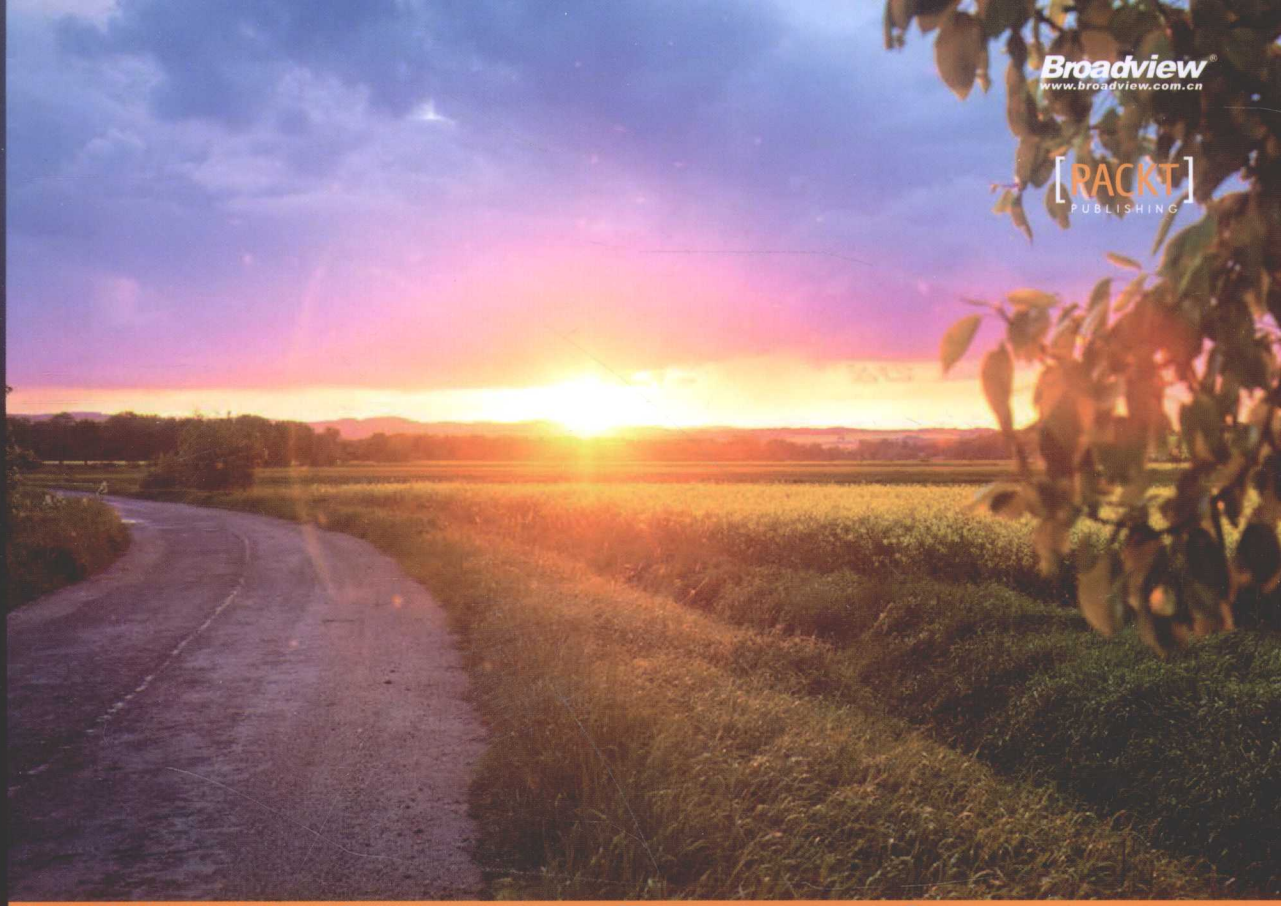


Broadview
www.broadview.com.cn

PACKT
PUBLISHING

R High Performance Programming

R高性能编程

用整套解决方案与高超技巧突破性能瓶颈

Aloysius Lim　William Tjhi　著
唐李洋　译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

R High Performance Programming

R高性能编程

用整套解决方案与高超技巧突破性能瓶颈

Aloysius Lim　William Tjhi　著
唐李洋　译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

