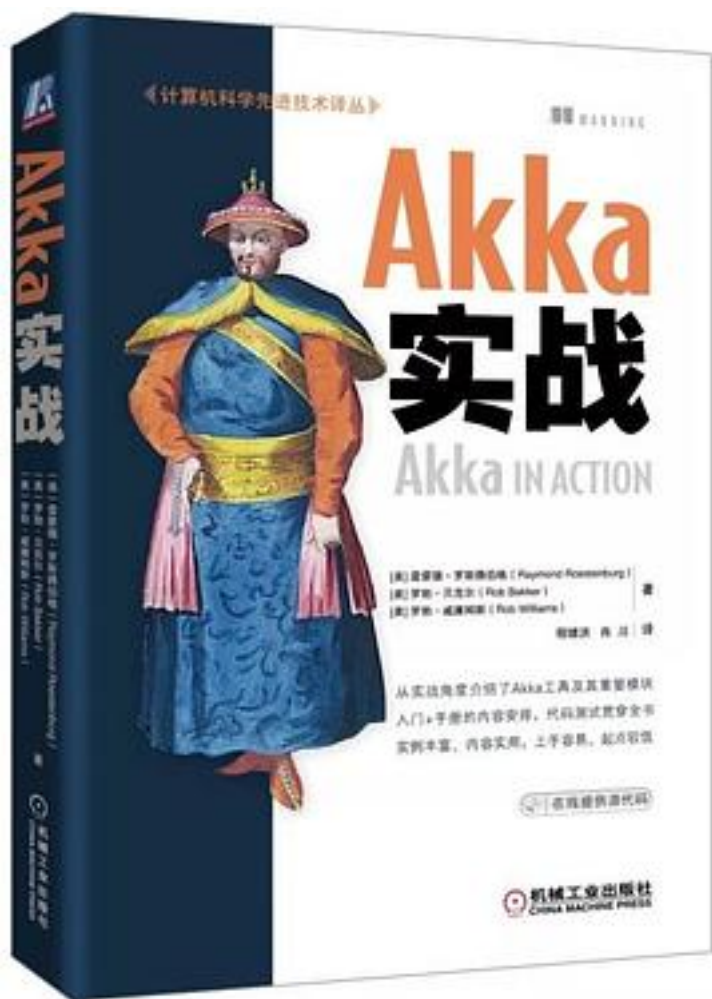


# Akka实战



[Akka实战\\_下载链接1](#)

著者:Raymond Roestenburg

出版者:机械工业出版社

出版时间:2018-12

装帧:平装

isbn:9787111613428

Akka是JAVA虚拟机JVM平台上构建高并发、分布式和容错应用的工具包和运行库，同

时提供了Scala和Java的开发接口。本书主要介绍了Akka的Actor开发模型，并行编程、消息传递、路由功能、集群、持久化等内容，还介绍了Akka的配置、系统集成和性能分析与度量等有关知识，全面介绍了Akka的主要功能，并给出了丰富的实例。

本书可作为程序员、软件工程师和架构师关于开发分布式并行应用的参考，也可以作为高等院校分布式并行开发的教材还可以作为对于分布式并行开发感兴趣的读者的入门参考书。

作者介绍:

Raymond Roestenburg is an experienced software craftsman, polyglot programmer, and software architect. He is an active member of the Scala community, an Akka committer, and contributed to the Akka-Camel module.

Rob Bakker is an experienced developer focused on concurrent back-end systems and system integration. He has used Scala and Akka in production from version 0.7.

Rob Williams is founder of ontometrics, a practice that has been focused on Java solutions that include machine learning. He first used Actor-based programming a decade ago, and found Akka while looking for someone who had given thought to the idea of supervision, and has used it for several projects since.

