PACKT PUBLISHING

异步图书 www.epubit.com.cn

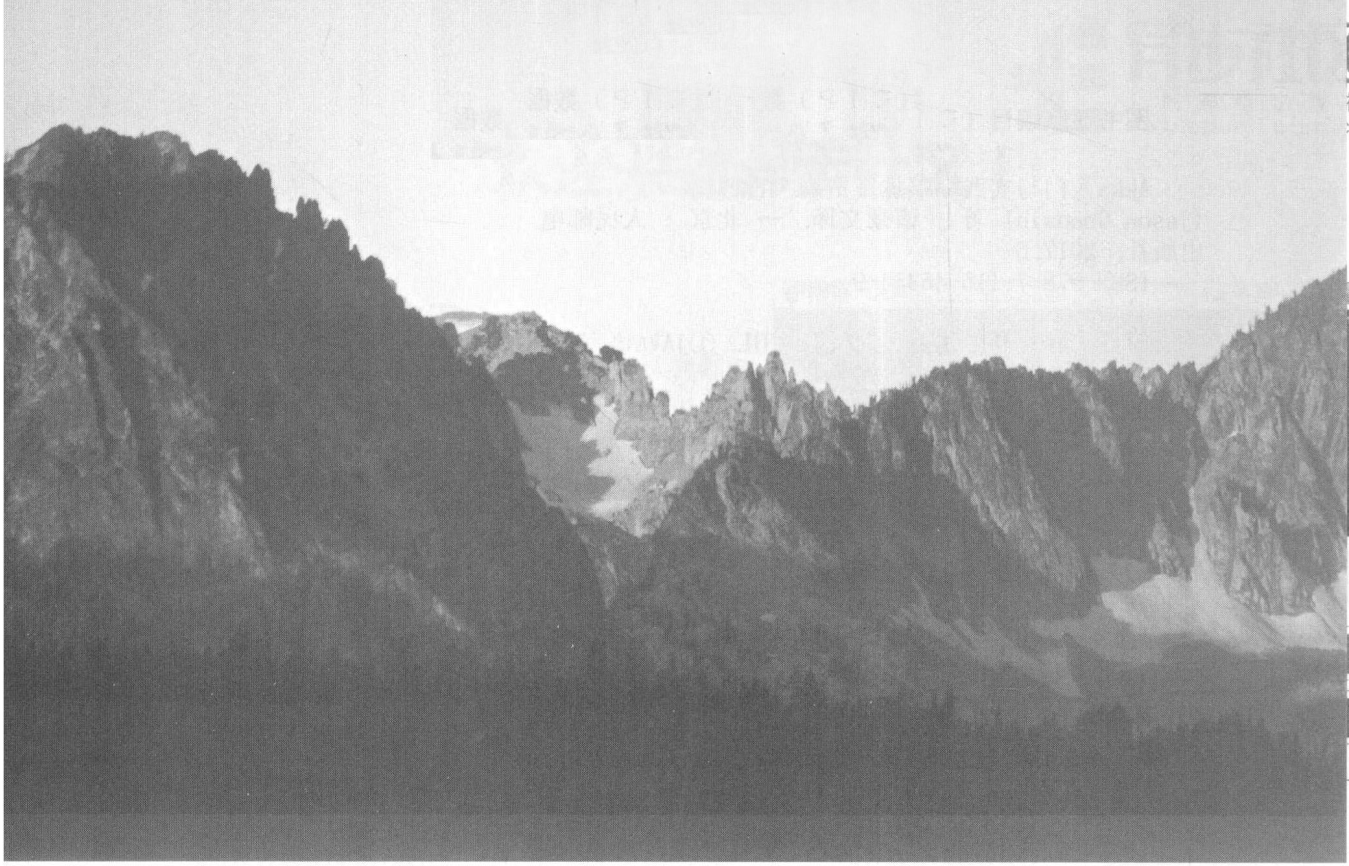
使用Akka构建高容错、并发、分布式应用程序

Akka 入门与实践

Learning Akka

[加] Jason Goodwin 著
诸豪文 译

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



Akka入门与实践

[加] Jason Goodwin 著
诸豪文 译

人民邮电出版社
北京

图书在版编目（CIP）数据

Akka入门与实践 / (加) 贾森·古德温 (Jason Goodwin) 著 ; 诸豪文译. -- 北京 : 人民邮电出版社, 2017.6
ISBN 978-7-115-45354-9