R 语言是专门为统计学和数据分析开发的解释型语言，主要用于数学建模、统计计算、数据处理、可视化等几个方面。近年来，受大数据的影响，R 语言备受业界追捧。与此同时，工业界和学术界都越来越要求 R 具备高效分析和处理大规模数据的能力。但是，由于 R 本身的设计问题，它能够有效处理的数据规模和计算复杂度有限。

为此，这本书提供了较为完整的参考方案和技术指南。本书首先解释了 CPU、内存和磁盘 I/O 等影响 R 性能的三个因素，剖析了 R 在处理大规模数据时出现性能瓶颈的原因。在理解了 R 的设计原理及其性能限制的基础之上，本书给出了提升 R 性能的方法和技术。例如：尽量使用向量化运算避免不必要的计算开销，预分配内存避免不必要的动态内存分配，使用编译代码减少 CPU 时间，删除不必要的中间数据释放内存占用，通过运行时计算代替永久存储减少内存使用，使用内存映射文件处理大型数据集，使用并行计算技术优化代码，以及接入数据库处理工具，等等。最后，本书提供了如何在 R 中使用 Hadoop 的方法，以处理和分析大数据。

版权贸易合同登记号　图字：01-2015-4873

图书在版编目（CIP）数据

R 高性能编程 / 利姆（Lim,A.），Tjhi（Tjhi,W.）著；唐李洋译. —北京：电子工业出版社，2015.12
书名原文：R High Performance Programming
ISBN 978-7-121-27396-4

Ⅰ.①R… Ⅱ.①利… ②T… ③唐… Ⅲ.①程序语言－程序设计 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字(2015)第 243360 号

策划编辑：张春雨
责任编辑：张春雨
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：10　字数：224 千字
版　　次：2015 年 12 月第 1 版
印　　次：2015 年 12 月第 1 次印刷
定　　价：55.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

作者简介

Aloysius Lim

擅长将复杂的数据和模型表述为简单易懂的见解。作为 About People 的联合创办者、数据科学家及设计顾问，他喜欢解决问题，以及帮助他人寻找利用数据解决业务难题的实践方案。在政府、教育和零售行业长达 7 年的丰富经验，造就了他寻找具有创造性的解决办法的独特视角。

衷心地感谢上帝给我机会撰写本书，让我分享这些知识。在这个过程中，我亲爱的妻子 Bethany 给予了极大的支持和鼓励。谢谢你全部的爱，亲爱的。非常感谢我的搭档 William，他是我灵感和洞察力的来源。

William Tjhi

数据科学家，在学术、政府及工业界具有多年的工作经验。他自读博期间开始了数据科学之旅，研究了一些新算法以提高高维数据聚类的健壮性。取得博士学位之后，他从基础研究转向了应用研究，致力于采用机器学习方法解决分子生物学和传染病学中的各种问题。他在一些同行评审的期刊和会议上发表了部分研究成果。随着大数据的兴起，William 离开了学术界，转而投入工业界，开始了商业和公共部门领域的数据科学实践。William 热衷于 R，从他搞研究的那天开始就一直使用 R 作为主要的分析工具。他曾经是 Revolution Analytics 的一员，那时候他做了很多使 R 更加适合大数据的贡献。

我要感谢我的共同作者 Aloysius，你的努力、耐心和决心成就了这本书。

审稿人简介

Richard Cotton

拥有蛋白质组学、商账追收和化学健康安全等混合背景的数据科学家。他在工具方面做了大量工作，向非技术用户提供使用统计模型的接口。他是 *Learning R*（O'Reilly）一书的作者，并且创建了很多常用的 R 包，比如 assertive、regex、pathological 和 sig 等。他就职于卡塔尔威尔康乃尔医学院。

Kirill Müller

拥有计算机科学文凭，目前是苏黎世瑞士联邦理工学院的交通规划与系统研究所的助理研究员。他是一名热心的 R 用户，贡献过若干 R 包。

John Silberholz

MIT 操作研究中心在读博士第四年，导师是 Dimitris Bertsimas。他的研究方向主要是基于数据驱动的方法设计晚期癌症的化疗方案以及癌症人群的筛查策略。此外，他的研究兴趣还包括文献计量学和启发式评估领域的分析型应用。John 参与开展了大型在线课程（MOOC）*15.071x: The Analytics Edge*（分析学的优势），讲述利用 R 和电子表格解决机器学习和优化问题。

