

RabbitMQ 实战

高效部署分布式 消息队列

[美] Alvaro Videla 著
Jason J.W. Williams
汪佳南 译

RabbitMQ in Action
Distributed Messaging for Everyone



Chris Richardson
大力荐

中国工信出版集团

电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

RabbitMQ实战

高效部署分布式消息队列

[美] Alvaro Videla 著
Jason J.W. Williams
汪佳南 译

RabbitMQ in Action
Distributed Messaging for Everyone

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

内 容 简 介

本书对RabbitMQ做了全面、翔实的讲解，体现了两位专家的真知灼见。本书首先介绍了有关MQ的历史，然后从基本的消息通信原理讲起，带领读者一路探索RabbitMQ的消息通信世界。这当中不仅包含了针对单台RabbitMQ服务器和RabbitMQ集群的讲解，还教导读者如何使用各种工具来进行监控。

本书内容浅显易懂，文笔风趣幽默。书中包含了丰富的、可以运行的示例程序源代码，读者可以自行下载并运行，这种亲身实践有助于加深对相关主题的理解。对于初学者来说，这样的编排非常适用。对于那些有一定RabbitMQ使用经验的读者来说，本书不仅可以巩固消息通信基本原理，同时也可加深对RabbitMQ高级特性的理解，譬如如何编写RabbitMQ插件。相信处于RabbitMQ探索之旅不同阶段的读者都能从本书获益。

本书适合有一定工作经验的开发者或者架构师阅读，也可以作为RabbitMQ运维工作的参考手册。

Original English Language edition published by Manning Publications, USA. Copyright © 2012 by Manning Publications. Simplified Chinese-language edition copyright © 2015 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文版专有出版权由Manning Publications 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2014-4907

图书在版编目（CIP）数据

RabbitMQ实战：高效部署分布式消息队列 /（美）维德拉（Videla,A.），（美）威廉姆斯（Williams,J. J.W.）著；汪佳南译. —北京：电子工业出版社，2015.10

书名原文：RabbitMQ in Action: Distributed Messaging for Everyone

ISBN 978-7-121-27164-9

I. ①R… II. ①维… ②威… ③汪… III. ①保密通信 IV. ①TN918.6

中国版本图书馆CIP数据核字（2015）第224246号

策划编辑：张春雨

责任编辑：李云静

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

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：20.75

字数：395千字

版 次：2015年10月第1版

印 次：2015年10月第1次印刷

定 价：75.00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。

质量投诉请发邮件至zltz@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

译者序

笔者从事互联网行业已五年有余，经历过大大小小十几个项目，其中包括金融领域方面的交易系统、医疗护理信息系统，再到后来的淘宝第三方软件供应商。这些项目不仅为我带来了丰富的阅历，更让我认识到软件开发的根本就在于降低软件复杂性。因此，我们需要采用可复用的软件设计模式，选取合适的软件构件来搭建自己的系统。这当中就用到了消息队列。

各位读者可能对消息队列并不陌生。消息队列提供一个异步通信协议，消息的发送者不用一直等待，直到消息被成功处理；而是立即返回。消息被暂存于队列当中，对消息感兴趣的消费者会订阅消息并处理它们。那么，在什么样的场景下需要使用消息队列呢？其中之一就是系统集成。这些系统可能是公司的遗留系统，抑或是第三方维护的系统。集成的方法有很多，其中之一是让这些系统往数据库的某几张表写入数据，然后让新的系统定时地去轮询并取出数据进行处理。这已经具备了消息队列通信的雏形：其中的数据库表就是消息队列，它定义了消息的标准结构，并暂存待处理的消息。对于小型系统和简单的需求来讲，这样的解决方案没什么问题，并且足够应付了。但是随着日后业务需求的扩张，接入的系统越来越多，对于消息通信的需求也变得越来越复杂：

1. 你需要将同一条消息发送给多个感兴趣的系统，同时又需要让系统只接受符合特定模式的消息。
2. 你开始觉得系统当中的同步处理方式严重影响了吞吐量，你想把用户订单处

