Springer

R语言应用系列

新视野·数据科学

Rcpp: R与C++的无缝整合

Seamless R and C++ Integration with Rcpp

[法] 德克·埃德比特尔 著

寇强 张晔 译

Dirk Eddelbuettel

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

R语言应用系列

Seamless R and C++ Integration with Rcpp

Rcpp:R与C++的无缝整合

[法]德克·埃德比特尔 著
Dirk Eddelbuettel

寇 强 张 晔 译

西安交通大学出版社
Xi'an Jiaotong University Press

陕西省版权局著作权合同登记号　图字 25-2014-077 号

图书在版编目(CIP)数据

Rcpp:R 与 C++的无缝整合 /(法)埃德比特尔著;寇强,张晔译. —西安:西安交通大学出版社,2015.12
(R 语言应用系列)
书名原文:Seamless R and C++ Integration with Rcpp
ISBN 978-7-5605-8110-1

Ⅰ.①R… Ⅱ.①埃… ②寇… ③张… 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 273528 号

书　　名	Rcpp:R 与 C++的无缝整合
著　　者	[法]德克·埃德比特尔
译　　者	寇　强　张　晔
责任编辑	李　颖
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)82668280
印　　刷	陕西宝石兰印务有限责任公司
开　　本	720mm×1000 mm　1/16　印　张　17
印　　数	0001～2500 册　字　数　264 千字
版次印次	2015 年 12 月第 1 版　2015 年 12 月第 1 次印刷
书　　号	ISBN 978-7-5605-8110-1/TP·706
定　　价	55.00 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

中译本序

R 语言是一门主要用于数据处理、统计分析和可视化作图的解释型脚本语言。作为一门编程语言，R（及其“前身”S 语言）在设计之初就面临一个二选一的难题：语言的设计是应该面向用户，让使用者可以快速地建模，还是应该面向机器，以使得代码可以高速地在计算机上运行？最终，语言的设计者们选择了前者，其理念是“人的时间”比“机器的时间”更为宝贵。在 R 语言诞生后的十几年间，事实证明这个最初的决定使得 R 逐渐发展为一门具有高度灵活性和可扩展性的统计编程语言，进而极大地促进了其背后 R 语言社区的发展壮大。

然而，语言的简洁性和灵活性并非恒久不变的法则。随着统计模型越来越复杂，数据量越来越大，众多的 R 语言开发者和使用者开始发现效率成为了这门语言的一个瓶颈。“人的时间”固然宝贵，但“人等待机器的时间”同样不可忽视。如何在保持语法不变的同时提升程序执行的效率，成为了 R 语言开发者们一个十分关注的话题。

事实上，在 R 语言诞生的初期，其核心开发团队就给出了一个解决方案：将计算密集的算法用 C/C++ 实现，然后在 R 中调用这部分代码。R 语言提供了一系列的 API（应用程序接口）来实现它与其他语言的交互，但在很长的一段时间里，积极使用这些接口的 R 软件包开发者并不占多数，其中可能最重要的一个原因就是这些接口的使用相对繁琐，且文档资料也不够丰富，开发者空有屠龙之刀，却无屠龙之技。

幸运的是，这一局面在 **Rcpp** 横空出世后被彻底打破。我第一次听说 **Rcpp** 是在 2009 年，当时在统计之都论坛的帖子上（`http://cos.name/cn/topic/17665/`）大家在讨论如何用 R 调用 C++ 程序，于是经过一些搜索后我从 R 的软件仓库中找到了这个软件包。当时的 **Rcpp** 核心只有两个文件，代码总量不到 2000 行，但那时它已经可以极大地简化 R 与 C++ 之间的交互。现如今，**Rcpp** 的代码量已经接近 10 万行，在 R 的官方软件包仓库中有超过 300 个软

件包直接依赖于 **Rcpp**，而它也成为了被依赖次数最多的 R 语言扩展包（除去 R 自身默认提供的扩展包），没有之一。

总的来说，**Rcpp** 定义了一系列的类、函数和接口来增强 R 与 C++ 之间的交互性。用户只需懂得基本的 C++ 知识，就可以写出丰富的可供 R 调用的 C++ 程序。与 R 中传统的 C 语言 API 相比，**Rcpp** 利用了更为现代的 C++ 编程技术，故而其语法更为简洁，也更富表现力和可读性。此外，**Rcpp** 还特意针对 R 软件包开发提供了一系列便捷的辅助程序，使得开发者可以快速地部署项目，开发软件包，省去了许多繁琐而枯燥的设置。或许，这正是 **Rcpp** 能迅速地获得 R 软件包开发者青睐的原因。

本书的原作者，Dirk Eddelbuettel，正是 **Rcpp** 从最早到现在开发工作的主导者。从这个角度来说，由作者自己来阐述 **Rcpp** 的设计理念和使用方法是最为恰当不过的了。而更为可贵的是，作者在全书中使用了大量的实例和代码来讲解 **Rcpp** 的细节，可以预想，读者无论是在理念上还是在实战中都能从本书中受益。