加入 MIT 之前，John 获得了马里兰大学数学与计算机科学的学士学位，曾在微软和谷歌做过软件开发的实习生，并参与创办了电网可靠性创业公司 Enertaq。

前言

随着数据越来越重要，商家和科学家们也越来越需要高效分析和处理大规模数据的工具。近年来，R 这一工具越来越普遍地应用于数据处理、统计分析和数据科学。虽然 R 最初源自学术界，但如今已经被工业界各个组织广泛使用。

然而，由于 R 本身的设计问题，它能够有效处理的数据规模和计算复杂度有限。这对于需要处理日益增长的大规模数据的 R 用户来说，是个极大的障碍。

《R 高性能编程》这本书有助于理解 R 的性能难题，比如内存和计算方面的限制。本书还给出了很多克服这些难题的技术。你可以根据需求和计算环境选择其中一种技术或者混合使用不同的技术。

这本书是关于如何提升 R 程序性能的实用指南，并适当地解释了使你能够理解每一个解决方案背后的原因。同样，对于本书提到的每一个技术我们都提供了代码示例，以及在我们的机器上生成的性能分析结果以展示性能的提升情况。我们建议大家根据自己的实际环境输入和运行这些代码，然后自己看看性能提升情况。

如果想要知道 R 的设计原理并理解其性能限制的原因，R 内部文档（`http://cran.r-project.org/doc/manuals/r-release/R-ints.html`）中提供了有用的线索。

本书是基于开源 R 编写的，因为开源版本的 R 是最为广泛使用的，而且任何人都可以免费获得。如果你用的是商业版的 R，请跟软件提供商确认一下他们都做了哪些性能优化。

R 社区有很多提升 R 性能的新包，可从 CRAN（`http://cran.r-project.org/`）获取。鉴于 CRAN 有成千上万个包，我们不可能逐一分析，来确定哪个包针对哪个操作提供了性能改善。相反，本书重点关注 R 程序员最常见的任务，介绍那些任何 R 项目都可以使用的技术。

本书的内容

第 1 章 理解 R 的性能：为什么 R 程序有时候很慢？从窥探 R 的底层原理开始，探索 R 程序达到性能极限的几种方式。我们讲述某些时候 R 的设计是如何给 R 程序造成计算（CPU）、内存（RAM）和磁盘输入/输出（I/O）上的性能瓶颈的。

第 2 章 性能分析：衡量代码的性能 介绍衡量 R 代码性能的几个技术，从而理解性能问题的本质。这些技术贯穿全书使用。

第 3 章 加快 R 运行的简单方法 介绍如何提升 R 代码的计算速度。这些基本技术在任何 R 程序中均可使用。

第 4 章 使用编译代码加快运行速度 讨论利用以另一种语言编写的编译代码，比如使用 C 编译的代码，来最大程度优化计算的性能。我们会看到编译代码是怎样加快 R 运行的，以及如何将编译代码集成到 R 程序中。

第 5 章 使用 GPU 让 R 运行得更快 这一章把我们带入现代加速器领域，利用 GPU 高速运行复杂计算。

第 6 章 减少内存使用的简单方法 介绍管理和优化 RAM 使用情况的基本技术，使 R 程序能够处理更大的数据集。

第 7 章 使用有限的内存处理大型数据集 解释怎样使用节约内存的数据结构和磁盘常驻的数据格式，处理超过可用内存大小的数据集。

第 8 章 使用并行计算加倍提升性能 介绍 R 中的并行性。我们会探索如何在单个机器和多个机器上并行地执行 R 代码。此外，还会讨论设计并行代码需要考虑哪些因素。

第 9 章 将数据处理交给数据库系统 描述如何将某些计算转交给外部数据库系统。这有助于将进出数据库的大数据（Big Data）迁移开销最小化，特别是当你已经拥有一个强大的数据库系统，你就可以利用它的计算能力和速度。

