Broadview
www.broadview.com.cn

InfoQ

[PACKT] PUBLISHING open source* community experience distilled

Python Parallel Programming Cookbook

Python并行编程手册

常见问题的快速解答
掌握高效的并行编程技术并使用Python构建强大的应用

[意] Giancarlo Zaccone 著
张龙 宋秉金 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

Python Parallel Programming Cookbook

Python并行编程手册

[意] Giancarlo Zaccone 著
张龙 宋秉金 译

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

内容简介

若想充分利用所有的计算资源来构建高效的软件系统，并行编程技术是不可或缺的一项技能。本书以Python为蓝本，对并行编程领域的各项技术与知识进行了广泛且深入的讲解。通过对本书的学习，读者将能够快速且准确地掌握并行编程方方面面的技能，从而应用在自己的项目开发中，提升系统运行效率。

本书共分为6章，从原理到实践系统化地对并行编程技术进行了层层剖析，并通过大量可运行的实例演示了每一个知识点的具体运用方式，是提升并行编程技能的一本不可多得的好书。相信本书的出版将会填补Python在并行编程领域应用的一大空白，能够帮助想要从事并行编程与并行计算的读者提升实践能力，并将这一能力应用到实际的项目开发中。

版权贸易合同登记号 图字：01-2015-8415

图书在版编目（CIP）数据

Python 并行编程手册 /（意）詹卡洛·扎克尼（Giancarlo Zaccone）著；张龙，宋秉金译 . —北京：电子工业出版社，2018.4
书名原文：Python Parallel Programming Cookbook
ISBN 978-7-121-33753-6

Ⅰ . ① P… Ⅱ . ①詹… ②张… ③宋… Ⅲ . ①软件工具—程序设计—手册 Ⅳ . ① TP311.561-62

中国版本图书馆 CIP 数据核字（2018）第 036159 号

策划编辑：许　艳
责任编辑：刘　舫
印　　刷：三河市君旺印务有限公司
装　　订：三河市君旺印务有限公司
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：15.25　　字数：361千字
版　　次：2018年4月第1版
印　　次：2018年4月第1次印刷
定　　价：59.00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）88254888，88258888。

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。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

译者序

今日之时代是并行编程与多核的时代，硬件成本越来越低，如何充分利用硬件所提供的各种资源是每一个软件开发者需要深入思考的问题。在并行编程的浪潮下，每一个有追求的软件开发者都应该拥抱这种变化，转换编程思维，使程序能够在多核及并行的环境下高效运行。

本书是一本专门介绍并行编程的图书。通过对硬件与软件的剖析，总结出了影响并行编程的多种因素，并通过大量的理论与实践来分析如何实现并行编程，以及如何更好地利用硬件资源来提升程序并行运行的效率。

值得一提的是，本书以 Python 语言作为范本对并行编程进行了深入的讲解。Python 是一门简洁且优雅的语言，目前的发展势头非常迅猛，在很多领域都能见到它的身影。以 Python 语言作为示范语言来讲解并行编程是非常恰当的。

本书首先对 Python 编程语言与并行编程理论进行了较为详尽的讲解，然后介绍了基于线程与基于进程的并行编程方式。通过对这两种方式进行介绍，读者可以迅速进入并行编程领域，掌握并行编程的一般方法与技巧，同时还会对 Python 中涉及线程与进程的模块有所了解，在后半部分，本书对异步编程与分布式 Python 以及如何使用 Python 进行 GPU 编程进行了深入的讲解，这些内容将会帮助读者实现从了解并行编程到深入掌握并行编程的蜕变。

值得一提的是，全书在理论讲解过程中辅以大量可运行的代码示例。这些示例非常好地演示了所讲理论的实际运用，读者可以通过这些示例加深对并行编程相关理论的理解与认识，也可以对这些示例稍加修改应用到实际的项目开发过程当中。