Ⅰ. ①A… Ⅱ. ①贾… ②诸… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2017)第077808号

版权声明

◆ 著 [加] Jason Goodwin
译 诸豪文
责任编辑 王峰松
责任印制 马振武

◆ 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
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：14.5
字数：272 千字 2017 年 6 月第 1 版
印数：1－2 500 册 2017 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2017-0521 号

定价：49.00 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：京东工商广字第 8052 号

内容提要

本书主要面向使用 Akka 工具集来构建大规模分布式应用程序的 Java 和 Scala 开发者，介绍了分布式系统的基本概念以及如何使用 Akka 来构建容错性高、可横向扩展的分布式应用程序。书中主要内容包括：Akka 工具集、Actor 模型、响应式编程、Actor 及 Future 的使用、Akka 消息传递模式、Actor 生命周期、监督机制、状态与错误处理、Akka 并发编程、路由、阻塞 IO 的处理、Akka Cluster、CAP 理论、Akka 邮箱问题的处理、Akka Testkit、领域驱动设计等。本书贯穿使用了分布式键值存储以及文章解析服务两个实例，将原理与实践结合，介绍了使用 Akka 设计并实现分布式应用程序的方法。

关于作者

Jason Goodwin 是一个基本上通过自学成才的开发者。他颇具企业家精神，在学校学习商学。不过他从 15 岁起就开始学习编程，并且一直对技术保持着浓厚的兴趣。这对他的职业生涯产生了重要的影响，从商学转向了软件开发。现在他主要从事大规模分布式系统的开发。在业余时间，他喜欢自己原创电子音乐。

他在 mDialog 公司第一次接触到 Akka 项目。mDialog 是一家使用 Scala/Akka 的公司，为主流出版商提供视频广告插入软件。这家公司最终被 Google 收购。他同时还是一名很有影响力的“技术控”，将 Akka 引入加拿大一家主要的电信公司，帮助该公司为客户提供容错性更高、响应更及时的软件。除此之外，他还为该公司中的一些团队教授 Akka、函数式以及并发编程等知识。

关于译者

诸豪文，网名 clasnake，毕业于清华大学，现为全职软件开发工程师，常用的开发语言有 Java、Scala、JavaScript 和 Python。其个人博客地址为 http://clasnake.net。他也是开源项目 Swagger 的贡献者，并译有《Python 网络编程》（第 3 版）一书。

技术审核

Taylor Jones 是一名全栈软件工程师，精通基于 Java 的 Web 应用程序开发，目前在 Cisco Systems 工作。他很喜欢使用开源技术设计并构建复杂的应用程序。

致谢

我在这里想要感谢一些人，你们对我的价值观的形成有着很大的影响，并且给予我不断的支持。

首先，要感谢我的妻子 Kate，谢谢你在我撰写本书期间以及做一些疯狂项目的时候对我的支持。如果没有你不断的支持、耐心和照顾，我就不可能改变我的职业生涯去做我喜欢的事情，也不可能写出来这本书。我们终于完成了。现在是时候去粉刷一下墙壁，修缮一下屋子，看着电影休息一下了！

感谢我的祖父母和父母，你们一直告诉我，只要下定了决心，什么事都难不倒我。谢谢你们的建议。你们是对的。

感谢我的 mDialog/Google 团队，感谢你们的审阅和批评。能有机会和你们共事，我觉得非常幸运。特别要感谢 Chris，感谢你相信我的兴趣足以帮助我成长为一个合格的工程师，也感谢你一直都希望团队能够给予我支持。

感谢 Graig 和 Herb，感谢你们对我的启蒙。如果我没有在 17 岁的时候编写冒泡排序、画像素圆或是移植客户数据库，那么我自己都不确定是不是能像现在一样找到自己这么热爱的工作。

