[obj setValue:val forKey:key]
```

```
obj.setValue val, forKey:key
```

HotCocoa

DSL for Cocoa

```
[[NSImage alloc] initWithContentsOfFile:path];
```

becomes

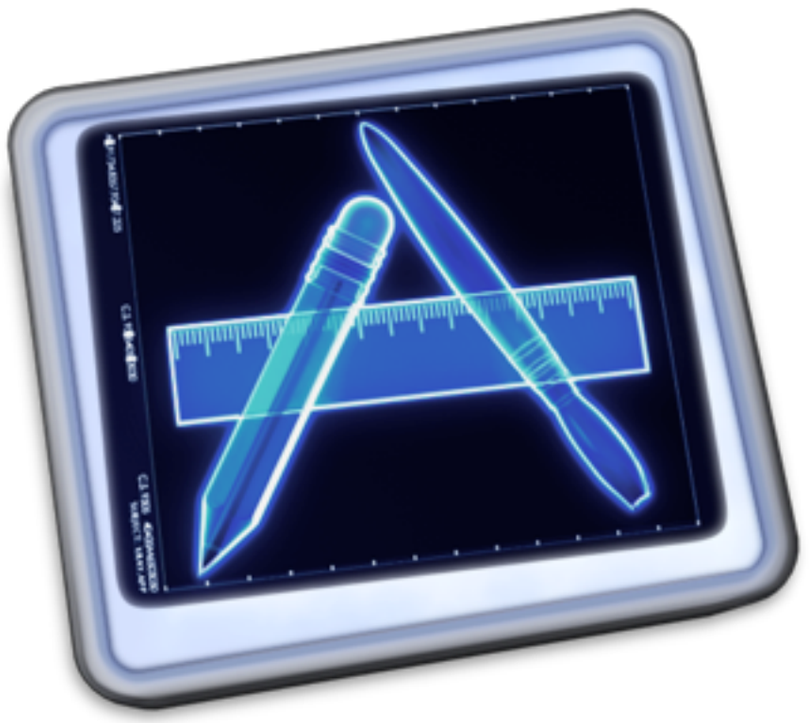
```
image(:file => path)
```

```objc
[[NSGradient alloc]
 initWithStartingColor: [NSColor greyColor]
           endingColor: [NSColor blueColor]];
```

becomes

```ruby
gradient(:start => color(:name => "grey"),
         :end   => color(:name => "blue"))
```





twitter

Home Profile Find People Settings Help Sign out

Spent an hour hunting for the method to import a local Obj-C class into MacRuby. Turns out there is none...it just works. Very Zen.

9:12 PM May 1st from web

topfunky
Geoffrey Grosenbach

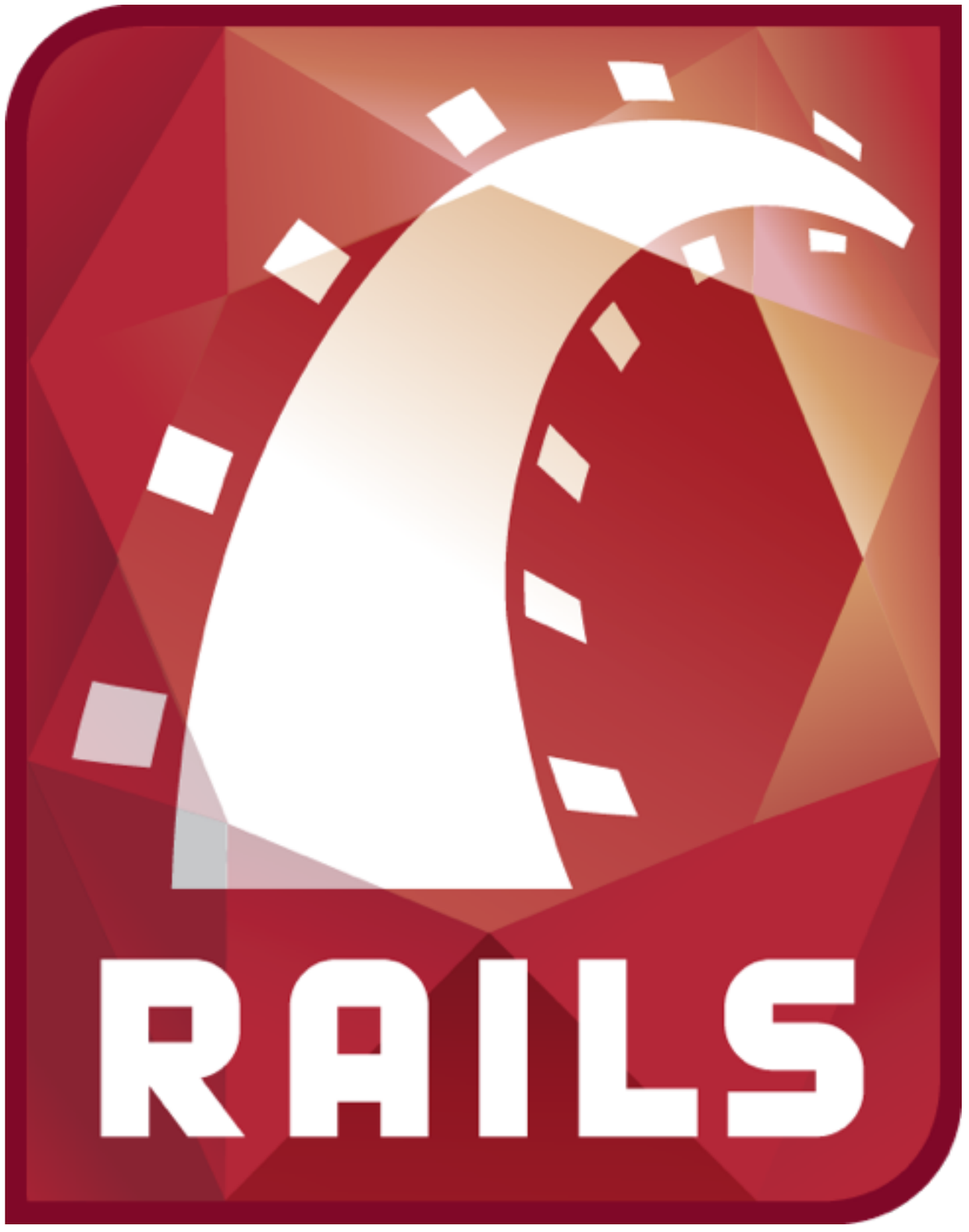
regular expressions
namespaces
mixins
operator overloading
runtime evaluation…

static typing

almost 0.5

nightly builds for 10.6 at

http://macruby.icoretech.org


RAILS

not yet

(speculation)


cingular
9:41 AM
SMS
Text
Tuesday
9
Calendar
Photos
Camera
Calculator
Stocks
280
Maps
73°
Weather
Notes
Clock
Settings
Phone
Mail
Web
iPod

possible? yes.

certain? no.

@MacRuby

@lrz
@importantshock
@benstiglitz
@vincentisambart
@mattetti
@alloy

http://macruby.org