本书的两位译者为本书中文版的面世付出了大量的时间和心血。需要特别提到的是，两位译者同样也是 R 社区活跃的开发者，他们在许多 R 软件包和编程项目中都大量使用了 **Rcpp**。也正是因为如此，两位译者在执笔过程中融入了自己使用 **Rcpp** 的心得和体会，在语言上将原本可能艰涩的编程概念用更加平易近人的方式表达出来，相信读者在阅读本书的过程中会体会到译者的用心。

邱怡轩

2015 年 3 月于普渡大学

译者序

在此致谢在本书翻译过程中给予帮助的各位朋友：华南统计科学研究中心的刘成烽、吴炳培、朱珊、黄巩怡、夏凡、田媛、朱俊贤、潘文亮。特别鸣谢潘文亮博士，感谢他对于本书统计学相关的部分给予的帮助。感谢普渡大学的邱怡轩，他为本书写了精彩的序言。

R 的世界精彩纷呈，C++ 的世界博大精深。Dirk 的 **Rcpp** 成为了沟通两个世界的桥梁。希望这本书能成为一个窗口，求实用者得见 **Rcpp** 提高开发效率、计算速度的妙用，求道者得窥 **Rcpp** 融和 R 与 C++ 的奥妙。

张晔

2015 年 11 月于北京

前言

Rcpp 是一个 R 语言扩展包，可以使用 C++ 函数对 R 进行扩展。

从使用 C++ 替代等价的 R 函数或开发新函数来快速构建插件，从而进行更流畅的实验和运算加速，到对已有的库进行大规模的整合，或者作为一个全新研究计算环境的基础，**Rcpp** 在很多方面都在被使用着。

尽管 **Rcpp** 还是一个相对新的项目，其已经在 R 社区的用户和开发者中被广泛使用。**Rcpp** 现在是 R 系统中最流行的语言扩展，被 CRAN 上超过 100 个扩展包和 BioConductor 上数十个扩展包使用。

本书的目标在于为 **Rcpp** 提供一份可靠的介绍。

目标读者

本书适用于希望使用 C++ 代码对 R 进行扩展的 R 用户。熟悉 R 语言对于阅读本书自然很有帮助；有很多其他书籍提供了回顾和特定的介绍。C++ 的知识也很有帮助，尽管我们不严格要求。附录为只熟悉 R 语言的读者提供了一个非常简短的 C++ 简介。

本书对于拥有更多 C++ 编程背景而开始使用 R 的用户也非常有用。然而，可能需要进行一些额外的背景阅读来对 R 本身有一个更好的认识。Chambers 的书 (Chambers, 2008) 提供了一个对 R 系统背后思想的良好介绍，这对于希望进一步了解 R 的读者是很有帮助的资料。

有一些读者可能会想了解 **Rcpp** 内部的工作原理。然而，考虑到这需要相当多的 C++ 知识，这不是本书的目的所在。本书仅仅着重于如何使用 **Rcpp**。

历史内容

Rcpp 最早在 2005 年出现。当时作为 Dominick Samperi 对 Eddelbuettel 从 2002 年开始开发的 **RQuantLib** 扩展包 (Eddelbuettel and Nguyen, 2014) 提交（和现在的规模相比不大）的一个补充出现。**Rcpp** 在 2006 年早些时候有了自己的名字，并成为一个 CRAN 扩展包。在 **Rcpp** 这个名字下，进行了一系列快速的发布（全部由 Samperi 进行）。之后这个扩展包更名为 **RcppTemplate**；并在 2006 年以新名字发布了一系列版本。但在 2007 年、2008 年和 2009 年的大部分时间内没有发布新版本。在 2009 年晚些时候的一些更新后，**RcppTemplate** 由于缺乏积极维护进入 CRAN 存档。

为了继续使用这个扩展包，Eddelbuettel 决定对其进行重新开发。从 2008 年 11 月起以原本的名字 **Rcpp** 发布新版本。这涵盖了改善的构建发布过程、增加的文档、新功能等内容，同时保存已有的 "经典 **Rcpp**" 接口。尽管本书没有涉及，这些 API 会继续通过 **RcppClassic** 扩展包 (Eddelbuettel and François, 2012c) 的形式存在和提供支持。

为了反映 C++ 代码标准的改进 (Meyers, 2005)，Eddelbuettel 和 François 从 2009 年开始了重新设计。这次改进添加了很多新特性，其中很多都在扩展包中不同的说明文档中进行了描述。经过重新设计的 **Rcpp** (Eddelbuettel and François, 2012a) 被广泛地使用，到 2012 年 11 月，CRAN 上有超过 90 个扩展包依赖于 **Rcpp**。本书描述的也正是这个版本。