并行编程是一个较为复杂的技术领域，常常令很多开发者望而却步。不过，本书的出现将会改变这一局面。通过对本书的学习，读者将会快速且准确地建立起对并行编程的认知，并进

一步加强对并行编程的理解；此外，读者还可以通过阅读本书，掌握用 Python 实现并行编程的各种方式与技巧，并形成良好的并行编程思维方式。

全书由张龙与宋秉金翻译完成，其中张龙翻译了第 1~3 章的内容，宋秉金翻译了第 4~6 章的内容。在图书翻译过程中，得到了电子工业出版社计算机出版分社的许艳、张春雨二位老师的大力协助，在此向二位老师表示深深的感谢。二位老师在专业素养与团队协调方面展现出了极高的专业性，确保了本书的翻译工作能够顺利完成。每次与二位老师沟通都非常顺畅，同时进一步确保了译稿的质量。

虽已尽心尽力，奈何技术与文字水平有限；虽已校对多次，但依然不敢保证全书没有任何错误。因此，读者在阅读本书的过程中如果发现任何问题都请不吝赐教，可以通过 zhanglong217@163.com 联系我，以期再版时改进。

最后，衷心期望本书能给希望系统学习并行编程的读者带来切实的帮助，帮助大家准确建立起并行编程的思维方式，并能应用到实际的项目开发当中。

张龙

2018 年 2 月 8 日于北京

关于作者

Giancarlo Zaccone 拥有超过 10 年的管理研发项目的经验，涉及科学与工业这两个领域。他曾以研究员身份就职于国家研究委员会（CNR），主要从事一些并行科学计算与科学可视化项目。

他目前作为一名软件工程师就职于一家咨询公司，主要负责开发和维护一些面向太空和防御应用的软件系统。

Giancarlo 拥有那不勒斯费德里科二世大学的物理学硕士学位，并且获得了罗马大学科学计算专业的第二研究生学位。

可以通过 https://it.linkedin.com/in/giancarlozaccone 了解到关于 Giancarlo 的更多信息。

关于审校者

Aditya Avinash 是一名研究生，专注在计算机图形学与GPU上。他感兴趣的领域有编译器、驱动程序、基于物理学的渲染以及实时渲染等。他目前正在为MESA（Linux的开源图形驱动栈）贡献力量，主要负责实现AMD后端的OpenGL扩展。这是他真正感兴趣的地方。他还喜欢编写编译器，将高层次抽象代码转换为GPU代码。他开发了Urutu，它可以使用Python赋予GPU线程级的并行能力。出于这一点，NVIDIA资助了他一对Tesla K40 GPU。目前，他致力于RockChuck的开发工作，负责根据不同后端将Python代码（使用数据并行抽象编写）转换为GPU/CPU代码。这是他听取了大量Python开发者希望实现针对Python与GPU的数据并行抽象的建议而启动的一个项目。

Aditya拥有计算机工程背景，设计了硬件与软件来适应某些应用（ASIC）。基于这一点，他获得了关于如何使用FPGA与HDL的经验。此外，他主要使用Python与C++编写代码。在C++中，他使用过OpenGL、CUDA、OpenCL以及其他多核编程API。由于他还是一名学生，因此他的大多数工作并不隶属于任何机构或个人。

Ravi Chityala 是Elekta Inc的一名高级工程师。他在图像处理与科学计算领域拥有超过12年的经验。他还是加利福尼亚大学的兼职讲师，负责讲授Python高级编程。一开始他将Python当作脚本工具来使用，并且喜欢上了这门简单、强大、富有表述力的语言。他现在使用Python进行Web开发、科学原型制造与计算，并将其作为胶水来自动化这个过程。他将自己在图像处理上的经验与对Python的热爱结合起来，与他人合著了图书 *Image Acquisition and Processing using Python*，该书由CRC出版社出版。

Mike Galloy 是一名软件开发者，他专注于高性能计算与科学编程中的可视化。他在工作中主要使用 IDL，不过偶尔也会使用 C、CUDA 与 Python。他目前在位于茂纳罗亚太阳天文台的国家大气研究中心（NCAR）工作。在这之前，他供职于 Tech-X 公司，是 GPULib 的主力开发者，GPULib 是一个针对 GPU 加速计算的 IDL 绑定库。他是开源项目 IDLdoc、mgunit 与 rIDL 的创建者与主力开发者，同时也是图书 *Modern IDL* 的作者。

