CRC Press
Taylor & Francis Group

R语言应用系列

新视野·数据科学

数据科学中的并行计算：以R，C++和CUDA为例

Parallel Computing for Data Science with Examples in R,C++ and CUDA

[美] 诺曼·马特洛夫 著
汪 磊 寇 强 译

Norman Matloff

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

CRC Press
Taylor & Francis Group

R语言应用系列

新视野·数据科学

数据科学中的并行计算：以R，C++和CUDA为例

Parallel Computing for Data Science with Examples in R, C++ and CUDA

[美] 诺曼·马特洛夫 著
汪嘉寇强 译

Norman Matloff

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

Parallel Computing for Data Science with Examples in R,C++ and CUDA

Norman Matloff

ISBN: 978-1-4665-8701-4

图书在版编目（CIP）数据

数据科学中的并行计算：以 R，C++和 CUDA 为例/
〔美〕诺曼·马特洛夫（Norman Matloff）著；汪磊，寇强译.
—西安：西安交通大学出版社，2017.9
书名原文：Parallel Computing for Data Science：with Examples in R,C++ and CUDA
ISBN 978-7-5605-9958-8
I. ①数… II. ①诺… ②汪… ③寇… III. ①并行算法-研究
IV. ①TP301.6
中国版本图书馆 CIP 数据核字(2017)第 196220 号

书　　名　数据科学中的并行计算：以 R，C++和 CUDA 为例
著　　者　〔美〕诺曼·马特洛夫
译　　者　汪磊　寇强

出版发行　西安交通大学出版社
（西安市兴庆南路 10 号　邮政编码 710049）
网　　址　http://www.xjtupress.com
电　　话　(029)82668357　82667874(发行中心)
(029)82668315 (总编办)
传　　真　(029)82669097
印　　刷　陕西宝石兰印务有限责任公司

开　　本　720 mm×1000 mm　1/16　印 张 20.75
印　　数　0001~2000 册　字 数 362 千字
版次印次　2017 年 12 月第 1 版　2017 年 12 月第 1 次印刷
书　　号　ISBN 978-7-5605-9958-8
定　　价　72.00 元

读者购书、书店添货如发现印装质量问题，请与本社发行中心联系、调换。
订购热线：（029）82665248　（029）82665249
投稿热线：（029）82665397
读者信箱：banquan1809@126.com

译者序

21 世纪的第二个十年，随着计算能力的巨大提升和移动互联网的迅猛发展，大数据时代拉开了它的帷幕。大数据时代的显著特点就是数据量大，对数据处理的速度和时效提出了苛刻的要求。

在传统的串行计算下，多核的计算机/集群等只有一个内核能够进行有效的工作，这就造成了计算性能的浪费。并行计算概念的提出，则解决了这个性能浪费的问题。它能够协调多个内核共同计算，极大地提升了计算速度，从而满足了大数据时代人们对高速处理数据的需求。

Norman Matloff 教授在加州大学戴维斯分校教授计算机科学，对计算机架构和算法了然于心。更值得一提的是，他还是该校统计系的创始人之一，不但教授本科的统计课程，还在统计系硕士和博士的考试委员会担任多个职务。对于统计理论的熟悉，使得他在使用计算机编程处理统计问题的时候，更加得心应手。该书即是他在并行计算方向上多年经验的总结。

本书不是一本并行计算的理论教材。该书别出心裁，使用实例手把手地教会读者掌握并行计算的基本概念和操作。在提纲挈领地介绍了如何在 R 中使用并行方法之后，作者带领我们学习了多线程和多进程，以及并行调度等方面的知识和技能。随后，作者用详尽的篇幅讲述了如何使用 R、C++ 和 CUDA 分别来进行共享内存范式编程和消息传递范式编程。本书在讲述了当前流行的 MapReduce 之后，又详细讲解了如何并行地实现串行计算下所对应的排序、扫描、矩阵乘法等经典算法。在本书的最后，作者讲述了如何使用并行计算来进行统计。此外，本书的附录中对线性代数、R 和 C 也做了简明的介绍，方便不熟悉的读者迅速入门。