理、消息通知等都修改成异步处理的方式。

3. 你开始关注系统的高可用性。同时你还想实时观测系统当中订单消息的处理速率，以便及时应对峰值压力。

通过使用标准的消息队列系统就可以轻松处理以上问题。当然它们提供的功能远不止这些。市面上有很多开放源码的消息队列系统，包括 JBoss Messaging、Joram、Apache ActiveMQ、Sun Open Message Queue、Apache Qpid、RabbitMQ，等等。其中就有本书讨论的 RabbitMQ。它实现了 AMQP 协议，并且遵循 Mozilla Public License 开源协议。它支持多种编程语言，可以方便地和 Spring 集成。最重要的一点是，使用消息队列并不是“杀鸡焉用牛刀”的问题，而是一种未雨绸缪，随着系统不断演进，你终将从消息队列中获益。

翻译一本书绝非易事。在此，我要感谢给予我帮助的赵震一和陈祥奎夫妇，你们是我的挚友与伯乐。感谢电子工业出版社的编辑张春雨和李云静，是你们的细心指导保证了本书的翻译质量。最后感谢我的女友，你是我的精神支柱，在你的陪伴和支持下，我得以全心全意地完成工作。

由于时间仓促，文中难免有不足之处，希望读者海涵，也希望各位能和我一样从中收获知识和快乐，谢谢大家。

汪佳南

2015 年 9 月于杭州

致我的祖父Maximiliano Godoy，
是你为我指引了人生的道路。谢谢。

——A.V.

致我的妈妈、爸爸和妹妹J'aime，
你们的爱、支持和信任
使我能得以攀越高峰……
致上帝，因为你的庇佑总能让我成功到达彼岸。

——J.W.

序

欢迎阅读本书。如果你像我一样，你很有可能在想：“我是否应该从头开始阅读呢？”唉，市面上的技术书籍琳琅满目，但并不是所有的图书都值得花费时间关注。所以请让我邀你继续往下阅读，看看下面这份说明是否合你的胃口：

- 你希望通过一种实用的方法来学习推技术、流数据，以及其他消息通信模式。
- 你想要获得专家级别的 RabbitMQ 技能，包括在设计方面和生产环境运行方面的最佳实践。

换句话说，本书不仅仅是一本 RabbitMQ 指南。它会为你讲解能覆盖多种使用场景的基本设计模式。它展现了为何越来越多的应用开始使用这些模式，以及哪些“可以做”、哪些“不能做”。

这些模式是怎样的呢？如果你曾经有想法将系统绘成一幅信息流或者网络图，而非栈的形式，那么你可能正在使用，或者正准备使用消息通信。你可能一直在考虑数据投递、非阻塞操作或者推送通知。或者，你想使用发布 / 订阅模式、异步处理或是工作队列。所有这些模式构成了众所周知的消息通信（messaging）。

消息通信是一种至关重要的能力：它使得软件应用可以相互连接并得以扩展。应用程序们既可以像一个巨大应用中的组件一样相互连接起来，也可以连接用户的设备和数据。消息通信从本质上来说是异步的，因为它通过分隔了数据的发送和接收来解耦应用。最奇妙的地方在于该连接模式可以在任何规模下以相同的方式工作。

规模是关键点。互联网无处不在，它作为应用程序投递的基础，使得规模在应

用程序设计中成为关键因素。我们已经无法停留在小规模思考方式上。最近，术语“大数据”（big data）已经变得非常流行。不过相较于最近几年来说，所有事物都往大的方向发展了。

举例来说，移动连接设备的数量很快就将超过地球上人口的数量。当我编写本书时，Facebook 正准备 IPO。CTO Bret Taylor 说道：“如果 Mark Zuckerberg 在宿舍编写 Facebook 的时候这些技术就已经成熟的话，那么现在的 Facebook 就已经是一个移动应用了。”

花点时间想想。大多数应用曾经都是这样运作的：加载文档或者从数据库获取数据，做一些处理，并将结果写入磁盘。将来的应用程序会更像 Facebook：总是在线，并且托管在云端，在任何地方都能访问。输入和处理是连续且自动的，并即时输出用户想要的过滤信息流。

如果不采用一系列专门的设计模式的话，那么这些级别的自动化、可访问性和伸缩性都无从谈起。在本书中你就能学到这些模式。作为现代消息通信技术发起者之一，Derek Collison 将消息通信描述为“移动的数据”，这一点令人印象深刻。很难想象有什么应用是不需要移动数据的。因此，消息通信无处不在。