Ludovic Gasc 是 Eyepea and ALLOcloud 公司的高级软件开发者与工程师，该公司是欧洲知名的开源 VoIP 与统一通信公司。

在过去的 5 年间，他基于 Python、AsyncIO、PostgreSQL 与 Redis 为电信部门开发了多款分布式系统。

可以通过他的博客 `http://www.gmludo.eu` 联系到他。

他还是博文 *API-Hour: Write efficient network daemons (HTTP, SSH) with ease* 的作者。要想了解更多信息，请访问 `http://www.api-hour.io`。

前言

计算机科学的研究不仅应该涵盖计算处理所基于的原则，还应该反映出这些领域当前的知识状态。时至今日，技术的发展要求来自计算机科学所有分支领域的专家通晓软件与硬件，它们之间的交互是理解计算处理基本原理的关键所在。

出于这个原因，本书特别关注硬件架构与软件之间的关系。

之前，程序员可以借助于硬件设计、编译器与芯片让其软件程序在不做任何修改的情况下，速度变得更快或效率更高。

这个时代已然结束。现在，如果希望程序运行速度能变得更快，那么它必须要成为一个并行程序。

虽然很多研究人员的目标是程序员不需要了解运行其程序的硬件所具有的并行性，不过要想实现这个梦想还需要很多年的时间。今天，大多数程序员还是需要透彻理解硬件与软件之间的联系，这样程序才能在现代计算机架构上高效运行。

为了介绍并行编程的概念，我们使用了 Python 编程语言。Python 很有趣，且易于使用，其流行度在近几年稳步上升。Python 是由 Guido van Rossum 在 10 多年前开发出来的，Python 语法的简洁性与易于使用的特性很大程度上来源于 ABC，这是一门于 20 世纪 80 年代开发出来的教学语言。

除了这个具体的上下文外，Python 还被用于解决实际的问题，它从一些典型的编程语言中借鉴了很多特性，比如说 C ++、Java 与 Scheme 等。这是其最卓越的特性之一，使得它广受专业软件开发者、科学研究产业以及计算机科学教育者的青睐。之所以有这么多人喜欢 Python，一个原因是它在实践与概念之间提供了最佳的平衡。Python 是一门解释型语言，因此无须陷入

编译与链接的泥潭中就可以立刻动手做事情。Python 还提供了广泛的软件库，可用在从 Web 开发、图形学到并行计算的各种任务中。这种侧重于实践的特性非常吸引广大读者，可以让大家运行本书所介绍的重要项目。

本书提供了大量的示例，这些示例的灵感来源于很多场景，可以用它们解决实际问题。本书介绍了并行架构的软件设计原则，并确保程序清晰可读，避免使用一些复杂技术，都是一些简单且直接的示例。每个主题都以一个完整、可运行的 Python 程序的一个片段来阐述，后面跟着的是程序的输出结果。

书中各章节采用模块化的方式组织，可以从最简单的讨论跳到高级的主题，这么做也适合于那些只想学习一些特定主题的读者。

我希望本书这样的结构与内容安排能够帮助读者更好地理解和掌握并行编程技术。

本书主要内容

第 1 章概览了并行编程架构与编程模型。本章介绍了 Python 编程语言、语言的特性、其易于使用和学习的特点、可扩展性以及丰富的软件库与应用。此外，本章还介绍了如何在应用以及并行计算中用好 Python 这个工具。

第 2 章介绍了如何通过 Python 线程模块来实现线程并行。通过完整的编程示例，读者将学习到如何同步和操纵线程来实现多线程应用。

第 3 章介绍了基于进程的用于并行化程序的方式。本章通过完整的示例展示了如何使用 Python 多线程模块。此外，本章还介绍了如何通过进程来实现通信，借助 Python 的 mpi4py 模块使用消息传递的并行编程范式。

第 4 章介绍了并发编程的异步模式。在某些方面，它要比基于线程的方式简单一些，因为它提供了单指令流，任务会明确放弃控制而非武断地挂起。本章介绍了如何通过 Python asyncio 模块将每个任务组织为一系列更小的步骤，并在异步模式下执行。

