HZ BOOKS 华章IT

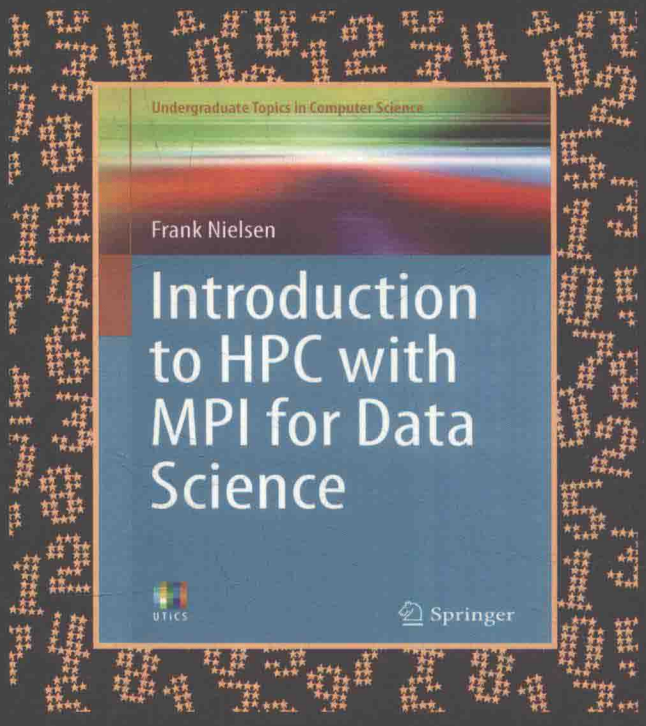
Springer

数据科学与工程技术丛书

基于MPI的大数据高性能计算导论

[法] 弗兰克·尼尔森（Frank Nielsen）著

张伟哲 等译



INTRODUCTION TO HPC WITH MPI FOR DATA SCIENCE

机械工业出版社
China Machine Press

数据科学与工程技术丛书

INTRODUCTION TO HPC WITH MPI FOR DATA SCIENCE

基于MPI的大数据高性能计算导论

[法] 弗兰克·尼尔森（Frank Nielsen）著

张伟哲 郝萌 鲁刚钊 王德胜 孙博文 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

基于 MPI 的大数据高性能计算导论 /（法）弗兰克·尼尔森（Frank Nielsen）著；张伟哲等译 .
—北京：机械工业出版社，2018.6
（数据科学与工程技术丛书）
书名原文：Introduction to HPC with MPI for Data Science

ISBN 978-7-111-60214-9

I. 基… II. ① 弗… ② 张… III. 数据处理 IV. TP274

中国版本图书馆 CIP 数据核字（2018）第 133210 号

本书版权登记号：图字 01-2017-4121

本书使用 MPI 标准介绍了数据科学中的高性能计算，帮助读者了解分布式存储模型中的并行编程的知识。全书分为两部分，第一部分（第 1～6 章）基于消息传递接口介绍高性能计算，内容包括：阻塞与非阻塞的点对点通信、死锁、全局通信函数（广播、散播等）、协同计算（归约）的基本概念；互联网络的拓扑结构（环、环面和超立方体）以及相应的全局通信程序；基于分布式内存的并行排序及其实现，涵盖相关并行线性代数知识；MapReduce 模型。第二部分（第 7～11 章）介绍计算机集群中的高性能数据分析，内容包括：数据聚类技术（平面划分聚类、层次聚类）；基于 k-NN 的有监督分类；核心集以及相关降维技术；图算法（最稠密子图、图同构检测）。每章章末附有各种难度的练习和参考文献，可供读者进行自测和深入学习。

本书适合作为“高性能计算”相关课程的本科生教材。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：唐晓琳　　责任校对：殷　虹
印　　刷：北京瑞德印刷有限公司　　版　　次：2018 年 7 月第 1 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：14.5
书　　号：ISBN 978-7-111-60214-9　　定　　价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

译 者 序

数据科学(Data Science，DS)的目的是研究数据本身，研究数据的各种类型、状态、属性及变化形式和变化规律。随着当前数据量的指数级增长，数据科学的战略意义已经不在于掌握“大数据”(big data)信息本身，而在于对这些含有意义的数据进行快速、准确、高效地分析处理。

高性能计算(High Performance Computing，HPC)是辅助“大数据”快速分析与处理的利器之一。通过高性能网络互连的很多处理器或者某一集群中的大量计算机，对“大数据”进行并行聚类、分类以及降维等操作，从而能够更加快速地获取数据中蕴含的规律和有价值的信息。

本书的主要目的是将数据科学与高性能计算两大研究分支交叉融合起来，让高性能计算更好地为数据科学服务，让数据科学对高性能计算发展提出新的需求。因此，本书分为两个部分：第一部分，介绍了高性能计算的基本概念及基于标准消息传递接口(Message Passing Interface，MPI)的并行编程方法；第二部分，给出了高性能计算在数据科学中的应用，尤其重点阐述了如何利用 MPI 来并行化数据科学中的算法。

