

一本基于生产环境指导使用Storm的实践应用指南, Andrew Montalenti
作序推荐

从实际案例出发, 结合丰富的代码示例, 循序渐进讲解Storm相关概念及技术, 覆盖Storm最有效的应用场景

Storm Applied: Strategies for real-time event processing

Storm应用实践

实时事务处理之策略

肖恩 T. 艾伦 (Sean T. Allen)
[美] 马修·扬科夫斯基 (Matthew Jankowski) 著
彼得·巴蒂罗纳 (Peter Pathirana)

罗晓翼 龚成志 译



机械工业出版社
China Machine Press



技术丛书

Storm Applied: Strategies for real-time event processing

Storm应用实践

实时事务处理之策略

[美] 肖恩·T·艾伦 (Sean T. Allen)
马修·扬科夫斯基 (Matthew Jankowski) 著
彼得·巴蒂罗纳 (Peter Patrino)

罗晓翼 蒋成志 译

藏书章



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Storm 应用实践: 实时事务处理之策略 / (美) 肖恩 T. 艾伦 (Sean T. Allen) 等著; 罗聪翼, 龚成志译. —北京: 机械工业出版社, 2017.12

(大数据技术丛书)

书名原文: Storm Applied: Strategies for real-time event processing

ISBN 978-7-111-58621-0

I.S… II. ①肖… ②罗… ③龚… III. 数据处理软件 IV. TP274

中国版本图书馆 CIP 数据核字 (2017) 第 297304 号

本书版权登记号: 图字 01-2015-8162

Sean T. Allen, Matthew Jankowski, Peter Pathirana: *Storm Applied: Strategies for real-time event processing* (ISBN 978-1-617291-89-0).

Original English edition published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, Connecticut 06830.

Copyright © 2015 by Manning Publications Co.

All rights reserved.

Simplified Chinese translation edition published by China Machine Press.

Copyright © 2018 by China Machine Press.

本书中文简体字版由 Manning 出版公司授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

Storm 应用实践: 实时事务处理之策略

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: 北京兆成印刷有限公司

版次: 2018 年 1 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 15.5

书号: ISBN 978-7-111-58621-0

定价: 69.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

大数据和物联网创业的时机已经越来越成熟，大数据相关技术也呈现出贬值效应，但这里说的贬值不是指不去钻研带来的落后，而是技术不再成为这个行业的独门利器。成熟的技术和方案已经降低了这个行业的准入门槛，我们要做的就是赶紧抓住机会切入互联网创业的下半场。

如果互联网的上半场是基于人口红利和流量红利在广度上进行扩展，那么下半场就应该将焦点放在深度扩展上。在面对业务去寻找解决方案的时候，我们本质上要做的应该是理解问题、提出问题，这是最难的事情，因为只有好的问题才能找到好的答案，然后才是如何用数据来实现解决问题的路径。所以学会如何分析和利用大数据对业务产生帮助才是关键，也就是说，要想提出好问题是非常困难的，通俗点讲就是要学会化繁为简。

在物联网行业中，数据扮演的角色是推动服务和功能，提供判断和解决问题的支撑。工业互联网对数据的低容错性和高实时性要求，让数据的价值拥有短期和长期之分。短期来讲，数据给予了最传统的统计和分析价值，比如基于单变量的描述性分析和多维度的分类聚合应用，在同方云我们常将这类数据应用于一些大型装备的维保支撑，开发的远程诊断工具提高了过程控制的效率。但大数据真正的价值发挥还需要更大的能量，积累的越多，可以挖掘的价值才越大，只有通过长期地反复尝试和对目标问题的调整，才能发现业务发展的规律，让数据在未来的预测上发挥其长期价值。所以团队中我们常说的一句话，就是“大数据不是告诉你 1000 台风机在做什么，而是应该告诉你第 1001 台风机在哪里”。

因此，大数据真正的机会是学会如何分析和使用数据，让其释放和体现自身的价值，而不是因为用了多么先进的技术和复杂的算法，才能发挥它的价值。我的统计学老师曾告诉我：“事物发展背后的逻辑永远是最简单的，统计学发展这么多年，就没有出现过复杂算法战胜了简单算法，没有简单的模型就没有复杂的模型，由大到小，才是分治递归的基础。”

