使用Python高效爬取页面数据的艺术

精通Python爬虫框架Scrapy

Learning Scrapy

[美] 迪米特里奥斯 考奇斯-劳卡斯（Dimitrios Kouzis-Loukas） 著
李斌 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

精通Python爬虫框架Scrapy

[美] 迪米特里奥斯 考奇斯-劳卡斯（Dimitrios Kouzis-Loukas） 著
李斌 译

人民邮电出版社
北京

图书在版编目（CIP）数据

精通Python爬虫框架Scrapy / （美）迪米特里奥斯·考奇斯-劳卡斯著 ; 李斌译. -- 北京 : 人民邮电出版社, 2018.2

ISBN 978-7-115-47420-9

Ⅰ. ①精… Ⅱ. ①迪… ②李… Ⅲ. ①软件工具－程序设计 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第305803号

版权声明

◆ 著　　[美]迪米特里奥斯 考奇斯-劳卡斯
（Dimitrios Kouzis-Loukas）
译　　李 斌
责任编辑　傅道坤
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：16
字数：225 千字　　2018 年 2 月第 1 版
印数：1 – 3 000 册　　2018 年 2 月北京第 1 次印刷

著作权合同登记号　图字：01-2017-2291 号

定价：59.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

内容提要

Scrapy 是使用 Python 开发的一个快速、高层次的屏幕抓取和 Web 抓取框架，用于抓 Web 站点并从页面中提取结构化的数据。本书以 Scrapy 1.0 版本为基础，讲解了 Scrapy 的基础知识，以及如何使用 Python 和三方 API 提取、整理数据，以满足自己的需求。

本书共 11 章，其内容涵盖了 Scrapy 基础知识，理解 HTML 和 XPath，安装 Scrapy 并爬取一个网站，使用爬虫填充数据库并输出到移动应用中，爬虫的强大功能，将爬虫部署到 Scrapinghub 云服务器，Scrapy 的配置与管理，Scrapy 编程，管道秘诀，理解 Scrapy 性能，使用 Scrapyd 与实时分析进行分布式爬取。本书附录还提供了各种必备软件的安装与故障排除等内容。

本书适合软件开发人员、数据科学家，以及对自然语言处理和机器学习感兴趣的人阅读。

关于作者

Dimitrios Kouzis-Loukas 作为一位顶级的软件开发人员，已经拥有超过 15 年的经验。同时，他还使用自己掌握的知识和技能，向广大读者讲授如何编写优秀的软件。

他学习并掌握了多门学科，包括数学、物理学以及微电子学。他对这些学科的透彻理解，提高了自身的标准，而不只是“实用的解决方案”。他知道真正的解决方案应当是像物理学规律一样确定，像 ECC 内存一样健壮，像数学一样通用。

Dimitrios 目前正在使用最新的数据中心技术开发低延迟、高可用的分布式系统。他是语言无关论者，不过对 Python、C++和 Java 略有偏好。他对开源软硬件有着坚定的信念，他希望他的贡献能够造福于各个社区和全人类。

关于审稿人

Lazar Telebak 是一位自由的 Web 开发人员，专注于使用 Python 库/框架进行网络爬取和对网页进行索引。

他主要从事于处理自动化和网站爬取以及导出数据到不同格式（包括 CSV、JSON、XML 和 TXT）和数据库（如 MongoDB、SQLAlchemy 和 Postgres）的项目。

他还拥有前端技术和语言的经验，包括 HTML、CSS、JS 和 jQuery。

前言

让我来做一个大胆的猜测。下面的两个故事之一会和你的经历有些相似。

你与 Scrapy 的第一次相遇是在网上搜索类似“Web scraping Python”的内容时。你快速对其进行了浏览，然后想“这太复杂了吧……我只需要一些简单的东西。”接下来，你使用 Requests 库开发了一个 Python 脚本，并且挣扎于 Beautiful Soup 中，但最终还是完成了很酷的工作。它有些慢，所以你让它整夜运行。你重新启动了几次，忽略了一些不完整的链接和非英文字符，到早上的时候，大部分网站已经“骄傲地”存在你的硬盘中了。然而难过的是，不知什么原因，你不想再看到自己写的代码。当你下一次再想抓取某些东西时，则会直接前往 scrapy.org，而这一次文档给了你很好的印象。现在你可以感受到 Scrapy 能够以优雅且轻松的方式解决了你面临的所有问题，甚至还考虑到了你没有想到的问题。你不会再回头了。

另一种情况是，你与 Scrapy 的第一次相遇是在进行网络爬取项目的研究时。你需要的是健壮、快速的企业级应用，而大部分花哨的一键式网络爬取工具无法满足需求。你希望它简单，但又有足够的灵活性，能够让你为不同源定制不同的行为，提供不同的输出类型，并且能够以自动化的形式保证 24/7 可靠运行。提供爬取服务的公司似乎太贵了，你觉得使用开源解决方案比固定供应商更加舒服。从一开始，Scrapy 就像一个确定的赢家。