本书由哈尔滨工业大学计算机科学与技术学院张伟哲教授对全书统稿和校对，其中第 1 章、第 4 章、第 5 章、第 11 章和附录由哈尔滨工业大学鲁刚钊翻译，第 2 章由哈尔滨工业大学王德胜翻译，第 3 章由哈尔滨工业大学孙博文翻译，第 6～10 章由哈尔滨工业大学郝萌翻译。同时，我们的工作还得到哈尔滨工业大学梁智颖、任雪健的帮助，在此一并表示感谢。

还要特别感谢机械工业出版社的编辑唐晓琳、王春华在本书的编辑、出版方面所付出的辛勤劳动。

我们希望本书的出版对相关科技人员和读者所有帮助，同时也期望广大专家和读者提出宝贵意见。

前　言

欢迎来到高性能计算的世界！欢迎来到高性能数据科学的世界！

在本书中，我们将介绍面向**数据科学**(Data Science，DS)的**高性能计算**(High Performance Computing，HPC)。因此，本书主要分为两个部分：第一部分(前 6 章)涵盖 HPC 的基本原理；第二部分(后 5 章)介绍了数据科学的基本知识，并展示了如何编写面向基本串行算法的分布式程序，以应对大规模数据集。当前，许多大规模数据集都是公开的，这些数据集中蕴含了丰富的信息，但是这些信息需要通过精心设计才能被提取出来。

我们主要区分两种并行算法的设计方法：在单个共享内存多核机器上使用多线程并行化算法；在分布式内存**集群系统**上并行化算法。

一方面，当在共享内存架构(如智能手机、平板电脑，以及智能手表和其他物联网设备)上设计并行化算法时，所有的硬件计算单元(核)位于同一芯片上，我们可以使用多线程来轻松地对视频解码、渲染等任务进行并行化。这种并行是**细粒度的**(fine-grained)，但它受到芯片上物理核数的限制(2015 年高端智能手机通常只有 8 个核)。另一方面，集群系统(即分布式内存架构)可以根据待处理的数据集规模来实时扩展资源。集群的构建具有很大的灵活性，例如可以选择异构的计算机节点，然后确定最适合这些节点的互连拓扑结构。这种并行是**粗粒度的**(coarse-grained)，因为在集群中发生节点间通信之前，每个节点可以独立地进行大量的本地计算。

本书侧重于在分布式内存系统上利用标准消息传递接口(Message Passing Interface，MPI)来设计并行算法。MPI 是管理集群节点之间通信和全局协同计算的**实际标准**。目前存在多种 MPI 标准的供应商实现，它们可以与 C、C++、Fortran、Python 等多种编程语言绑定。我们选择面向对象的语言 C++来实现数据科学中的算法，并使用和 C 语言绑定的 OpenMPI 应用程序编程接口(Application Programming Interface，API)来编写并行程序。

本书中两部分内容的简要介绍如下。

第一部分：基于消息传递接口的高性能计算

第 1 章首先简单介绍了 HPC 世界，然后讲解了 Amdahl 定律和 Gustafson 定律，这两个定律刻画了并行程序的理论最优加速比和扩展加速比。

第 2 章讲解了 MPI 的主要概念和编程接口：阻塞/非阻塞通信的概念、死锁和多种全局通信函数(例如 broadcast、scatter、gather、all-to-all、reduce、parallel prefix 等)。

第 3 章着重介绍了互联网络拓扑的作用。我们首先区分*物理拓扑*和*虚拟拓扑*(或称为*逻辑拓扑*)，并在设计并行算法的时候考虑不同网络拓扑对性能的影响。特别讲解了环形(包括优化的流水线广播)和超立方体形网络拓扑上的通信过程，后者依赖于节点的特定编号，称为格雷码。

第 4 章讲解了基于分布式内存的主要的并行排序算法。首先对著名的快速排序算法(Quicksort)进行了简单的并行化，然后介绍实际中广泛使用的 Hyper-Quicksort 和 PSRS(Parallel Sorting by Regular Sampling)算法。

第 5 章研究了一些矩阵相乘和向量相乘的算法，并简要介绍了在环和环面(torus)的拓扑结构中计算矩阵乘积的各种技术。

第 6 章介绍了一个比较热门的并行编程范式，称为 MapReduce(通常与开源系统 Hadoop 一起使用)。MapReduce 可以通过两个主要的用户定义的函数(map 和 reduce)来构建程序，然后部署到大量的网络互连的计算机上来完成计算任务。然而，MapReduce 也是一个完整的框架，包括一个主从架构。该主从架构能够处理各种硬件故障，或者当一些机器执行得太慢时，将这些机器上的并行计算任务(作业)重新发送到其他的机器上执行。该章还讲解了如何利用专门的名为 MR-MPI 的软件库在 MPI(MPI 没有容错能力)中实现这些类型的 MapReduce 算法。