值得一提的是，Matloff 教授的汉语也非常熟练，在本书的翻译过程中，他也给出了相应的建议和意见。编辑李颖为本书的编辑工作提出了不少中肯的建议和意见，并为本书的顺利出版做出了巨大的努力。在这里对他们一并表示感谢。

本书两位译者的协作，跨越了大洋和时差。翻译的日子中酸甜苦乐，都化作段子互相慰藉。不禁让人想到，人的命运啊，当然要靠自我奋斗，但是也要考虑到历史的行程。时代带给我们的，永远值得珍惜。

译者于北京
2017 年 8 月

前言

感谢你对本书感兴趣。我很享受写书的过程，也希望这本书对你非常有用。为达此目的，这里有几点事情我希望说清楚。

本书目标：

我很希望这本书能充分体现它标题的含义——*数据科学中的并行计算*。和我所知道的其他并行计算的书籍不同，这本书里你不会碰到任何一个求解偏微分方程或其他物理学上的应用。这本书真的是为了数据科学所写——无论你怎样定义数据科学，是统计学、数据挖掘、机器学习、模式识别、数据分析或其他的内容[1]。

这不仅仅意味着书里的实例包括了从数据科学领域中选取的应用，这也意味着能够反映这一主题的数据结构、算法和其他内容。从经典的“n 个观测，p 个变量”的矩阵形式，到时间序列，到网络图模型和其他数据科学中常见的结构都会囊括其中。

本书包含了大量实例，以用于强调普遍的原理。因此，在第 1 章介绍了入门的代码实例后（没有配套的实例，这些普遍的原理也就没有任何意义），我决定在第 2 章里解释可以影响并行计算速度的一般因素，而不是集中介绍如何写并行代码。这是一个至关重要的章节，在后续的章节中会经常提到它。事实上，你可以把整本书看成如何解决第 2 章开头所描述的那个可怜的家伙的困境：

> 这里有一个很常见的情景：一个分析师拿到了一台崭新的多核机器，这台机器能做各种神奇的事情。带着激动的心情，他在这台新机器上写代码求解他最

[1] 比较讽刺的是，我个人并不是很喜欢数据科学这个术语，但它的确可以包含很多不同的观点，这个词也可以强调本书是关于数据的，而不是物理学问题。

> 喜欢的大规模的问题，却发现并行版本的运行速度比串行的还慢。太令人失望了！现在让我来看看究竟什么因素导致了这种情形……

本书标题里的计算一词反映了本书的重点真的是在计算上。这和诸如以 Hadoop 为代表的分布式文件存储等的并行数据处理不同，尽管我还是为相关话题专门写了一个章节。

本书主要涵盖的计算平台是多核平台、集群和 GPU。另外，对 Thrust 也有相当程度的介绍。Thrust 极大地简化了在多核机器和 GPU 上的编程任务，并且同样的代码在两种平台上都可以运行。我相信读者会发现这部分材料非常有价值。

需要指出一点，这本书不是一本用户手册。尽管书中使用了诸如 R 的 **parallel** 和 **Rmpi** 扩展包、OpenMP、CUDA 等特定工具，但这么做仅仅是为了让问题具体化。本书会给读者带来有关这些工具的非常扎实的入门介绍，但不会提供诸如不同函数的参数、环境选项等内容。本书的目的是，希望读者阅读完本书后，为进一步学习这些工具打下良好基础，更重要的是，读者今后可以使用多种语言编写高效的并行代码，无论是 Python、Julia，还是任何其他语言。

必要的背景知识：

如果你认为你已经可以相对熟练地使用 R，那本书的大多数内容你应该都可以读懂。在一些章节里，我们需要使用 C/C++，如果你想仔细阅读学习相关章节，需要具备相关的背景知识。然而，即使你不怎么了解 C/C++，你也应该会发现这些章节很容易读懂，并且相当有价值。附录里包括了针对 C 程序员的 R 简介和针对 R 用户的 C 语言简介。