所以本书最好的一点，就是用最简单的语言把问题讲清楚，帮助你理解并学会使用 Storm 来辅助解决实际中遇到的问题，基于详尽的案例讲解，覆盖 Storm 最有效的应用场景，支撑你在选型上做出合理的判断。以授之以渔的方式，让你在技术与业务之间游刃有余，将更多的精力放在问题定义和数据分析上。本书倡导的就是要学会以“ What”而不是“ How”的策略来思考问题，把那些计算和处理的过程都交给 Storm 来代劳吧！

感谢机械工业出版社华章公司的副总编杨福川先生对本书的高度重视，感谢责编王春华和唐晓琳老师的日夜工作，提高了本译著的整体翻译质量，本书出版过程中参与服务的还有多位台前幕后的工作人员，对此予以特别感谢！

感谢和我一同翻译本书的龚成志先生，他对技术的热爱和严谨支撑了本书的专业性，也很荣幸能和他一起共事。感谢我的妻子李洁，在过去的几个月中支持我的翻译工作，帮助我完成了大量的审阅工作，还有在此期间出生的可爱女儿雪桐，感谢她活泼可爱的笑容，消除了我一切的疲惫。感谢父母家人对我的鼓励，感谢同方云计算的领导对我的信任，感谢我的团队陪我一起度过多少次不知白天与黑夜的攻关和冲刺！

虽然我已经竭尽全力尽量准确地描述书中的技术要点，也进行了多轮的审校和通读，但仍难免有疏漏之处，如读者发现任何不理想之处，望能不吝赐教。

罗聪翼

2017 年 8 月成都

2016 年我主导了一个物联网项目：重型采矿设备的实时监控系统，需要对工业流数据进行实时处理，主要是对大型设备的各项状态数据进行过滤以及计算。设计之初我选择使用 Kafka 作为数据总线，并用 Kafka 的消费接口直接订阅消息完成计算处理。最开始的时候由于单机数据量比较小，并且只接入了几十台设备，业务逻辑也相对简单，只是简单的状态监控，一切计算都处理得称心得手，系统状态表现得非常良好稳定。

不过随着业务的发展，接入设备量越来越多，监控点位增多以及监控频次的需求提升，数据量越来越大，业务逻辑也开始越来越复杂，而此时我们才意识到当初低估了这些重型工业设备！整个系统的并发性能出现了严重的问题，并且代码也越来越难以维护，必须将整个系统进行重构。

在做方案调研时，复盘了之前流数据在实际业务中带来的压力和困难，我决定重新选择一个实时流处理框架，并且要求在重构结束后能保证业务的平滑迁移。这时市面上可供选择的实时计算框架已经有很多种了，经过初步调研发现，Storm 提供的功能应该正是我所

需要的：一个可扩展的高性能流式处理框架！

在重构期间，我从公司的书架上翻到了本书的原版英文书，阅读之后发现，其从实际案例出发，由简到繁，逐步深入和清晰地讲解了 Storm 的相关概念，以及如何有效地借助 Storm 解决现实中的问题。书中的案例帮助我快速地理解了 Storm 的各个技术要点，而且恰好还解决了当时遇到的一些实际问题。在这本书的帮助下，团队顺利完成了整个系统的重构工作，到目前为止，系统的性能与稳定性还没有发现太大的问题，达到了重构的预期效果。所以当好友罗聪翼邀请我一同参与本书的翻译工作时，我义不容辞地答应了他！

如果你先放下本书，退远一步以上帝视角来看目前的计算机应用，就会发现大数据在整个信息技术中其实越来越抢眼，Storm 已经成为主流的流式数据处理技术之一，但凡一提到流式处理，大家都会想到 Storm。

所以为了让读者更容易跟上技术的发展潮流，本书尽可能采取通俗易懂的方式，不讲高深的概念，从实际例子出发，逐步加深你对 Storm 的理解。正是如此，每读完一章你都能更进一步认识 Storm 的各项技术要点，逐步学会如何使用 Storm 解决实际问题。当完成整本书的阅读后，你就应该能够解决足够复杂的流式计算相关问题了。