第二部分：面向数据科学的高性能计算

这部分简要介绍了数据科学，并进一步讲解了如何使用 MPI 并行化数据科学中的算法。

首先介绍了两个最基本的数据聚类技术，分别是平面划分聚类(第 7 章)和层次树聚类(第 8 章)。聚类是探索性数据科学中一个非常重要的概念，用于发现数据集中的分类、同质数据中的分组。

第 9 章介绍了基于 k 最近邻规则(k-nearest neighbor)的有监督分类，并和 k 均值(k-means)聚类算法进行关联。

第 10 章介绍了另一个计算科学中的新范式，允许人们在大型数据集(潜在的高维度)上解决优化问题。这种新范式就是寻找核心集(core-set)，这些核心集就是原数据集的子集，而且和原数据集相比具有良好的近似性。这种技术最近

变得非常流行，能够将大数据(big data)缩小到小数据(tiny data)！由于数据通常具有高维度特征，所以还简要介绍了一种有效的线性降维技术，其中讲解了 Johnson-Lindenstrauss 定理，并给出一个简单的方法计算低失真嵌入，从而将数据从高维转化为低维，并确保在规定的近似因子内数据点之间的距离保持不变。有趣的是，嵌入的维度与原始外在维度无关，而是依赖于数据集大小的对数和近似因子。

第 11 章涵盖了一些图(graph)算法。图在社交网络分析和其他应用领域中是比较常见的。因此首先介绍一个顺序启发式方法和一个并行启发式方法来查找图的稠密子图，该子图近似于“最稠密”子图。然后介绍了在计算机集群上利用分支限界法来进行图同构检测。图同构检测是一个备受关注的问题，因为它的理论复杂度还没有得到解决(尽管对于图的某些特定子类存在一些多项式算法)。

每章最后会对该章的一些要点进行总结。请读者浏览这些总结，以便进行第一遍快速阅读。在一些章节结束时会给出 40 多道练习题，这些练习标有各种难度，并允许读者对练习所涵盖内容的理解程度进行自测。以星号开头的部分可以先跳过，稍后再进行阅读。

本书的主要目的是帮助读者设计并行算法，然后利用 C++和 C 语言绑定的 MPI 编写程序实现相应的并行算法。第二个目的是让读者对高性能计算和数据科学有更深刻的了解，并希望更好地促进两者之间的交叉。

本书是关于高性能计算和数据科学的入门教材，面向具有基本算法知识和编程能力的读者。因此，本书不包含(也没有提及)高性能计算和数据科学领域的高级概念。例如，任务调度问题和嵌套循环的自动并行化虽然在高性能计算中很重要，但是本书并没有涉及。类似地，本书也省略了数据科学领域中的回归技术和核心机器学习方法。

教辅资源

本书的额外资源(包括超过 35 个用 MPI/C++/R/Scilab/Gnuplot/Processing 编写的程序、幻灯片、相关链接和其他精彩内容)可以通过网址 https://www.lix.polytechnique.fr/nielsen/HPC4DS/获取。

程序的源代码可以在上述网址以下列方式获取：

```
WWW source code: example.cpp
```

祝阅读愉快！

Frank Nielsen

2015 年 12 月

致　谢

非常感谢以下这些有才华的同事，他们给了我非常宝贵的反馈意见（姓名按随机顺序排列）并帮助我完善了本书：Claudiad' Ambrosio，Ulysse Beaugnon，Annaël Bonneton，Jean-Baptiste Bordes，PatriceCalégari，Henri Casanova，Antoine Delignat-Lavaud，Amélie Héliou，Alice Héliou，Léo Liberti，Frédéric Magoulès，Gautier Marti，Sameh Mohamed，François Morain，Richard Nock，Pierre-Louis Poirion，Stéphane Redon，Thomas Sibut-Pinote，Benjamin Smith，Antoine Soulé，Bogdan Tomchuk，Sonia Toubaline 和 Frédéric Vivien。除了以上这些同事，我还与其他很多同事进行了讨论，当你们读到这句话时，希望你们能够知道，从这些宝贵的交谈中，我受益匪浅。我还要感谢所有巴黎综合理工大学 INF442 课程的学生，感谢他们富有成效的意见和反馈，并且感谢巴黎综合理工大学计算机科学学院（DIX）的支持。