无论你是出于何种目的选择了本书，我都很高兴能够在这本专注于 Scrapy 的图书中遇到你。Scrapy 是全世界爬虫专家的秘密。他们知道如何使用它以节省工作时间，提供出色的性能，并且使他们的主机费用达到最低限度。如果你没有太多经验，但是还想实现同样的结果，那么很不幸的是，Google 并没有能够帮到你。网络上大多数 Scrapy 信息

要么太简单低效，要么太复杂。对于那些想要了解如何充分利用 Scrapy 找到准确、易理解且组织良好的信息的人们来说，本书是非常有必要的。我希望本书能够帮助 Scrapy 社区进一步发展，并使其得以广泛应用。

本书内容

第 1 章，Scrapy 简介，介绍本书和 Scrapy，可以让你对该框架及本书剩余部分有一个明确的期望。

第 2 章，理解 HTML 和 XPath，旨在使爬虫初学者能够快速了解 Web 相关技术以及我们后续将会使用的技巧。

第 3 章，爬虫基础，介绍了如何安装 Scrapy，并爬取一个网站。我们通过向你展示每一个行动背后的方法和思路，逐步开发该示例。学习完本章之后，你将能够爬取大部分简单的网站。

第 4 章，从 Scrapy 到移动应用，展示了如何使用我们的爬虫填充数据库并输出给移动应用。本章过后，你将清晰地认识到爬虫在市场方面所带来的好处。

第 5 章，迅速的爬虫技巧，展示了更强大的爬虫功能，包括登录、更快速地抓取、消费 API 以及爬取 URL 列表。

第 6 章，部署到 Scrapinghub，展示了如何将爬虫部署到 Scrapinghub 的云服务器中，并享受其带来的可用性、易部署以及可控性等特性。

第 7 章，配置与管理，以组织良好的表现形式介绍了大量的 Scrapy 功能，这些功能可以通过 Scrapy 配置启用或调整。

第 8 章，Scrapy 编程，通过展示如何使用底层的 Twisted 引擎和 Scrapy 架构对其功能的各个方面进行扩展，将我们的知识带入一个全新的水平。

第 9 章，管道秘诀，提供了许多示例，在这里我们修改了 Scrapy 的一些功能，在不会造成性能退化的情况下，将数据插入到数据库（比如 MySQL、Elasticsearch 及 Redis）、

接口 API，以及遗留应用中。

第 10 章，理解 Scrapy 性能，将帮助我们理解 Scrapy 的时间是如何花费的，以及我们需要怎么做来提升其性能。

第 11 章，使用 Scrapyd 与实时分析进行分布式爬取，这是本书最后一章，展示了如何在多台服务器中使用 Scrapyd 实现横向扩展，以及如何将爬取得到的数据提供给 Apache Spark 服务器以执行数据流分析。

阅读本书的前提

为了使本书代码和内容的受众尽可能广泛，我们付出了大量的努力。我们希望提供涉及多服务器和数据库的有趣示例，不过我们并不希望你必须完全了解如何创建它们。我们使用了一个称为 Vagrant 的伟大技术，用于在你的计算机中自动下载和创建一次性的多服务器环境。我们的 Vagrant 配置在 Mac OS X 和 Windows 上时使用了虚拟机，而在 Linux 上则是原生运行。

对于 Windows 和 Mac OS X，你需要一个支持 Intel 或 AMD 虚拟化技术（VT-x 或 AMD-v）的 64 位计算机。大多数现代计算机都没有问题。对于大部分章节来说，你还需要专门为虚拟机准备 1GB 内存，不过在第 9 章和第 11 章中则需要 2GB 内存。附录 A 讲解了安装必要软件的所有细节。

Scrapy 本身对硬件和软件的需求更加有限。如果你是一位有经验的读者，并且不想使用 Vagrant，也可以根据第 3 章的内容在任何操作系统中安装 Scrapy，即使其内存十分有限。

当你成功创建 Vagrant 环境后，无需网络连接，就可以运行本书几乎全部示例了（第 4 章和第 6 章的示例除外）。是的，你可以在航班上阅读本书了。

本书读者

本书尝试着去适应广泛的读者群体。它可能适合如下人群：

- 需要源数据驱动应用的互联网创业者；
- 需要抽取数据进行分析或训练模型的数据科学家与机器学习从业者；
- 需要开发大规模爬虫基础架构的软件工程师；
- 想要为其下一个很酷的项目在树莓派上运行 Scrapy 的爱好者。