另外，本书还结合了代码驱动的学习方法，对 Storm 技术进行了操作讲解，从最开始的概念到后面的案例解析，都是逐步展开的，让读者在循序渐进的实际操作中巩固知识点，哪怕你从来都没有接触过流式处理，也能在练习中逐步掌握 Storm 的相关知识。所以建议对于没有接触过 Storm 的读者，一定要从第 1 章开始顺序学习，并对每段代码都要运行一次，千万别跳过这些步骤！

在此我要特别感谢我的这位朋友，也就是本书另外一位译者罗聪翼，感谢他邀请我一起参与本书的翻译工作。我在翻译过程中由于各种原因曾多次想放弃，正因为他的鼓励和支持，我才能完成全部的翻译工作。

龚成志

2017 年 8 月

序 *Foreword*

“重写后端真是太难了！”

就从这说起吧，听到我那位聪明靠谱的同事 Keith Bourgoïn 发出的抱怨，而我俩已经在 Parse.ly 的分析后台上忙活一年多了，我们称呼这个系统叫“PTrack”。

Parse.ly 使用 Python 来构建，所以我们在设计系统时，可以很轻松地使用一些社区中常用的分布式计算工具，例如 Multiprocessing 和 Celery。尽管我们很熟悉这些工具，但实际上每 3 个月，数据量都会翻倍，而我们不得不在各系统间突破瓶颈，这里必须想点其他办法了。

于是，我们非常小心地对后端做了重写，使用轻量级 Python 进程来处理数据，而进程间使用 ZeroMQ 通信。鉴于 Python 语言的创造者将下一代语言命名为“Python3000”，我们戏称新系统为“PTrack3000”，即使大家还不清楚这个新系统是否真的可以解决之前的问题。

基于 ZeroMQ，我们认为它可以在每秒中交换更多的消息至各进程中，还可以确保系统间的运行更轻量。整个设计确实赢得了运行性能和效率，却在数据的稳定性上出现了问题。

而此时一件有意思的事件出现了，我们一直在公开市场中跟进的初创项目 BackType[⊖] 被 Twitter 收购了，而它被收购后的第一件事，就是将其流处理架构 Storm 正式对外发布。

我和 Keith 仔细研究了相关文档和代码，发现 Storm 正是我们想要的解决方案！

它也在内部使用到了 ZeroMQ，可以分别和其他工具叠加，共同组成并行处理运算，简化了整个操作，并提供了更清晰可靠的数据模型。尽管它是用 Java 写的，但文档中还是给出了大量其他语言的范例，以便支撑与框架间的兼容应用，例如 Python。就这样，我们很

⊖ 在 Storm 发布之前，我的团队发表了文章“BackType 数据工程师的秘密”（2011 年）：<http://readwrite.com/2011/01/12/secrets-of-backtypes-data-engineers>。

快就实现了基于 Storm 的 Parse.ly 数据分析后台改造，并命名为“PTrack9000！”。

那时 Storm 的原作者 Nathan Marz 也花了大量的时间去维护技术社区，包括参与各交流会、维护博客和用户论坛[⊖]。在早期阶段，Storm 的相关资料十分稀缺，你不得不在每个网站中去一点点地探索和整理思路。

我真希望在 2011 年就能看到类似你手上这本书，尽管当时 Storm 的文档已经十分完善，但还是缺乏实用类的案例，特别是与生产实施相关。接下来的 3 年里，尽管 Storm 在技术圈内十分热门，但直到 2014 年年底，市面上依然缺少一本实用类书籍！

一直没有人花精力去整理 Storm 的组件应用细节，Storm 的代码逻辑，如何优化拓扑性能，以及如何在生产环境中部署这些集群。现在，Sean、Matthew 和 Peter 一起合作完成的这本书，就将他们过去在 TheLadders 的项目实践经验都整理归纳出来，并完整呈现。这无疑是一本指导 Storm 用户实践应用的权威指南！

