**$ ruby class\_check1.rb**

0.022767841

chp2/class\_check2.rb

require *'benchmark'*

obj = *"sample string"*

time = Benchmark.realtime **do**

100000.times **do**

obj.is\_a?(String)

**end**

**end**

puts time

**$ ruby class\_check2.rb**

0.019568893

在一个中等大的循环中，再次进行100,000次迭代，这样的检查需要19-22 ms。这听起来并不糟糕，但是，例如，Rails应用程序可以为每个请求调用比较运算符超过100万次，并且花费超过200毫秒进行类型检查。

将类型检查从迭代器或经常调用的函数和运算符移开是个好主意。如果你不能，不幸的是，你无能为力。

BigDecimal::==(String)

从数据库获取数据的代码大量使用大小数。对于Rails应用程序尤其如此。此类代码通常从它从数据库中读取的字符串创建BigDecimal，然后将其直接与字符串进行比较。

问题是，在Ruby 1.9.3及更低版本中进行此比较的自然方式令人难以置信地缓慢：

chp2/bigdecimal1.rb

require *'bigdecimal'*

require *'benchmark'*

x = BigDecimal(*"10.2"*)

time = Benchmark.realtime **do**

100000.times **do**

x == *"10.2"*

**end**

**end**

puts time

**$ rbenv shell 1.9.3-p551**

**$ ruby bigdecimal1.rb**

0.773866128