就必备知识而言，阅读本书只需要用到很少的部分。在最开始的几章中，本书为那些几乎没有爬虫经验的读者提供了网络技术和爬虫的基础知识。Python 易于阅读，对于有其他编程语言基本经验的任何读者来说，与爬虫相关的章节中给出的大部分代码都很易于理解。

坦率地说，我相信如果一个人在心中有一个项目，并且想使用 Scrapy 的话，他就能够修改本书中的示例代码，并在几个小时之内良好地运行起来，即使这个人之前没有爬虫、Scrapy 或 Python 经验。

在本书的后半部分中，我们将变得更加依赖于 Python，此时初学者可能希望在进一步研究之前，先让自己用几个星期的时间丰富 Scrapy 的基础经验。此时，更有经验的 Python/Scrapy 开发者将学习使用 Twisted 进行事件驱动的 Python 开发，以及非常有趣的 Scrapy 内部知识。在性能章节，一些数学知识可能会有用处，不过即使没有，大多数图表也能给我们清晰的感受。

目录

第 1 章
Scrapy 简介

欢迎来到你的 Scrapy 之旅。通过本书，我们旨在将你从一个只有很少经验甚至没有经验的 Scrapy 初学者，打造成拥有信心使用这个强大的框架从网络或者其他源爬取大数据集的 Scrapy 专家。本章将介绍 Scrapy，并且告诉你一些可以用它实现的很棒的事情。

1.1 初识 Scrapy

Scrapy 是一个健壮的网络框架，它可以从各种数据源中抓取数据。作为一个普通的网络用户，你会发现自己经常需要从网站上获取数据，使用类似 Excel 的电子表格程序进行浏览（参见第 3 章），以便离线访问数据或者执行计算。而作为一个开发者，你需要经常整合多个数据源的数据，但又十分清楚获得和抽取数据的复杂性。无论难易，Scrapy 都可以帮助你完成数据抽取的行动。

以健壮而又有效的方式抽取大量数据，Scrapy 已经拥有了多年经验。使用 Scrapy，你只需一个简单的设置，就能完成其他爬虫框架中需要很多类、插件和配置项才能完成的工作。快速浏览第 7 章，你就能体会到通过简单的几行配置，Scrapy 可以实现多少功能。

从开发者的角度来说，你也会十分欣赏 Scrapy 的基于事件的架构.（我们将在第 8 章和第 9 章中对其进行深入探讨）。它允许我们将数据清洗、格式化、装饰以及将这些数据存储到数据库中等操作级联起来，只要我们操作得当，性能降低就会很小。在本书中，

你将学会怎样可以达到这一目的。从技术上讲，由于 Scrapy 是基于事件的，这就能够让我们在拥有上千个打开的连接时，可以通过平稳的操作拆分吞吐量的延迟。来看这样一个极端的例子，假设你需要从一个拥有汇总页的网站中抽取房源，其中每个汇总页包含 100 个房源。Scrapy 可以非常轻松地在该网站中并行执行 16 个请求，假设完成一个请求平均需要花费 1 秒钟的时间，你可以每秒爬取 16 个页面。如果将其与每页的房源数相乘，可以得出每秒将产生 1600 个房源。想象一下，如果每个房源都必须在大规模并行云存储当中执行一次写入，每次写入平均需要耗费 3 秒钟的时间（非常差的主意）。为了支持每秒 16 个请求的吞吐量，就需要我们并行运行 $1600 \times 3 = 4800$ 次写入请求（你将在第 9 章中看到很多这样有趣的计算）。对于一个传统的多线程应用而言，则需要转变为 4800 个线程，无论是对你，还是对操作系统来说，这都会是一个非常糟糕的体验。而在 Scrapy 的世界中，只要操作系统没有问题，4800 个并发请求就能够处理。此外，Scrapy 的内存需求和你需要的房源数据量很接近，而对于多线程应用而言，则需要为每个线程增加与房源大小相比十分明显的开销。

简而言之，缓慢或不可预测的网站、数据库或远程 API 都不会对 Scrapy 的性能产生毁灭性的结果，因为你可以并行运行多个请求，并通过单一线程来管理它们。这意味着更低的主机托管费用，与其他应用的协作机会，以及相比于传统多线程应用而言更简单的代码（无同步需求）。

1.2　喜欢 Scrapy 的更多理由

Scrapy 已经拥有超过 5 年的历史了，成熟而又稳定。除了上一节中提到的性能优势外，还有下面这些能够让你爱上 Scrapy 的理由。

- Scrapy 能够识别残缺的 HTML

你可以在 Scrapy 中直接使用 Beautiful Soup 或 lxml，不过 Scrapy 还提供了一种在 lxml 之上更高级的 XPath（主要）接口——**selectors**。它能够更高效地处理残缺的 HTML 代码和混乱的编码。