他们的表述清晰简洁，并配上大量图例说明，还附带了应用代码演示，你可以在短时间内学到我们团队过去花了多年才总结出来的 Storm 知识精华，这不仅省掉了你大量花在摸索上的时间，还避免了你一个人独自在电脑面前，被重构代码中的各种问题反复折磨。

我相信在完整学习本书之后，如果你同事再跟你讲“重写后端真是太难了”的时候，你可以自信地告诉他：“这次绝对不会！”

祝阅读愉快！

Andrew Montalenti

Parse.ly 联合创始人 & 首席技术官[⊖]

适用于 Storm 的 Python 包 streamparse 的创建者[⊖]

⊖ Nathan Marz 在他的博客上写下文章“Apache Storm 的过去和反思（2014 年）”，描述他的布道之路：<http://nathanmarz.com/blog/history-of-apache-storm-and-lessons-learned.html>。

⊖ Parse.ly 应用于数字可视化的网页版分析系统基于 Storm 构建：<http://parse.ly>。

⊖ 要在 Storm 上使用 Python，你可以在 Github 上参考项目 streamparse：<https://github.com/Parsely/streamparse>。

前言 *Preface*

在 TheLadders，我们从 Storm 刚发布时就开始使用（那时的版本号还是 0.5.x）。刚开始，我们只在一些非关键的业务流程上部署 Storm，在很长的一段时间里，我们的 Storm 集群都一直处于持续运行的状态，且十分稳定。正因为没出过什么问题，所以我们也没花太多的心思在上面。直到需要应对更多业务时，我们意识到 Storm 刚好是最合适的解决方案，可结果在实施的过程中暴露出了各种各样的问题。例如，我们需要在生产环境中去应对资源争夺，缺乏对底层运行原理的充分认知，不断寻找优化性能的次优方案，面临缺少可视化的系统运行状态监控，等等。

这促使我们花费大量的时间和精力去研究本书中即将呈现的内容。在学习理解 Storm 的过程中，我们多次翻阅了所有可找得到的文档，深入研究相关源代码，整理最适合的 Storm 解决方案，不断总结“最佳实践”，并且我们还增加了自定义的监控系统，便于更有效地排查故障和优化方案。

你可以在网上轻松查到有关 Storm 的原理文档，但我们发现，市面上依然缺少可以基于生产环境，指导使用 Storm 的实践应用文档。我们为此在博客上撰写了大量有关 Storm 的使用经验，所以当 Manning 找到我们希望合作一本 Storm 书籍时，大家一拍即合。我们有太多的知识想和大家一起分享，希望可以帮助大家少走我们曾走过的弯路，避开一些我们曾踩过的坑。

虽然我们分享出来的内容主要是基于在生产环境中，如何对 Storm 集群做优化、调试和故障排查，但我们更希望强调这里对 Storm 原理所需要的深入理解，同时展示出 Storm 的灵活性和广泛适用性，即使我们仅能代表众多使用 Storm 公司中的一员。

在本书中，我们将尽可能演示基于 Storm 下不同类型的应用案例，讲解 Storm 的核心概念，以便更容易理解如何在生产环境中执行优化、调试和故障排查。希望这种形式能适

用于不同层次的读者，无论是刚接触 Storm 的新人，还是拥有丰富经验而且遇到过和我们有相似经历的开发者。

本书绝对是一个团队协作的结晶，无论是来自 Manning 的伙伴，还是来自 TheLadders 的同事，大家都从最开始就尽可能地支持我们，耐心地协助我们完成测试和验证所有想法。

无论你的 Storm 使用经验处于什么样的层次，都希望本书能对你有所帮助，我们也很享受撰写本书的过程，因为每一天我们都学到了更多关于 Storm 的知识。

致 谢 *Acknowledgements*

感谢 TheLadders 所有为我们提供反馈和支持的同事，无论如何，这都是一本属于集体的书，指导着我们在集群上实现更多更酷的功能。

也感谢来自 Manning 并为本书撰写提供大量帮助的伙伴，这是一个很棒的团队，在合作期间我们从他们身上学到了很多关于写作的知识。特别感谢编辑 Dan Maharry，从第 1 章开始一直到最后一章完成，他为第一次写书的我们提供了大量的帮助，指导我们在错误和挫折中成长。