Rcpp 在继续保持着积极的开发，新的扩展也正在持续添加。我们会确保本书中所描述的内容会一直是可行的，并提供支持。

相关工作

很多人都尝试进行过 C++ 和 R 的整合工作；最早的出版文献可能来自 Bates 和 DebRoy (Bates and DebRoy, 2001)。*Writing R Extensions* 手册 (R Development Core Team, 2012d) 也大约从那时起开始提到 C++ 和 R 的整合。Java 等人一篇未公开出版的论文 (Java et al., 2007) 中表达了和我们一些方法很接近的想法，但没有完善充实。**Rserve** 扩展包 (Urbanek, 2003, 2012) 可以作为 R 的一个 socket 服务器。在服务器端，**Rserve** 将 R 的数据结构转换为一个二进制序列化格式，并使用 TCP/IP 进行传递。在客户端，对象被重构为模仿 R 对象结构的 Java 或 C++ 类实例。

rcppbind (Liang, 2008)、**RAbstraction** (Armstrong, 2009a) 和 **RObjects** (Armstrong, 2009b) 扩展包都使用 C++ 模板进行实现，但没有一个成熟到在 CRAN 上进行发布。**CXXR** (Runnalls, 2011) 从另一个方向来解决这个问题：其目标在于在一个更强大的 C++ 基础上完全重构 R。因此 **CXXR** 要考虑 R 解析器的方方面面、Read-Eval-Print Loop（REPL）[①] 和线程问题；R 和 C++ 之间的对象交换只是其中一部分。Temple Lang 也曾经讨论一个类似的方法 (Temple Lang, 2009a)，其建议由扩展包开发来对底层内核进行扩展，从而辅助 R 的扩展。Temple Lang 提出使用编译器输出作为代码参考，从而添加语言结合和封装器 (Temple Lang, 2009b)，这也提供了一个有些不同的观点。最后，**rdyncall** 扩展包 (Adler, 2012) 提供了一个与现有 R 到 C 接口不同的一个直接接口。这可能会让想直接获取底层编程接口的 R 程序员感兴趣。然而，这和我们在 **Rcpp** 中所着重介绍的 C++ 接口所处的对象层面的交换不同。

排版说明

在排版上，我们遵守了出版社和 *Journal of Statistical Software* 的惯例。

- 编程语言使用 Sans-serif 字体，如 R 或 C++；
- CRAN 所提供的或其他扩展包名称使用粗体，如 **Rcpp** 或 **inline**；
- 简短代码或变量使用 `Courier` 字体，比如 `x <- y + z`。

在正文中，我们使用了一个定制的环境来展示源代码。

Dirk Eddelbuettel
River Forest, IL, USA

[①] Read-Eval-Print Loop，简称 REPL，是一个简单的交互式编程环境。常常用于指代一个 Lisp 交互式开发环境，R 从 scheme 继承而来，本质也是门函数式语言。——译者注

致 谢

Rcpp 是众多合作者的共同成果，这里按顺序表达感谢。Dominick Samperi 开发了最初的代码，尽管比起现在的 **Rcpp**，其所覆盖的范畴有限得多，但其指明了使用 C++ 模板在 R 和 C++ 之间进行类型转换的正确方向。

Romain François 设计和实现了当前 **Rcpp** 中很大部分，展示了无可挑剔的品位。现行 **Rcpp** 设计所带来的强大功能很大一部分归功于其工作和无穷的精力。module 和 sugar 的关键部分，以及 "magic" 模板的很大一部分都是他的贡献。最早这只是作为我们 **RProtoBuf** 扩展包的一部分，让对象之间的交互更容易，但这让我们踏上了完全不同、非常让人兴奋的旅程。和 Romain 工作是莫大的愉悦，我现在依然感激他的工作，我也很期待 **Rcpp** 的进一步发展。

Doug Bates 自从一开始就提供了莫大的帮助：如果不是其为解析 `SEXP` 类型中列表成员的简单的宏，我们可能从来不会在十年前开始 **RQuantLib** 的开发。Doug 后来加入了这个项目，在关于 **Rcpp** 和 **RcppArmadillo** 的关键决定上提供了意见，并且负责了 **RcppEigen** 项目。

John Chambers 在 Rcpp module 开始时成为了很关键的支持者，并且在 R 和 C++ 之间进行交换上贡献了很多重要的代码。当我和 Romain 从 John 那里得知，**Rcpp** 和 20 世纪 70 年代贝尔实验室手稿中和系统间整体对象交换的最初设计观点如此接近时，这是对我们莫大的称赞。

JJ Allaire 一直以来都是 **Rcpp** 重要的贡献者，同时也是在 R 和 C++ 之间进行近乎自然匹配想法的重要支持者。其所贡献的 Rcpp attributes 正展现着巨大的潜力，我们很期待在其之上构建很棒的东西。

R 核心团队的众多成员，包括 Kurt Hornik、Uwe Ligges、Martyn Plummer、Brian Ripley、Luke Tierney 和 Simon Urbanek，在从编译和可移植性到 R 内核等众多问题上提供了帮助。最后，也非常重要的是，如果没有 R 系统，自然就不会有在其之上构建的 **Rcpp**。

最后，R 和 **Rcpp** 社区的很多成员在不同的论坛、会议演讲以及通过邮件列表对 **Rcpp** 的发展给予支持，同时带来了很多非常好的问题和建议。正是 **Rcpp** 被如此活跃地使用激励着我们，让我们和 **Rcpp** 继续向前发展。

目 录