Jason Goodwin

前言

本书将尝试帮助入门级、中级以及高级读者理解基本的分布式计算概念，并且展示如何使用 Akka 来构建具备高容错性、可以横向扩展的分布式网络应用程序。Akka 是一个强大的工具集，提供了很多选项，可以对在本地机器上处理或网络远程机器上处理的某项工作进行抽象封装，使之对开发者不可见。本书将介绍各种概念，帮助读者理解网络上各系统进行交互的困难之处，并介绍如何使用 Akka 提供的解决方案来解决这些问题。

编写本书的过程也是作者自我学习、自我发现的过程。希望读者也能一起分享这些知识。作者在工作中有很多使用 Java 8 和 Scala 来编写 Akka 应用程序的经验，但是在编写本书的过程中，学到了很多更深入的 Akka 细节。这本书很好地介绍了为什么要使用 Akka，以及如何使用 Akka，并且展示了如何使用 Akka 工具集来开始构建可扩展的分布式应用程序。本书并不仅仅是官方文档的重复，更涉及了许多作为当代程序员要成功构建能够处理扩展性相关问题的系统时应该要理解的重要话题和方法。

本书涉及的内容

第 1 章　初识 Actor：Akka 工具集以及 Actor 模型的介绍。

第 2 章　Actor 与并发：响应式编程。Actor 与 Future 的使用。

第 3 章　传递消息：消息传递模式。

第 4 章　Actor 的生命周期——处理状态与错误：Actor 生命周期、监督机制、Stash/Unstash、Become/Unbecome 以及有限自动机。

第 5 章　纵向扩展：并发编程、Router Group/Pool、Dispatcher、阻塞 I/O 的处理以及 API。

第 6 章　横向扩展——集群化：集群、CAP 理论以及 Akka Cluster。

第 7 章　处理邮箱问题：加大邮箱负载、不同邮箱的选择、熔断机制。

第 8 章　测试与设计：行为说明、领域驱动设计以及 Akka Testkit。

第 9 章　尾声：其他 Akka 特性。下一步需要学习的知识。

阅读本书的前提条件

读者需要一台能够安装各种工具的计算机，比如 Java JDK8（用于 Java 开发）或是 Java JDK6（用于 Scala 开发）。除此之外，还需要 SBT 简单构建工具（Simple Build Tool）或是 Typesafe Activator（已经包含 SBT）。本书会介绍这些工具的安装。

本书的目标读者

本书面向想要构建满足大规模用户需求的应用程序的初级至中级 Java 或 Scala 开发者。如果应用程序在处理日益增加的用户量以及数据量时需要满足高性能要求，那么建议阅读本书。本书可以让我们只编写更少、更简单的代码就轻松构建并扩展网络应用程序，给用户提供更强大的功能。

读者反馈

我们欢迎读者的反馈。对本书的任何看法敬请告知我们。读者反馈对我们至关重要，可以帮助我们出版读者真正能够从中获益的书籍。

如果有普通的反馈，请直接向 feedback@packtpub.com 发送电子邮件，并且在邮件标题中提及书名。

如果有您精通的话题并且有兴趣撰写书籍或是向某本书做贡献，请从 www.packtpub.com/authors 查阅作者手册。

客户支持

您是 Packt 书籍的拥有者，所以我们提供了很多服务，帮助您从所购买的书中获得最大的收获。

下载示例代码

读者可以使用自己的账户从 http://www.packtpub.com 下载购买过的所有 Packt Publishing 书籍的示例代码文件。如果从别处购买了本书，那么可以访问 http://www.packtpub.com/support 并进行注册，我们会通过电子邮件将文件发送给您。

下载本书的彩色图片

我们还提供了一个 PDF 文件，其中包含本书中使用的截屏/图表的彩色图片。这些彩色图片可以帮助读者更好地理解输出结果中的不同之处。读者可以从 https://www.packtpub.com/sites/default/files/downloads/LearningAkka_ColoredImages.pdf 处下载该文件。

电子书、折扣以及其他

Packt 出版社为出版的每一本书都提供了 PDF 和 ePub 的电子书版本。读者可以从 www.packtpub.com 获取电子书版本。纸质书的客户在购买电子书时可以享受折扣优惠。请通过 customercare@packtpub.com 联系我们，获取更多细节。