感谢所有参与本书的技术审校人员，感谢他们贡献了自己私人的时间来核实书中的各技术要点：Antonios Tsaltas, Eugene Dvorkin, Gavin Whyte, Gianluca Righetto, Ioanis Polyzos, John Guthrie, Jon Miller, Kasper Madsen, Lars Francke, Lokesh Kumar, Lorcon Coyle, Mahmoud Alnahlawi, Massimo Ilario, Michael Noll, Muthusamy Manigandan, Rodrigo Abreau, Romit Singhai, Satish Devarapalli, Shay Elkin, Sorbo Bagchi 以及 Tanguy Leroux。其中我要着重感谢 Michael Rose，感谢他为本书提供了大量高质量的反馈，可以说是本书最重要的技术审校人员。

感谢那些创造了 Storm 的人，没有他们，我们就不会有日夜奋斗的理由！我们还会坚持使用 Storm，也期待未来 Storm 可以带来更新的改进。

感谢 Andrew Montalenti 在我们早期手稿中提供的反馈，这给了我们很大的启发，并支撑我们完成了本书，他写的推荐序也很棒，我们没办法要求更多了。

最后还要感谢 Eleanor Roosevelt，她那句被大量错误引用的励志名言“美国在哪里都讲速度，燥热、肮脏、惹是生非的速度”，鼓舞着我们在困难中前进，持续地学习 Storm。

我们在看颁奖仪式时学到的一件事情，就是一定要感谢一路走来帮助我们的每一个人。

Sean Allen

感谢 Chas Emerick，如果不是因为和他激烈的争论，我可能根本不会下决心来写一本书，如果没有他的付出，可能就不会有人有机会读到这本书了。Stephanie，感谢他在我每次都想要放弃的时候鼓励我坚持下去。Kathy Sierra，感谢他在 Twitter 上和我沟通，让我能梳理清晰写作的思路。感谢 Matt Chesler 和 Doug Grove，他们帮助纠正了第 7 章的写作方向。感谢在 TheLadders 向我咨询问题的伙伴，是他们帮助我完成了第 8 章。感谢 Tom Santero，帮我审阅了我在分布式系统上的一些细节。感谢 Matt，帮我做了大量写书期间必须要做但我又不想做的事情。

Matthew Jankowski

首先感谢我的妻子 Megan，她是我永恒的动力来源，无论写书会占用多少时间，她都表现出无限的耐心，给予我坚定的支持。可以说没有她，我是无法完成本书的。还有我的女儿 Rylan，感谢她出生在写作的这段时间里，她给了我很大的启迪，也许她现在还根本没意识到吧。感谢我的家人、朋友和同事，感谢他们的无限支持和建议。感谢 Sean 和 Peter 在刚开始听到这个想法后，就愿意一起参与到本书的创作，这的确是一段漫长的经历，感谢有他们一起一路走来。

关于本书 *About this book*

大数据的概念日趋流行，能用于处理实时流数据的工具显得尤其重要，Apache Storm 就是这样一个能处理无限流数据的工具。

本书不仅供新手入门，也不只针对高阶学习。尽管理解大数据技术以及分布式系统可以帮助阅读，但我们并不希望这是作为阅读本书的前提条件。我们尽可能尝试迎合新人或是熟悉该领域的读者，本书最初的目的就在于呈现如何在生产环境中应用 Storm 的“最佳实践”，但为了更深刻地了解 Storm 的应用，一些基础知识还是有必要预习的，所以我们希望本书的内容可以面向不同经验层次的工程师。

如果你是刚开始学习 Storm 的新人，那么我们建议先阅读第 1 ~ 4 章，并且确保要全面理解，因为这几章包含了后面所需要的全部基础概念知识。如果你是有 Storm 应用经验的读者，那希望后面的章节会对你们更有帮助。总之，设计开发基于 Storm 的解决方案仅仅是个开始，在生产环境中实践这些方案才是我们需要在 Storm 上思考的重点。