第 10 章 R 和大数据 介绍利用大数据技术达到 R 的性能极限，并总结全书。

如果你赶时间，我们推荐优先阅读以下几个章节，然后再根据具体情况补充阅读其他相关章节：

- 第 1 章 理解 R 的性能：为什么 R 程序有时候很慢？
- 第 2 章 性能分析：衡量代码的性能
- 第 3 章 加快 R 运行的简单方法
- 第 6 章 减少内存使用的简单方法

阅读本书需要什么

本书中的所有代码都是在 Mac OS X 10.9 上编写的，使用 64 位的 R 3.1.1 版本。我们还尽可能地在 Ubuntu 14.04 LTS 和 Windows 8.1 上进行了测试。所有代码示例都可以到（`https://github.com/com/r-high-performance-programming/rhpp-2015`）下载。

要使用代码示例，建议安装 R 3.1.1 64 位版本，或者更高版本。

我们还建议在 UNIX 环境（包括 Linux 和 Mac OS X）中运行 R。因为如果在 Windows 上运行 R，有些示例中的包，比如 bigmemory，只能在 Unix 环境下运行。示例代码中凡是 UNIX 和 Windows 环境下存在差异的，我们都会予以说明。

需要使用 64 位的 R，因为有些操作（比如创建一个包含 231 个或更多元素的向量）在 32 位版本上是不可实现的。而且，64 位版本能够尽可能多地利用系统的可用内存，而 32 位版本只能使用至多 4GB 的内存（有些操作系统最多只有 2GB）。

此外，还需要安装一些 R 包。有些章节中的例子需要使用额外的包。

有些章节中的示例需要其他软件和包才能运行。这些在相应章节中都有说明，还给出了它们的安装指南。

如果无法获得示例中需要的那些软件和工具，可以在 Amazon Web Services (AWS)上运行这些示例。例如，第 5 章的例子需要支持 CUDA 的 NVIDIA GPU，第 9 章的例子需要几种数据库系统，第 10 章的例子需要 Hadoop。

使用 Amazon 账号登录（`http://aws.amazon.con/`）就可以使用 AWS。如果还没有账号，就创建一个账号。创建账号是免费的，但是使用服务器、存储和其他资源是收费的。请咨询 AWS 网站了解你所在地区的最新收费情况。

全球不同区域都提供 AWS 服务。截至本书编写之时，AWS 有 8 个服务区域，其中 3 个在美国，1 个在欧洲，3 个在亚太地区，还有 1 个在南美。任选一个区域，比如选择离你最近的或者价格最低的。进入 AWS 控制台（`http://console.aws.amazon.com`），在右上角选择区域。选好区域以后，使用本区域内的 AWS 资源实现书中的示例代码。

在配置任何计算资源之前，比如配置服务器或者 Hadoop 集群，首先需要登入服务器的密钥对。如果还没有 AWS Elastic Cmpute Cloud (EC2)的密钥对，按照以下步骤生成密钥对：

1. 进入 AWS 控制台，单击 EC2。
2. 单击左侧菜单的密钥对。
3. 单击创建密钥对。
4. 输入新密钥对的名字（例如 mykey）。
5. 一旦单击创建以后，私有密钥（例如 mykey.pem）就会下载到本地电脑。

在 Linux 和 Mac OS X 上，修改私有密钥文件的权限，只允许读操作。具体操作是在终端窗口中对 mykey.pem 文件执行 chmod 400。

本书的读者对象

如果你已经是一个 R 程序员，想要寻找提升代码性能的方法，那么这本书适合你。你需要熟悉并舒适地使用 R，但却不需要很高深的技术。阅读本书需要的技能包括：

- 安装、升级和运行 R
- 在 R 环境中安装和升级 CRAN 包
- 创建和操纵基本的数据结构，例如向量、矩阵、列表和数据框
- 使用和转换不同的 R 数据类型
- 执行算术、逻辑和其他基本的 R 操作
- 使用 R 控制语句，例如 if、for、while 和 repeat
- 编写 R 函数
- 使用 R Graphics 画图