这本书可以让你立即上手。消息通信模式将以可运行代码的形式展现在你面前。作者也会帮助你在自己的系统上运行起来。在 Jason J. W. Williams 和 Alvaro Videla 的帮助下，你就能零距离接触那些拥有多年运行大规模 RabbitMQ 系统经验的专家了。本书是他们将经验分享给社区这一杰出工作的成果。

在对 RabbitMQ 有了一点感觉之后，不管你使用何种语言编写程序，都能很容易地在广大 RabbitMQ 用户社区里寻求到帮助并找到更多示例。这使得 RabbitMQ 成为能满足你消息通信需求的最佳选择。

希望你已经被我吊起了胃口，准备翻开下一页继续阅读。书中会有关于消息和 RabbitMQ 的一切，作者将会为你揭开它们的神秘“面纱”。

Alexis Richardson

Rabbit Technologies 公司创始人和前任 CEO

VMware 云应用平台高级总监

前言

编写本书就像是在探索 RabbitMQ 一样——遇到需要解决的问题，但是却不知道解决方案是什么。在 2010 年 5 月前，我们甚至都不认识对方。在过去的两年里，我俩都活跃在 RabbitMQ 社区，但是我们并未真正相遇过。之后有一天在与 Alexis Richardson（时任 Rabbit 的 CEO）的一次谈话中，我与 Alvaro 互相认识了对方，这使得你现在手中所捧的这本书成为可能。我们之间的共同之处在于，我们都期望将自己通过艰难困苦才掌握的所有 RabbitMQ 知识写下来。回到 2010 年，这些知识（并且现今大部分时候仍然如此）散落在网络上那些一知半解的博客文章和简洁的技术指引中。换句话说，我俩都想要编写本书。真心希望本书在我们着手使用 RabbitMQ 的两年前就已经存在了。

我俩并非传统的消息通信背景出身，这使得我们很快成为朋友，并在很大程度上形成了本书的基调；我们假定本书的读者是那些之前从未听说过队列或者绑定（binding）的人。事实上，当我俩探索 RabbitMQ 时，我们甚至不知道消息通信是什么，或者说我们不知道这就是自己所寻找问题的解决方案。我（Jason）的处境是公司需要一种方法来获取从顾客那里收到的垃圾邮件报告，并在不同于消息主流的带外数据（out-of-band）中对它们进行处理。而 Alvaro 的情况是，他们公司的社交网络成员通信系统在 200GB 数据库负载下无法正常工作。就像很多其他诉诸消息通信的那些人一样，我俩首先尝试的是使用数据库表来解决以队列为中心的问题。像确保只有一个应用实例消费任意特定队列条目这样的问题，让我们在数据库驱动的方案上的尝试备受折磨，逼迫我们寻找更好的方法。毕竟，我们认识到自己不会是软

件历史上第一个碰到这类问题的人。

我们这两个问题的解决方案竟然惊人地相似：一位在 Plaxo 的朋友告诉我看看“RabbitMQ 之类的东西”是否能解决我的以队列为中心的问题，同时在中国的一位 Alvaro 的 Erlang 同事则给了他相同的建议。相距半个地球，我俩以相同的方式发现了 RabbitMQ，为的是尝试解决几乎同样的问题！事实上，你正阅读有关 RabbitMQ 的书，很有可能是因为相似的挑战以相同的方式指引着你来发现 RabbitMQ。这也解释了为何 RabbitMQ 如此流行：当尝试扩大我们构建的软件规模时，它能轻而易举地解决那些挥之不去的分布式数据的基本问题。

我们希望本书能够帮助你通过使用 RabbitMQ 来更快、更轻松的设计这些挑战的解决方案，从而使你可以将更多的时间用于编写改变世界的软件，同时花费更少的时间来了解如何加速消息通信代理。在这一路上，也许 RabbitMQ 会给你带来一位了不起的合著者，你从没想过他会成为你终生的朋友¹。本书是我们热爱编写软件的产物，同时也希望它能以你从未想象的方式来帮助你完成相同的事情。

Alvaro Videla

迪本多夫，瑞士

Jason J. W. Williams

爱达荷州博伊西市，美国

¹ 大家都说合著者的关系相比婚姻来说拥有更高的“离婚率”。这倒是一个不错的比较，因为合著一本书需要合著者不断地相互迁就和尊重，这样彼此才能生活在同一“屋檐”下。感谢上帝能让我撰写本书，同时能交到这么一位志同道合的朋友。