本书的另外一个目的就是希望尽可能描述 Storm 的应用领域，基于此我们选择了一些典型的用户场景，希望对于理解未覆盖到的用户场景可以起到举一反三的作用。我们在选择用户场景的时候也做了不同难度的区分，希望至少能有一种可以适用于你当前正在使用的 Storm 场景。

本书旨在关注 Storm 的运行方式，而我们意识到 Storm 需要和许多不同的技术一起使用，包括不同的消息队列实现以及数据库操作实现等，所以在讲解每种用户场景的时候，我们会很谨慎地选择使用到的技术。我们不希望花太多精力在技术选型上，从而忽视了 Storm 使用上的重点讲解，所以你看到的每一个演示都默认使用的是 Java 语言。如果将案例中的应用切换到使用另外一门语言，这么做其实很容易，但我们还是希望明确一点，那就是本书的核心讲解并不在这些上面（事实上，我们在自己写的拓扑上大量使用了 Scala）。

路线图

第 1 章介绍大数据和 Storm 在大数据中所处的地位，该章的目的是展示一个选择 Storm 的理由和时机，一些关于大数据应用的关键特性，各类用于处理大数据的工具，以及明确 Storm 的工具类型。

第 2 章借助一个对某 GitHub 库提交数的统计案例，解释 Storm 的核心概念。该章将建立学习 Storm 的相关术语基础，尝试一小段代码来学习建立 Storm 工程，而这个案例中的概念也将贯穿本书。

第 3 章讲解在 Storm 下设计拓扑结构的最佳实践，同时以一个社交热力图的应用为例，展示了如何将问题基于 Storm 的结构来做分解，以便适用于程序的上下文实现部署。该章还讨论了如何处理不稳定的数据源，或者是不可靠的外部服务。同时在该章中介绍的首字节并行性，也将成为后续章节中的重点，最后在该章中还深入讨论了高级拓扑设计范式。

第 4 章以一个信用卡的授权系统为例，探讨 Storm 如何确保消息以上下文的形式传输，阐述 Storm 的实现机制，并且如何基于一套方案的部署，提供不同层面的可靠性支持。同时该章在最后做一个总结，说明如何在 Storm 的拓扑结构上，实现这种不同层次的可靠性支持。

第 5 章涵盖 Storm 集群的相关细节，还将讨论 Storm 集群的各类组件，Storm 集群如何提供容错机制，以及如何配置一个 Storm 集群，并在 Storm 集群生产环境中部署并启动拓扑。该章的提示内容将重点解释 Storm 的 UI 部分，因为后面的章节会越来越多地涉及 Storm UI 中的相关操作。

第 6 章阐述在 Storm 的拓扑结构中，基于一个限时抢购系统的应用案例，如何实现反复调优的过程。同时还讨论如何与外部系统协作，以及会对现有拓扑结构产生的影响。最后，我们将借助 Storm 现有的指标收集 API，讲解如何建立你的自定义指标。

第 7 章涵盖在同时使用多拓扑的时候，Storm 集群可能发生的各种冲突。我们将从一个拓扑中出现资源冲突的情况开始，逐步展开至拓扑间的系统资源冲突，以及 Storm 进程和其他进程甚至是操作系统之间的系统资源冲突分析，该章将带你领略 Storm 集群的完整应用效果。

第 8 章深入讲解 Storm，在完整理解后，你基本上就能应对各种情况下的调试了。还深入讲解 Storm 的并行化和执行器的中心单元，Storm 的内部缓存调用方式，溢出的前提条件，如何优化这些缓冲配置等，最后讨论 Storm 的调试日志输出。

第 9 章讲解 Trident 架构，它是一个基于 Storm 的上层抽象应用架构，同时演示基于它

开发一个互联网广播应用。还解释 Trident 结构的优势，以及什么情况下你会使用到它。我们会比较一个常规的 Storm 拓扑和一个基于 Trident 结构的拓扑，分析两者之间的区别。该章还将涉及 Storm 的分布式远程过程调用（DRPC）组件，以及如何借助它来实现拓扑的状态查询。最后，演示一个完整的 Trident 拓扑部署，以及如何实现该结构的扩展。