你需要熟悉基础的矩阵运算，主要是相乘和相加。有时我们也会使用一些更高级的运算，比如求逆（以及与之相关的 QR 分解）和对角化。这些内容在附录 A 中有涉及。

机器设备：

除了特别说明的地方，本书中所有的计时实例都运行在一台 16 核允许两个超线程的 Ubuntu 机器上。我一般使用 2 到 24 个核，这应该和多数读者可以使用的平台类似。希望读者可以使用 4 到 16 核的多核系统，或者一个有几

十个节点的集群。但即使你只有一个双核机器，应该仍会发现本书的材料非常有用。

对于那些少数有幸可以使用拥有几千个内核的集群的读者，书中的内容仍然适用。依据本书中对这种系统的观点，那个著名问题“这可扩展吗？”的答案一般是否定的。

CRAN 扩展包和代码：

本书使用了我在 CRAN，即 R 的软件贡献库（`http://cran.r-project.org`）上的几个扩展包：**Rdsm**、**partools** 和 **matpow**。

本书的示例代码都可以从作者的网站下载，`http://heather.cs.ucdavis.edu/pardatasci.html`。

关于字体的说明：

函数名和变量名，还有 R 扩展包（不包括其他语言的扩展包）使用粗体。代码片段使用 LaTeX 中的 **lstlisting**，并根据 R 或 C 语言有所改动。数学量使用斜体[2]。

致谢：

感谢所有为本书提供直接或间接有用信息的各位人士。按照姓氏字母排序，他们有 JJ Allaire、Stuart Ambler、Matt Butner、Federico De Giuli、Matt Dowle、Dirk Eddelbuettel、David Giles、Stuart Hansen、Richard Heiberger、Bill Hsu、Michael Kane、Sameer Khan、Bryan Lewis、Mikel McDaniel、Richard Minner、George Ostrouchov、Drew Schmidt、Lars Seeman、Marc Sosnick 和 Johan Wikström。我非常感谢 Hsu 教授为我提供了 GPU 的高级设备，也非常感谢 Hao Chen 教授允许我使用他的多核系统。

在 Michael Kane 和 Jay Emerson 开发 **bigmemory** 扩展包的同时，我开始了 **Rdsm** 扩展包的开发，本书里部分章节也使用了它。自从我们发现了彼此的工作之后，开始了非常有价值的邮件交流。事实上，在我新版本的扩展

[2]原书中排版前后多处不一致，现已根据我国习惯进行了统一调整。正文与程序段有相同的变量时，字体已作统一。——译者注

包中，之所以决定使用 **bigmemory** 作为底层之一，是因为它可以使用外部存储。在和 Michael 以及 Bryan Lewis 后来的对话中，我也获益良多。

这里非常感谢内部的审阅者 David Giles、Mike Hannon 和 Michael Kane。我要特别感谢我的老朋友 Mike Hannon，他提供了非常详尽的反馈。非常感谢 John Kimmel，Chapman and Hall 出版社的统计主编，他从一开始就非常支持我。

我的妻子 Gamis 和女儿 Laura 那感染性的幽默感和对生活的热情，极大地帮助了我的所有工作。

作者简介

Matloff 博士出生于洛杉矶，在东洛杉矶和圣盖博谷两个地方长大。他在加州大学洛杉矶分校获得了纯粹数学的博士学位，学术研究方向为概率论和统计。他在计算机科学和统计学方向发表了大量论文，现在的研究方向是并行处理、统计计算和回归方法。他也是 *Journal of Statistical Software* 的编委之一。

Matloff 教授曾是国际信息处理联合会 11.3 工作组的成员，该组织是联合国教科文组织（UNESCO）下设的一个数据库软件安全国际委员会。他也是加州大学戴维斯分校统计系的创始人之一，并参与了该校计算机科学系的建立。他在戴维斯分校被授予了杰出教学奖和杰出公众服务奖。