致 谢

虽然本书封面上只出现了两个名字，但是如果没有其他人的帮助，也就不会有这本书的存在了。首先，在我们开始编写本书的时候，我们想要感谢 RabbitMQ 的 CEO Alexis Richardson。如果没有他的引荐，Manning 就不会给我们这个机会，我们也就不会一起创作这本书了。同时，我们也要感谢他为本书作序。同样地，我们也要向 RabbitMQ 团队表达最真挚的感谢，感谢他们一直以来的帮助，以及对我们关于 Rabbit 细节之处提问的耐心解答。特别是，我们要感谢 Matthew Sackman 和 Matthias Radestock，如果没有他们两位的话，就不会有关于集群和 RabbitMQ 内部机制这几个章节了。

尤其是 RabbitMQ 团队的 Jerry Kuch，我们对他的感激之情无以言表。Jerry 自愿花费了无数时间来为每一章草稿进行准确的审校，包括对已完成的图书进行“官方”技术审校。每当由于我们经验不足而需要说明和建议时，他总能在 IM 上快速回复。作为 RabbitMQ 团队的接口人，他非常耐心且从不抱怨。如果你发现自己正在探索未知的 Rabbit 操作，可能就需要感谢 Jerry Kuch。他让本书更为出色；同时他也是一名优秀的工程师。

我们再怎么感谢 Manning 初级项目编辑 Maria Townsley 也不为过。Maria 帮助我们按计划写作。她忍受着我们的工作安排，以及时好时坏的原稿。最重要的是，她是我们的拥护者，并且任何对我们来说重要的资源她都努力争取。如果你满意本书的写作风格，那就感谢 Maria 吧，是她指引着我们做到这一点的。同时，我们也要感谢 Cynthia Kane，是她让我们完成最后几章并付诸印刷。当我们的工作方式太老套的时候，Cynthia 以最终项目编辑的身份加入进来。她适应了我们的工作方式，

并且将本书的工作当作自己从第一天起就加入进来那样看待。Cynthia 就像我们的三垒教练并护送我们回家。

最后，我们要感谢忠实的读者，他们从 Manning 的早期预览项目（MEAP）中购买了本书，同时也要感谢我们的审校们：Barry Alexander、P. David Pull、Bruce Snyder、Tony Garnock-Jones、James Williams、Patrick Lemieux、Bruce Lowekamp、Carlton Gibson、Paul Grebenc、Richard Siddaway、Gordon Dickens、Gene Campbell、Karsten Strøbæk、Jeff Addison、David Dossot、Daniel Bretoi 和 Ben Rockwood。你们没有报酬。但是你们把本书当作自己的孩子一样，给我们提出了详细的反馈和深思熟虑的见解。正是由于你们的功劳，才让本书变得无比美好。谢谢你们！

Alvaro

我想要感谢我的太太 Silvana，在本书的写作过程中她一直支持着我。我已经记不得由于编写本书的原因，我们错过了多少部电影、多久没有到外面逛逛了。我能说的只是“多谢理解”。另外，我还要感谢我的母亲一直以来对我的信任。毕竟，编写本书是全家人的功劳。我还要感谢我在中国 The Netcircle 的朋友们，在那里我开始对 Rabbit 深深地着迷了，并且还一天到晚不停地打扰他们。最后，我要感谢 Jason。Manning 为我引荐了这位合著者，我们最后也成了好朋友。

Jason

我要感谢我的父母和我的妹妹在整个写作过程中给予我的支持和爱。他们信任我，并且敦促我前进。当我不想写作的时候，他们会把我推出门外到咖啡馆继续写作。他们一直相信我能够完成本书的编写（即便这看来遥不可及）。

在和父母一起创建的初创公司里，我很荣幸能称呼父母为合伙人。作为我的合伙人，我亏欠了他们以及 DigiTar 太多太多了。当写作需要插入到工作时间的时候，他们从未抱怨过。并且给予我很大的自由度来平衡写作和工作。如果没有我们的公司，我也不会有动力去探索 Rabbit，也不会因写一些博客教程而受邀编写本书了。DigiTar 带给我许多祝福和机会，这本书就是其中之一。