目录: 目录

序I

译者序II

致谢III

关于本书IV

1章 Akka简介1

1.1 什么是Akka? 3

1.2 Actor简介4

1.3 两种扩展方法：建立我们的实例4

1.4 传统扩展5

1.4.1 传统扩展和持久性：一切移入数据库6

1.4.2 传统扩展和交互应用：轮询8

1.4.3 传统扩展和交互应用：Web服务9

1.5 用Akka进行扩展10

1.5.1 用Akka扩展和持久化：发送和接收消息11

1.5.2 用Akka扩展和交互应用：消息推送13

1.5.3 用Akka扩展和容错：异步解耦14

1.5.4 Akka方式：发送和接收消息15

1.6 Actor：向上和向外扩展的编程模型16

1.6.1 异步模型16

1.6.2 Actor操作17

1.7 Akka Actor20

1.7.1 ActorSystem20

1.7.2 ActorRef、邮箱和Actor21

1.7.3 分发器21

1.7.4 Actor和网络24

1.8 总结24

2章 搭建和运行25

2.1 克隆、构建和测试接口25

2.1.1 用sbt进行构建26

|                                |     |
|--------------------------------|-----|
| 2.1.2快进到GoTicks.com REST服务器    | 28  |
| 2.2 探索应用中的app                  | 32  |
| 2.2.1 app结构                    | 32  |
| 2.2.2 处理销售的Actor: TicketSeller | 39  |
| 2.2.3 BoxOffice Actor          | 40  |
| 2.2.4 RestApi                  | 43  |
| 2.3 部署到云上                      | 46  |
| 2.3.1 在Heroku上创建app            | 46  |
| 2.3.2 在Heroku上部署并运行            | 48  |
| 2.4 总结                         | 49  |
| 3章 Actor测试驱动开发                 | 50  |
| 3.1 测试Actor                    | 50  |
| 3.2 单向消息                       | 52  |
| 3.2.1 SilentActor实例            | 52  |
| 3.2.2 SendingActor实例           | 57  |
| 3.2.3 SideEffectingActor实例     | 63  |
| 3.3 双向消息                       | 66  |
| 3.4 总结                         | 68  |
| 4章 容错                          | 69  |
| 4.1 容错是什么 (不是什么) ?             | 69  |
| 4.1.1 普通对象与异常                  | 70  |
| 4.1.2 Let it crash             | 73  |
| 4.2 Actor生命周期                  | 76  |
| 4.2.1 启动事件                     | 77  |
| 4.2.2 停止事件                     | 78  |
| 4.2.3 重启事件                     | 78  |
| 4.2.4 生命周期综合                   | 80  |
| 4.2.5 生命周期监控                   | 82  |
| 4.3 监视                         | 83  |
| 4.3.1 监视器层次结构                  | 83  |
| 4.3.2 预定义策略                    | 85  |
| 4.3.3 自定义策略                    | 87  |
| 4.4 总结                         | 94  |
| 5章 Futures                     | 95  |
| 5.1 Future的应用实例                | 95  |
| 5.2 Future无阻塞                  | 99  |
| 5.3 Future错误处理                 | 106 |
| 5.4 Future组合                   | 111 |
| 5.5 Future组合Actor              | 122 |
| 5.6 总结                         | 123 |
| 6章 一个分布式Akka app               | 125 |
| 6.1 向外扩展                       | 125 |
| 6.1.1 通用网络术语                   | 125 |
| 6.1.2 采用分布式编程模型的原因             | 126 |
| 6.2 远程扩展                       | 127 |
| 6.2.1 把GoTicks.com app改造成分布式应用 | 128 |
| 6.2.2 远程REPL活动                 | 129 |
| 6.2.3 远程查找                     | 135 |
| 6.2.4 远程部署                     | 143 |
| 6.2.5 多JVM测试                   | 149 |
| 6.3 总结                         | 156 |
| 7 章 配置、日志和部署                   | 158 |
| 7.1 配置                         | 158 |
| 7.1.1 尝试Akka配置                 | 158 |