目　录

第一部分

基于消息传递接口的高性能计算

第 1 章

走进高性能计算

1.1 什么是高性能计算

高性能计算(High Performance Computing，HPC)是一个综合的领域，包括各种并行编程范式、与各范式相关的编程语言和应用编程接口(Application Programming Interface，API)、定制的软件工具、专门的国际会议(ACM/IEEE Super-Computing，SC)等。笼统地讲，HPC 是一个在科学和技术上研究超级计算机(Super Computer，SC)的领域。

世界排名前 500 的超级计算机榜单(简称 Top500[⊖])会定期更新并发布在互联网上。在 2014 年，由中国国家超级计算广州中心研制的“天河二号”超级计算机(英文翻译为 MilkyWay-2)获得了该榜单的第一名。该超级计算机包含了惊人的 312 万个核，每个核是一个处理单元(PU)，整体性能达到了 54.9PFlops，PFlops 表示每秒 10^{15} 次浮点运算。这个排名第一的超级计算机需要 17.8MW 的电力才能正常工作！粗略地算，1MW 的电力大约花费 100 美元/时，这意味着该超级计算机每年的电费大约是 1 百万美元。

表 1-1 总结了(超级)计算机的处理能力和内存大小对应的规模数量级。如今，国内外的各超级计算团队正在研制 Exaflop 性能的超级计算机(10^{18} Flops，1024PFlops)，期望在 2017～2020 年间实现这一目标，紧接着，在 2030 年让超级计算机的性能迎来 zetaFlops(10^{21})时代，然后是 yottaFlops(10^{24})，并一直保持性能不断提升。

表 1-1 超级计算机的计算性能和内存大小所对应的规模数量级：超级计算机是根据 Flops(每秒的浮点运算次数)和内存的字节(8 位为 1 字节)容量进行排序的

单位	规模	计算性能	内存大小(字节)
K (kilo)	10^3	KFlops	KB
M (mega)	10^6	MFlops	MB

⊖ http://www.top500.org/。

（续）

单位	规模	计算性能	内存大小（字节）
G（giga）	10^9	GFlops	GB
T（tera）	10^{12}	TFlops	TB
P（peta）	10^{15}	PFlops	PB
E（exa）	10^{18}	EFlops（2017～2020 年）	EB
Z（zeta）	10^{21}	ZFlops	ZB
Y（yotta）	10^{24}	YFlops	YB
…	…	…	…
googol	10^{100}	googolFlops	googol 字节
…	…	…	…

注：当选用 $1024=2^{10}$（2 的次幂）来代替 $1000=10^3$ 作为两个连续的数量级之间的倍数时，我们可以得到以下的国际单位（SI）：Ki（2^{10}）、Mi（2^{20}）、Gi（2^{30}）、Ti（2^{40}）、Pi（2^{50}）、Eo（2^{60}）、Zo（2^{70}）和 Yi（2^{80}）

这个传统的超级计算机评价标准只依据算术运算的峰值计算速度，而完全忽略了为了达到这样的计算速度所消耗的能源。另外一个绿色的榜单叫作绿色 HPC[⊖]（green HPC），每年评比两次，该榜单根据超级计算机的 MFlops/W 对各超级计算机进行评比。在 2014 年 11 月，来自 GSI 亥姆霍兹中心（德国，达姆施塔特）的 L-CSC 超级计算机取得了 5.27 GFlops/W 的性能。比照来看，L-CSC 超级计算机在 Top500 排行榜中列 168 位，共10 976 个核，并且其峰值计算速度达到了 593.6TFlops。尽管处理能力是一个非常重要的评价标准（同时也是激励研究人员不断开发 HPC 解决方案的重要因素），但同时我们也需要考虑其他因素，例如整体的内存大小、互联网络的带宽等。最后，我们需要指出每 GFlops 的花费正在呈指数下降，在 2015 年 1 月，每 GFlops 估计花费 0.08 美元。

1.2　为什么我们需要 HPC

对此，浮现在人们脑海中的第一个答案是 HPC 有助于更快、更加准确地（对于模拟类的应用，例如天气预报、汽车碰撞测试中的计算力学，或者其他各种复杂现象的建模）运行程序。HPC 还能解决更大规模的问题：在更细粒度的网格或者更大的数据集（大数据趋势）上进行模拟。但是很少有人知道 HPC 同样有助于节约能源：在相同的浮点计算性能下，我们更倾向于使用低功耗的处理器，因为高功耗的处理器需要消耗更多的能源才能正常工作。最后一点，HPC 非常适合某些在本质上具备天然并行性的算法。实际上，在图像或视频处理领域用到的算法经常需要进行滤波计算，这种运算是针对每个像素或体素（医学图像中）独立进行的，因此滤波计算可以并行执行。在后一种情况下，图像显卡（图像处理单元，GPU）是一个众核（现今已经有几千个核）硬件显卡。举例来说，一个高端的 NVIDIA 显卡拥有 1536 个核，计算速度能够达到 2.3TFlops。

⊖ http://www.green500.org/。