最后我要感谢 Alvaro。我从不知道还存在你这么一位朋友，情同手足，就像来自另一位母亲的亲兄弟一样。谢谢你，你是我收到的最意外的祝福。

关于本书

RabbitMQ 是一个开源的消息代理和队列服务器，用来通过普通协议在完全不同的应用之间共享数据，或者简单地将作业排队以便让分布式服务器进行处理。这无关乎项目的大小：RabbitMQ 能够适应你的需要。你想要使用 X 编程语言来为其中一个应用组件快速建立原型，并能确保在第二天就可将其切换到一门更高性能的编程语言上吗？通过解耦通信协议，RabbitMQ 可以帮你做到这一点。你是否需要即时为你的社交网站处理图片上传，同时轻而易举地增加或者删除服务器？你可以使用 Rabbit 队列来存放作业，并让服务器负责为你做负载均衡和作业分派。通过使用 RabbitMQ 可以又快又简单地解决这类问题。本书的目的就在于向你展示如何围绕消息通信来实现最佳架构。

编写应用程序相对简单，保障应用正常运行才是挑战的开始。别担心，本书会讲到 RabbitMQ 管理、集群、安全和监控的最佳实践，因此你同样能够学习到运维方面的知识。

最后，我们将深入介绍 RabbitMQ 的核心和内部细节，让你理解服务器使用的系统资源，以便在设计系统架构的时候可以做好容量规划。同时，你将学习如何通过安装插件和创建自己的插件来扩展服务器。将你的编辑器准备好，因为我们就要开始用 Python、PHP、Erlang、Java 和 C# 来编程了。

路线图

第 1 章解释 AMQP 协议的起源、RabbitMQ 的诞生，以及它能解决什么样的行

业问题。下一步，你将安装服务器并创建第一个 Hello World 程序，通过 RabbitMQ 来发送数据。

第 2 章，你将沉浸在消息通信的世界中。我们从基础概念讲起，再到如何将这些概念映射到 AMQP（该协议用于 RabbitMQ）。之后，你将学习消息持久化，以及消息的生命周期：从消息的发布开始，直到在网络的另一头进行消费。

第 3 章展示基本服务器管理。你将看到如何启动和停止节点、如何配置权限，以及如何获取服务器数据统计。我们会给你一些有用的提示来为服务器做故障排除。

第 4 章介绍消息通信模式和最佳实践。你将学习发后即忘（fire-and-forget）模型、RPC 架构等。

第 5 章开启了 RabbitMQ 集群和高可用性设置系列，一共三章。在此，你将在本机和物理服务器上设置 RabbitMQ 集群。你将学习如何升级 RabbitMQ 集群节点，以及如何使用镜像队列。

第 6 章介绍如何使用 HAProxy 为 RabbitMQ 服务器做负载均衡。你也会学到如何创建智能的消息通信客户端，知道如何在服务器发生故障的时候进行重连。

第 7 章介绍主/备服务器是如何工作的，并以此结束了高可用性系列的讨论。你也将学习 Shovel 插件，它允许 RabbitMQ 跨越数据中心复制数据。

第 8 章展示如何对 RabbitMQ 进行管理。你将学习 RabbitMQ Management 插件及其 Web 界面。这一章的内容不止于此，我们还将展示该插件所提供的 REST API 概要。

第 9 章与第 8 章相衔接，详细介绍 REST API。你将学到如何通过使用该 API 来完成绝大多数的管理任务。为应用程序配置用户和虚拟主机提供前所未有的便利。

第 10 章教你如何监控 RabbitMQ，包括使用 Nagios 检测、使用 AMQP 和 REST API 来监控服务器的内部状态。你将学会如何通过检测问题来防患于未然。

第 11 章详细介绍交换器的内部工作机制（RabbitMQ 使用的路由算法）。我们将深入了解消息通信结构使用的资源细节，看看可以从架构决策中预见到什么。我们也会通过介绍为应用程序启用 SSL 连接来讲解 RabbitMQ 的安全性。

第 12 章将以如何通过添加新的插件来扩展 RabbitMQ 的行为来结束整本书。这

些插件既可以是别人编写发布的，也可以是你自己创建的。

编码规范及下载

所有源代码都以等宽字体排版（像这样 `fixed-width font like this`），以示区别。许多清单都伴有代码注释以突出重要的概念。在某些情况下，清单之后会有编号项目来连接到说明。