您还可以在 www.packtpub.com 阅读更多免费的技术文章，订阅免费新闻，并接收到大量 Packt 书籍以及电子书的折扣优惠。

问题

如果对本书有任何问题，请通过 questions@packtpub.com 联系我们，我们将尽最大的努力解决您的问题。

目录

第 1 章
初识 Actor

1.1 本章概述

Actor 模型是一种并发计算的理论模型，而 Akka 的核心其实是 Actor 模型的一种实现。在本章中，我们将通过了解 Akka 和 Actor 模型的历史来介绍 Akka 的核心概念。这会帮助读者更好地理解 Akka 到底是什么，以及 Akka 试图要解决什么样的问题。其次，本章中将重复使用同一个例子来阐述本书的目的。

在介绍了上面这些概念后，本章将会把篇幅放在开发环境及工具的配置方法上。我们将配置好机器的环境，集成开发环境（Integrated Development Environment，IDE）并介绍第一个 Akka 项目（包括该项目的单元测试）。

1.2 什么是 Akka

本节将介绍 Akka 和 Actor 模型。Akka 一词据说来源于瑞典的一座山，我们说到 Akka 时，通常是指一个分布式工具集，用于协调远程计算资源来进行一些工作。Akka 是 Actor 并发模型的一种现代化实现。现在的 Akka 可以认为是从许多其他技术发展演化而来的，它借鉴了 Erlang 的 Actor 模型实现，同时又引入了许多新特性，帮助构建能够处理如今大规模问题的应用程序。

1.2.1 Actor 模型的起源

为了更好地理解 Akka 的含义及其使用方法，我们将快速地了解 Actor 模型的历史，理解 Actor 模型的含义，以及它是如何一步一步发展到如今的 Akka 这样一个用于构建高容错性分布式系统的框架。

Actor 并发模型最早出现于一篇叫作《A Universal Modular Actor Formalism for Artificial Intelligence》的论文，该论文发表于 1973 年，提出了一种并发计算的理论模型，Actor 就源于该模型。我们将在本节中学习 Actor 模型的特性，理解它的优点，能够在并发计算中帮助我们解决共享状态带来的常见问题。

1.2.2　什么是 Actor

首先，让我们来定义什么是 Actor。在 Actor 模型中，Actor 是一个并发原语；更简单地说，可以把一个 Actor 看作是一个工人，就像能够工作或是处理任务的进程和线程一样。把 Actor 看成是某个机构中拥有特定职位及职责的员工可能会对理解有所帮助。比如说一个寿司餐馆。餐馆的职员需要做各种各样不同的工作，给客人准备餐盘就是其中之一。

1.2.3　Actor 和消息传递

在面向对象编程语言中，对象的特点之一就是能够被直接调用：一个对象可以访问或修改另一个对象的属性，也可以直接调用另一个对象的方法。这在只有一个线程进行这些操作时是没有问题的，但是如果多个线程同时读取并修改同一个值，那么可能就需要进行同步并加锁。

Actor 和对象的不同之处在于其不能被直接读取、修改或是调用。反之，Actor 只能通过消息传递的方式与外界进行通信。简单来说，消息传递指的是一个 Actor 可以接收消息（在我们的例子中该消息是一个对象），本身可以发送消息，也可以对接收到的消息作出回复。尽管我们可以将这种方式与向某个方法传递参数并接收返回值进行类比，但是消息传递与方法调用在本质上是不同的：消息传递是异步的。无论是处理消息还是回复消息，Actor 对外界都没有依赖。

Actor 每次只同步处理一个消息。邮箱本质上是等待 Actor 处理的一个工作队列，如图 1-1 所示。处理一个消息时，为了能够做出响应，Actor 可以修改内部状态，创建更多 Actor 或是将消息发送给其他 Actor。

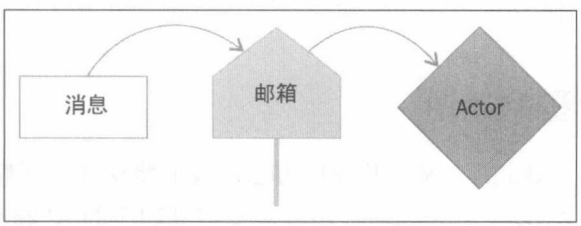


图 1-1