**$ rbenv shell 2.0.0-p598**

**$ ruby bigdecimal1.rb**

0.025224029

**$ rbenv shell 2.1.5**

**$ ruby bigdecimal1.rb**

0.027570681

**$ rbenv shell 2.2.0**

**$ ruby bigdecimal1.rb**

0.02474011096637696

较旧的Rubys具有令人无法接受的BigDecimal :: ==函数的慢实现。这个性能问题随着Ruby 2.0升级而消失。但如果你无法升级，请使用这个聪明的技巧。在比较之前将BigDecimal转换为String：

chp2/bigdecimal2.rb

require *'bigdecimal'*

require *'benchmark'*

x = BigDecimal(*"10.2"*)

time = Benchmark.realtime **do**

100000.times **do**

x.to\_s == *"10.2"*

**end**

**end**

puts time

**$ rbenv shell 1.9.3-p545**

**$ ruby bigdecimal2.rb**

0.195041792

这个hack比Ruby 2.x实现快三到四倍 - 不快四十倍，但仍然是一个改进。

Write Less Ruby

教我编程的大师之一曾经说过最好的代码是不存在的代码。如果我们可以在不编写任何代码的情况下解决问题，那么我们就不必对其进行优化。对？

不幸的是，在现实世界中，我们仍然编写代码来解决我们的问题。但这并不意味着它必须是Ruby代码。其他工具更好地做某些事情。我们已经看到Ruby在两个方面特别糟糕：大型数据集处理和复杂计算。那么让我们看看你可以使用什么，以及它如何提高性能。