第 5 章介绍了分布式计算。它指的是从逻辑上（甚至可能是地理位置上分布的）聚合几个计算单元，并以一种透明且一致的方式协同运行单个计算任务的过程。本章将会介绍 Python 所提供的，使用面向对象、Celery、SCOOP、远程过程调用的架构实现解决方案，比如，Pyro4 与 RPyC。本章也介绍了其他不同的方式，如 PyCSP、Disco，后者是 Python 版本的 MapReduce 算法。

第 6 章介绍了现代图形处理单元（GPU），它使数值计算的性能有了突破性提升，代价则是编程复杂度的增加。实际上，GPU 编程模型要求程序员手工管理 CPU 与 GPU 之间的数据传输。本章将会通过程序示例与用例介绍如何使用 GPU 卡所提供的计算能力，其中使用了强大的 Python 模块：PyCUDA、NumbaPro 与 PyOpenlCL。

阅读前的准备

本书所有示例都已经在 Windows 7 的 32 位机器上进行过测试。此外，使用 Linux 环境将会对学习大有裨益。

运行示例所需的 Python 版本是：

- Python 3.3（前 5 章）
- Python 2.7（只有第 6 章需要）

需要使用到如下模块（都是可以自由下载的）：

- mpich-3.1.4
- pip 6.1.1
- mpi4py1.3.1
- asyncio 3.4.3
- Celery 3.1.18
- Numpy 1.9.2
- Flower 0.8.32（可选）
- SCOOP 0.7.2
- Pyro 4.4.36
- PyCSP 0.9.0
- DISCO 0.5.2
- RPyC 3.3.0
- PyCUDA 2015.1.2
- CUDA Toolkit 4.2.9（最低为此版本）
- NVIDIA GPU SDK 4.2.9（最低为此版本）
- NVIDIA GPU 驱动

- Microsoft Visual Studio 2008 C++ Express Edition（最低为此版本）
- Anaconda Python Distribution
- NumbaPro 编译器
- PyOpenCL 2015.1
- Win32 OpenCL Driver 15.1（最低为此版本）

本书面向的读者

本书是写给那些想要使用并行编程技术来编写强大且高效代码的开发者的。阅读完本书后，读者将掌握并行计算的基础知识与高级特性。Python 编程语言易于学习，即便不是专家也能够轻松理解本书所介绍的主题。

本书结构

本书包含如下组成部分。

准备工作

这部分介绍了攻略的主题，描述了如何为该攻略搭建好软件或是进行一些设置。

具体操作

这部分介绍了实现攻略所需要的步骤。

实例精解

这部分通常会对“具体操作”部分的内容进行解释和说明。

知识扩展

这部分包含了关于攻略的一些额外信息，目的在于使读者更加深入地理解攻略的内容。

参考

这部分包含一些关于攻略的参考信息。

约定

在本书中，你会看到各种样式的文本，用于区分不同类型的信息。下面对这些文本样式及其含义进行一些说明。

文本中的代码、数据库表名、目录名、文件名、文件扩展、路径名、虚拟的URL、用户输入以及Twitter handle会这样表示“要执行第一个示例，我们需要用到程序`helloPython-WithThreads.py`.”。

代码块样式如下所示：

```
print ("Hello Python Parallel Cookbook!!")
closeInput = raw_input("Press ENTER to exit")
print "Closing calledProcess"
```

希望读者注意到代码块中的某一部分时，相关行或是代码会加粗处理：

```
@asyncio.coroutine
def factorial(number):
do Something

@asyncio.coroutine
```

任何命令行输入或是输出的样式如下所示：

```
C:\>mpiexec -n 4 python virtualTopology.py
```

警告或是重要的说明位于框中。

提示与技巧位于这个图标后。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **下载资源**：本书如提供示例代码及资源文件，均可在下载资源处下载。
- **提交勘误**：你对书中内容的修改意见可在提交勘误处提交，若被采纳，将获赠博文视点社区积分（在购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方读者评论处留下你的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/33753

目录