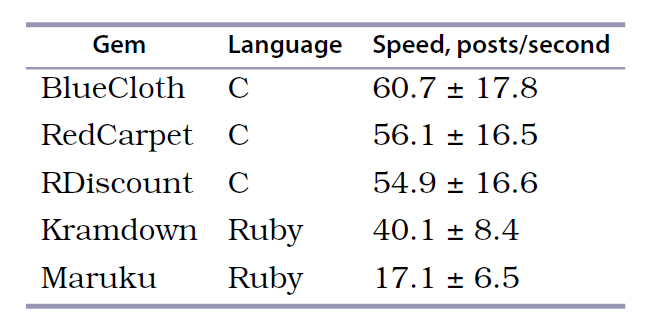


用C编写的相同代码几乎快八倍！ 作为奖励，它使用175 MB（或更少）的内存。 两者都有很大的改进。 这就是为什么我建议每个坚持使用旧版Ruby 1.8的人都应该使用Date :: Performance gem的原因。

还有一些gems在C中实现特定的任务。最好的例子是markdown libraries。 有些是用C语言编写的，有些是用Ruby编写的。 以下是Jekyll博客引擎贡献者之一Jashank Jeremy的性能比较：



最慢的C实现（RDiscount）比最快的Ruby实现（Kramdown）快1.4倍。 最快和最慢之间的差异是惊人的的3.5倍。 正如你所看到的，在本机代码中搜索gems这项困难的任务是完全有意义的。

**小贴士**

我们在本章中看到，为了使Ruby代码更快，只需要考虑三件事：

①通过避免额外的分配和内存泄漏来优化内存；

②编写速度更快的迭代器，减少时间和内存；

③通过专业工具完成工作来减少Ruby代码。

这些技术的优点在于，你可以将它们应用于任何Ruby程序，使其速度提高十倍。 但是大多数Ruby开发人员都在编写Rails应用程序，所以现在是时候深入了解并将我们的优化技术应用于Ruby on Rails。