由于 RabbitMQ 的最大优势之一在于将不同语言编写的应用程序胶合在一起，因此我们同时使用 Python 和 PHP 作为示例的主要编程语言（在附录中会额外夹带一些 .NET 和 Java 的代码）。不过，我们想让这些示例对于使用不同编程语言的读者来说尽可能广泛地适用。由于我们不可能将每个示例都用不同的编程语言重写一遍，因此我们将这些示例代码提交到了 Github 代码库，以便你也能贡献代码：<https://github.com/rabbitinaction/sourcecode>。

在官方 Github 代码库中，你将能找到本书中最新版本的示例代码。这些代码中的一部分已经由其他读者转换成了不同的编程语言，譬如 Ruby。找不到你喜欢的编程语言？Fork 代码库，然后添加进去吧！之后发送给我们一个 pull request，我们会尽全力合并你的示例代码版本。（注意：你必须使用和我们的代码相同的 BSD 许可，以便我们将你的变更合并进来。）

如果你喜欢本书中权威且真正“官方”的示例代码的话，可以从出版商的网站中进行下载：<http://www.manning.com/videla/>。本书的最新版本中出现的具体代码总可以在这里找到。

关于作者

Alvaro Videla 是一位在基于 MQ 的应用方面的专业开发者和架构师。他对于 RabbitMQ 的演讲足迹遍布亚洲、欧洲和美国。

Jason J. W. Williams 是 DigiTar 的 CTO。该公司是消息通信服务供应商。从 2008 年开始，他就在公司指导设计和开发工作，包括使用 RabbitMQ 进行实时分析操作。

关于封面插图

本书封面的配图文字是“一位来自克罗地亚科尔丘拉岛上鲁巴达镇的当地农夫”。封面取自 19 世纪中期 Nikola Arsenovic 的一本克罗地亚传统服饰图集的复刻版，于 2003 年发表于克罗地亚斯普利特的人种学博物馆。封面插图是由人种学博物馆一位热心的图书管理员提供的。斯普利特在中世纪时是罗马帝国的核心：大概从公元 304 年起，退位的帝国国王 Diocletian 所居住的皇宫就在这里。这本书中囊括了克罗地亚各个地区的精美彩色插图，并对服饰和日常生活做了介绍。

鲁巴达镇是一个大约有着 1200 名居民的小渔村。它位于克罗地亚西海岸亚得里亚海的科尔丘拉岛东北岸的一个小岛上。封面上的这位农夫穿着该地区典型的工作服。这是一种没有过多修饰的简朴服装（他们通常只在周日和其他特殊场合才会穿着正式服装）。日常工作服由打满补丁的棕色裤子、棕色背心、白色亚麻衬衫和一顶草帽组成。他穿着简单朴素，抽着烟斗，拄着铁锹，在休息的时候等候小白兔自投罗网。

在过去的 200 年中，人们的着装和生活方式都发生了巨大的改变，在当时看来如此丰富的多样性已随时间流逝。现在，来自不同大陆的人都已难以区分，更不用说相隔仅数英里的村子和城镇居民了。也许文化的多样性是我们为了有更多丰富多彩的个人生活而必须付出的代价吧——相应地，人们得到更加多样化和快节奏的高科技生活。

Manning 出版社用两个世纪前各地别具一格的生活方式来诠释计算机行业的诞生与发展，用古老典籍和珍藏册把我们带回那个充满风土人情的年代。

目 录

第1章 天降奇兵	1
1.1 住在别人的地下城堡	3
1.2 救世主AMQP	5
1.3 RabbitMQ简史	5
1.4 百里挑一	8
1.5 在UNIX系统上安装RabbitMQ	8
1.5.1 为什么环境很重要——生活在 Erlang 的世界里	9
1.5.2 获取安装包	9
1.5.3 设置文件夹结构	9
1.5.4 首次运行 Rabbit	10
1.6 总结	12
第2章 理解消息通信	13
2.1 消费者和生产者（这可不是经济学课程哦）	14
2.2 从底部开始构造：队列	17
2.3 联合起来：交换器和绑定	22
2.4 多租户模式：虚拟主机和隔离	27
2.5 我的消息去哪儿了呢？持久化和你的策略	28
2.6 把所有内容结合起来：一条消息的一生	32
2.7 使用发送方确认模式来确认投递	37
2.8 总结	40