|                                |     |
|--------------------------------|-----|
| 7.1.2 使用默认值                    | 162 |
| 7.1.3 Akka配置                   | 165 |
| 7.1.4 多系统                      | 166 |
| 7.2 日志                         | 168 |
| 7.2.1 Akka中的日志记录               | 168 |
| 7.2.2 使用日志                     | 170 |
| 7.2.3 Akka的日志控制                | 171 |
| 7.3 部署基于Actor的应用               | 173 |
| 7.4 总结                         | 178 |
| 8章 Actor的结构模式                  | 179 |
| 8.1 管道和过滤器                     | 179 |
| 8.1.1 企业集成模式：管道和过滤器            | 179 |
| 8.1.2 Akka中的管道和过滤器             | 180 |
| 8.2 企业集成模式：分发-收集模式             | 185 |
| 8.2.1 适用性                      | 185 |
| 8.2.2 Akka处理并行任务               | 187 |
| 8.2.3 使用接收者列表实现分发组件            | 188 |
| 8.2.4 使用聚合器模式实现收集组件            | 189 |
| 8.2.5 组合组件实现分发-收集模式            | 198 |
| 8.3 企业集成模式：路由表模式               | 199 |
| 8.4 总结                         | 205 |
| 9章 路由消息                        | 207 |
| 9.1 企业集成路由模式                   | 207 |
| 9.2 使用Akka Router实现负载均衡        | 208 |
| 9.2.1 Akka Router池             | 211 |
| 9.2.2 Akka Router群组            | 217 |
| 9.2.3 ConsistentHashing Router | 225 |
| 9.3 用Actor实现路由模式               | 229 |
| 9.3.1 基于内容的路由                  | 229 |
| 9.3.2 基于状态的路由                  | 230 |
| 9.3.3 Router的实现                | 233 |
| 9.4 总结                         | 234 |
| 10章 消息通道                       | 235 |
| 10.1 通道类型                      | 235 |
| 10.1.1 点对点通道                   | 235 |
| 10.1.2 发布-订阅通道                 | 236 |
| 10.2 特殊通道                      | 245 |
| 10.2.1 死信                      | 245 |
| 10.2.2 保证投递                    | 248 |
| 10.3 总结                        | 254 |
| 11章 有限状态机和代理                   | 256 |
| 11.1 使用有限状态机                   | 256 |
| 11.1.1 有限状态机简介                 | 256 |
| 11.1.2 创建FSM模型                 | 257 |
| 11.2 FSM模型的实现                  | 258 |
| 11.2.1 实现转换                    | 259 |
| 11.2.2 实现入口动作                  | 264 |
| 11.2.3 FSM定时器                  | 270 |
| 11.2.4 FSM的终止                  | 273 |
| 11.3 使用代理实现共享状态                | 274 |
| 11.3.1 使用代理简单地共享状态             | 274 |
| 11.3.2 等待状态更新                  | 276 |
| 11.4 总结                        | 277 |
| 12章 系统集成                       | 278 |

|                            |      |
|----------------------------|------|
| 12.1 消息终端                  | 278  |
| 12.1.1 归一化                 | 279  |
| 12.1.2 规范数据模型              | 280  |
| 12.2 使用Apache Camel实现终端    | 282  |
| 12.2.1 创建从外部系统接收消息的消费者终端   | 283  |
| 12.2.2 实现生产者向外部系统发送消息      | 291  |
| 12.3 实现HTTP接口              | 297  |
| 12.3.1 HTTP实例              | 298  |
| 12.3.2 用Akka-https实现REST终端 | 300  |
| 12.4 总结                    | 307  |
| 13章 流                      | 309  |
| 13.1 基本流处理                 | 309  |
| 13.1.1 使用源和接收器复制文件         | 313  |
| 13.1.2 实体化可运行图             | 316  |
| 13.1.3 用Flow处理事件           | 321  |
| 13.1.4 处理流中的错误             | 324  |
| 13.1.5 用BidiFlow创建协议       | 326  |
| 13.2 HTTP流                 | 330  |
| 13.2.1 接收HTTP流             | 330  |
| 13.2.2 HTTP响应流             | 333  |
| 13.2.3 内容类型和协调的自定义编组与解组    | 334  |
| 13.3 用Graph DSL进行扇入和扇出     | 340  |
| 13.3.1 广播流                 | 340  |
| 13.3.2 合并流                 | 343  |
| 13.4 协调生产者和消费者             | 347  |
| 13.4.1 使用缓冲区               | 347  |
| 13.5 图的速率隔离                | 350  |
| 13.5.1 对较慢的消费者，对事件进行汇总     | 351  |
| 13.5.2 快速消费者的扩展度量          | 351  |
| 13.6 总结                    | 352  |
| 14章 集群                     | 353  |
| 14.1 为什么使用集群？              | 353  |
| 14.2 集群成员关系                | 355  |
| 14.2.1 加入集群                | 356  |
| 14.2.2 离开集群                | 364  |
| 14.3 集群作业处理                | 369  |
| 14.3.1 启动集群                |      |
| • • • • •                  | (收起) |

[Akka实战\\_下载链接1](#)

标签

计算机

分布式

akka

这翻译糟蹋原版书啊

孔夫子

中间件

评论

-----  
[Akka实战\\_下载链接1](#)

书评

-----  
[Akka实战\\_下载链接1](#)