如果你是 R 新手，想学习如何编写 R 程序，有很多书籍、在线课程、教程和其他资源

可供参考。用你喜欢的搜索引擎搜一下就可以了。

约定

本书中使用了很多格式的文本，以区分各种不同的信息。这里我们举例说明这些格式，并解释它们的含义。

正文中的代码、数据库表名、文件夹名、文件名、文件扩展名、路径名、URL、用户输入以及 Twitter 用户名的格式是这样的，例如："要编译这个函数，我们在编译包中使用 `cmpfun()` 函数。"

代码块像这样表示：

```
fibonacci_rec \<- function(n) {
    if (n \<= 1) {
        return(n)
    }
    return(fibonacci_rec(n - 1) + fibonacci_rec(n - 2))
}
```

新术语和重要词汇用黑体表示。屏幕上出现的词，例如，菜单或对话框，像这样表示："在安装向导中一定要勾选**'制作安装包'**和**'编辑系统路径'**两个选项。"

警告或重要事项以这种方式表示。

技巧和提示像这样表示。

下载示例代码

你可以从 http://www.broadview.com.cn 的"下载专区"，下载所有已购买的博文视点书籍的示例代码文件。

勘误表

虽然我们已经尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。你可以随时浏览图书页面，查看已发布的勘误信息。

目录

1

理解 R 的性能：为什么 R 程序有时候很慢？

R 是用于统计分析和数据处理的强大工具，开发于 1993 年，最初是作为数据分析课程的授课工具。由于使用简单，在其后的 20 年，R 变得越来越受欢迎，不止是在学术界，在政府和工业界也得到广泛使用。而且，R 是开源工具，所以用户可以免费使用，并能够通过一个称为 Comprehensive R Archive Network(CRAN)的公共库共享新的统计包。随着 CRAN 库日渐壮大，编写本书的时候，CRAN 已经有超过 6000 个文档完善的包可供使用，这进一步增强了 R 的吸引力。这 20 年来，组织和个人创建、传输、存储和分析的数据规模也在指数型增长。要处理和分析这些日益增长的数据，程序员有时会发现 R 不堪重负。为什么有时候 R 程序运行不良？如何克服这些性能限制？本书讲述了 R 性能背后的因素，并介绍了很多提升 R 程序性能的技术，例如优化内存使用，执行并行计算，甚至利用外部数据处理系统的计算能力等。

在找到 R 性能问题的解决方案之前，我们需要先弄清，是什么导致 R 的性能在某些情况下变差的。本章通过理解 R 底层的设计原理，以及这些设计怎样限制了 R 程序的性能，来开启我们探索 R 高性能编程之旅。

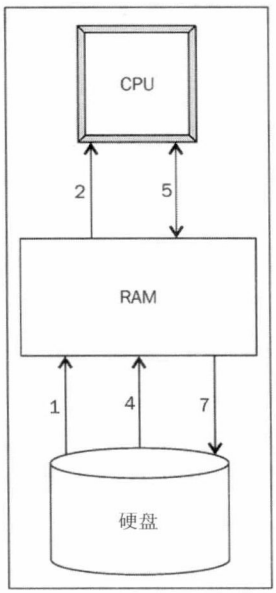
我们将讨论任何计算任务都会面临的三个主要限制，即 CPU、RAM 和磁盘输入/输出（I/O），并介绍这些限制在 R 程序中的具体体现，最后对 R 程序可能会遇到的瓶颈给出一些见解。

本章包含以下话题：

- 计算性能的三个限制因素——CPU、RAM 和磁盘 I/O
- R 是运行时解释的
- R 是单线程的
- R 需要将全部数据加载到内存
- 算法设计影响时间和空间复杂度

计算性能的三个限制因素：CPU、RAM 和磁盘 I/O

首先，我们看一下 R 程序在电脑中是如何运行的。这里提供的是一个简化版本，不过足以用来理解 R 的性能限制了。下面这幅图展示了执行 R 程序需要的步骤。



R 程序的运行步骤

例如，这个简单的 R 程序从 CSV 文件装载数据、计算列的和，然后将结果写入另一个 CSV 文件：

```
data <- read.csv("mydata.csv")
totals <- colSums(data)
write.csv(totals, "totals.csv")
```