目 录

第 1 章 R 语言中的并行处理入门

抛开并行处理的理论总结，我们先从一个 R 语言的具体示例开始。理论总结的部分可以放在后面。但我们需要先把 R 语言放到一个恰当的语境中。

1.1 反复出现的主题：良好并行所具有的标准

本书大部分的示例都涉及到 R 编程语言，这是一种解释性语言。R 的核心操作都在语言内部进行了高效的实现，因而只要正确使用，R 语言一般能够提供较好的性能。

1.1.1 足够快

在亟需最大化执行速度的情景下，你可能希望使用编译语言，比如 C/C++，我们在本书中也偶尔这样做。但是，使用这些编译语言所得到的速度上的提升，相比付出的精力而言，有些不值。换句话说，我们有和 Pretty Good Privacy 安全系统[1]的一个类比：

良好并行所具有的标准：

> 在大多数情形下，“比较快”就足够了。从 R 语言换到 C/C++ 语言，可能使得编程、调试、维护所花费的时间变得非常长，因而把运行速度的提升作为更换语言的理由，显得不那么充分。

这就是各种并行 R 扩展包流行起来的原因。它们满足了编写并行代码，但仍然使用 R 的需求。例如，**Rmpi** 包提供了将 R 与消息传递接口（message

[1] Pretty Good Privacy，缩写为 PGP，是一个基于 RSA 公钥加密体系的加密软件。——译者注

passing interface, MPI）连接起来的功能。MPI 是一个被大范围使用的并行进程系统，通常 MPI 应用程序是用 C/C++ 和 FORTRAN [2] 所开发的。**Rmpi** 包可以让分析师在 R 语言中使用 MPI。**Rmpi** 包实际上是调用了 MPI，除此之外，R 用户也可以用 C/C++ 编写自己的应用代码，调用 MPI 函数，然后在 R 里面调用生成的 C/C++ 函数。要是这么做，用户就用不到前面提到的编程便捷性以及 R 中丰富的包。因此，最好的选择是通过 **Rmpi** 接口使用 MPI，而不是直接使用 C/C++。

本书的目标是给数据科学中的并行处理提供一个一般性的介绍。R 为数据与统计运算提供了一个强大、高级的丰富集合，这使得使用 R 语言编写的示例通常要比使用其他语言编写的示例更加简短。这样读者可以更加关注并行计算的方法本身，而不是被复杂的嵌套循环等其他细节分散注意力。这不仅在学习并行计算这个角度上很有用，而且当读者把这里所展示的代码和技术移植到 Python 和 Julia 等其他语言时，也会比较简单。

1.1.2 “R+X”

现在 R 中有个很主流的趋势叫做 “R+X”，其中，“X” 指的是一些别的语言或函数库。R+C 早在 R 的初期就很常见了，人们用 R 语言写主要的代码，用 C 或 C++ 写那些对速度有要求的部分代码。现在，X 可能是 Python、Julia、Hadoop、H_2O 或者其他很多东西。

1.2 关于机器

本书将会描述三种类型的机器：多核系统、集群以及图形处理单元（graphics processing unit, GPU）。就像在前言中所述，本书的目标读者不包括那些能够操作超级计算机的少数幸运儿（尽管这里所述的方法也适用于这些机器）。相反，本书假设大多数读者都能操作稍微普通一些的系统，比如 4 ~ 16 核的多核系统，或者有几十个节点的集群，或者一块并非最新版的 GPU。

本书的大多数多核的示例都在一个 32 核的系统上运行过，在这个系统上我几乎没用过全部的内核（因为我的账户只是一个来宾帐户）。通常我都是由较小数目的核来开始计时的试验的，比如 2 到 4 个内核。

对于集群来说，我的 “消息传递” 软件通常都是在多核系统上运行的，有时也会在真的集群上运行，来演示一下开销的影响。

[2] 为了简洁，后面将不再提及 FORTRAN，因为它在数据科学领域用的不多。