代码的下载和使用规范

本书中的所有代码可以在 <https://github.com/Storm-Applied> 中下载，包含以下章节中涉及的源代码。

- ❑ 第 2 章，GitHub 的提交次数计数。
- ❑ 第 3 章，社交热力图。
- ❑ 第 4 章，信用卡授权。
- ❑ 第 6 章，限时抢购系统。
- ❑ 第 9 章，互联网广播应用的播放日志统计。

很多源代码都以代码清单的方式展示，以提供完整的代码段。有些代码清单会加以注释，用于配合书中的功能讲解部分。在其他地方，只在需要的地方才会演示代码片段。无论是代码清单还是片段，我们都通过对字体加粗，用来强调这段代码正是文中配合解释的那一段。

软件要求

软件要求如下所示：

- ❑ 解决方案采用的版本是 Storm 0.9.3。
- ❑ 全部方案编程使用的语言为 Java 6。
- ❑ 代码的编译和打包使用的是 Maven 3.2.0。

About the cover illustration 关于原书封面插图

原书封面插图标题为“来自克罗地亚达尔马提亚的克拉维尔男人”，这是一幅传统的克罗地亚服饰展示图，由 19 世纪中期的 Nikola Arsenovic 创作，于 2003 年由克罗地亚位于斯普利特的人种学博物馆出版。该插图也是由一名来自斯普利特人种学博物馆的管理员在一片废墟中发现的，出土地点位于中世纪时期罗马城市的核心地带，也是公元 304 年 Diocletian 统治时期的帝国疗养宫殿所在地。这本彩色画册描绘了克罗地亚不同区域的人物画像，包含了各式各样的民族服饰和日常生活写照。

克拉维尔是一个位于克罗地亚杜布罗夫尼克东南部的狭小地区，地处于斯涅日卡山和亚得里亚海之间的狭窄区域，紧邻蒙特内格罗。图中的人物扛着自己的步枪，身上还别着手枪和匕首，枪套塞在他那宽大的腰带之间。从他警惕的架势和紧绷的神情上可以看出，他应该是在守卫着边境，或者是在提防偷猎者。他服饰上最有特色的一点，就是他那双印有复杂花纹设计的亮红色袜子，这是典型的达尔马提亚当地服饰风格。

人们的穿着打扮和生活方式在过去的 200 年里发生了天翻地覆的变化，而地区之间的多样性在时间的作用下逐渐被模糊淡化了，所以现在基本很难再说出不同大陆上原住民之间的区别，更别提相隔不远的不同村落和城镇。也许我们已经舍弃了追求文化之间的差异性，在尝试以更广泛的生活方式，来拥抱更为丰富多彩的个人生活和快节奏的科技类生活。

同样地，在这个难以分辨不同计算机书籍的时代，Manning 试图以两个世纪前的不同区域生活方式作为封面，将过去丰富多彩的场景引入至今，来赞美当下现代计算机技术不断创新和敢为人先的精神。

目 录 Contents

译者序	2.2.1 拓扑	13
序	2.2.2 元组	15
前言	2.2.3 流	16
致谢	2.2.4 spout	17
关于本书	2.2.5 bolt	18
关于原书封面插图	2.2.6 流分组	20
第1章 Storm简介	2.3 在 Storm 中实现 GitHub 提交数	
1.1 什么是大数据	监控看板	22
1.1.1 大数据的四大特性	2.3.1 建立一个 Storm 工程	22
1.1.2 大数据工具	2.3.2 实现 spout	23
1.2 Storm 如何应用于大数据应用	2.3.3 实现 bolt	26
场景	2.3.4 集成各个部分组成拓扑	29
1.3 为什么你希望使用 Storm	2.4 小结	30
1.4 小结	第3章 拓扑设计	31
第2章 Storm核心概念	3.1 拓扑设计方法	32
2.1 问题定义：GitHub 提交数监控	3.2 问题定义：一个社交热力图	32
看板	3.3 将解决方案映射至 Storm 的	
2.1.1 数据：起点和终点	逻辑	33
2.1.2 分解问题	3.3.1 考虑数据流本身施加的要求	33
2.2 Storm 基础概念	3.3.2 将数据点表示为元组	34
	3.3.3 确定拓扑组成的步骤	35