

УДК 004.424  
ББК 32.972  
Р43

Р43 Раймонд Рестенбург, Роб Баккер, Роб Уильямс

Акка в действии / пер. с англ. А. Н. Киселев – М.: ДМК Пресс, 2018. – 522 с.: ил.

**ISBN 978-5-97060-642-1**

В книге рассказывается о фреймворке Akka и описываются его наиболее важные модули. Большое внимание уделено модели программирования с акторами и модулям поддержки акторов, часто используемых при создании конкурентных и распределенных приложений. Продемонстрированы подходы к разработке через тестирование и приемы развертывания и масштабирования отказоустойчивых систем. Во всех примерах книги используется язык программирования Scala.

Издание адресовано разработчикам на Java и Scala, желающим научиться создавать приложения с использованием фреймворка Akka.

УДК 004.424  
ББК 32.972

Original English language edition published by Manning Publications. Copyright © 2017 by Manning Publications. Russian language edition copyright © 2018 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-61729-101-2 (англ.)  
ISBN 978-5-97060-642-1 (рус.)

Copyright © 2017 by Manning Publications Co.  
© Оформление, перевод на русский язык,  
издание, ДМК Пресс, 2018

# Оглавление

<b>Предисловие .....</b>	<b>11</b>
<b>Благодарности .....</b>	<b>12</b>
<b>О книге .....</b>	<b>13</b>
Кому адресована эта книга .....	13
Содержание книги .....	13
Соглашения об оформлении программного кода .....	15
Требования к программному обеспечению .....	15
Автор в сети.....	15
Об авторах .....	16
Об иллюстрации на обложке.....	16
<b>Глава 1. Введение в Akka .....</b>	<b>18</b>
1.1. Что такое Akka?.....	22
1.2. Акторы: краткий обзор .....	22
1.3. Два подхода к масштабированию: подготовка примера .....	24
1.4. Традиционное масштабирование .....	26
1.4.1. Традиционный подход к масштабированию и хранению: переместить все в базу данных .....	26
1.4.2. Традиционное масштабирование и интерактивная работа: опрос.....	29
1.4.3. Традиционное масштабирование и интерактивная работа: обработка ошибок .....	31
1.5. Масштабирование с Akka.....	32
1.5.1. Подход к масштабированию и хранению с Akka: отправка и прием сообщений .....	33
1.5.2. Масштабирование с Akka и интерактивная работа: отправка сообщений .....	36
1.5.3. Масштабирование с Akka и отказы: асинхронное разделение .....	37
1.5.4. Подход Akka: отправка и получение сообщений .....	37
1.6. Акторы: универсальная модель программирования .....	39
1.6.1. Модель асинхронного выполнения.....	40
1.6.2. Операции с актерами.....	41
1.7. Акторы Akka .....	44
1.7.1 ActorSystem .....	45
1.7.2. ActorRef, почтовый ящик и актер .....	47
1.7.3. Диспетчеры .....	47
1.7.4. Акторы и сеть .....	49
1.8. В заключение .....	49
<b>Глава 2. Подготовка и запуск .....</b>	<b>51</b>
2.1. Клонирование, сборка и интерфейс тестирования .....	52

2.1.1. Сборка с помощью sbt.....	53
2.1.2. Забегая вперед: REST-сервер GoTicks.com.....	54
2.2. Исследование акторов в приложении.....	59
2.2.1. Структура приложения .....	59
2.2.2. Актор, осуществляющий продажу: TicketSeller.....	64
2.2.3. Актор BoxOffice .....	65
2.2.4. Актор RestApi .....	67
2.3. Вперед, в облако .....	70
2.3.1. Создание приложения в облаке Heroku .....	71
2.3.2. Развертывание и запуск в Heroku .....	72
2.4. В заключение .....	73
<b>Глава 3. Разработка с акторами через тестирование.....</b>	<b>75</b>
3.1. Тестирование акторов.....	76
3.2. Односторонние взаимодействия .....	78
3.2.1. Примеры SilentActor .....	79
3.2.2. Пример SendingActor .....	84
3.2.3. Пример SideEffectingActor .....	89
3.3. Двусторонние взаимодействия.....	92
3.4. В заключение.....	93
<b>Глава 4. Отказоустойчивость.....</b>	<b>95</b>
4.1. Что такое отказоустойчивость.....	95
4.1.1. Простые объекты и исключения .....	98
4.1.2. И пусть падает .....	103
4.2. Жизненный цикл актора .....	107
4.2.1. Событие start .....	107
4.2.2. Событие stop.....	108
4.2.3. Событие restart .....	109
4.2.4. Объединяем фрагменты жизненного цикла вместе.....	111
4.2.5. Мониторинг жизненного цикла .....	113
4.3. Супервизор .....	114
4.3.1. Иерархия супервизора .....	114
4.3.2. Предопределенные стратегии .....	117
4.3.3. Собственные стратегии.....	118
4.4. В заключение.....	124
<b>Глава 5. Объекты Future .....</b>	<b>125</b>
5.1. Примеры использования объектов Future .....	126
5.2. Объекты Future не блокируют выполнение потока .....	131
5.2.1. Объекты Promise – это обещания.....	135
5.3. Обработка ошибок в объектах Future .....	138
5.4. Комбинирование объектов Future .....	143
5.5. Объединение объектов Future с акторами .....	152

5.6. В заключение .....	153
<b>Глава 6. Первое распределенное приложение .....</b>	<b>155</b>
6.1. Горизонтальное масштабирование .....	156
6.1.1. Общая терминология .....	156
6.1.2. Причины использования модели распределенного программирования .....	158
6.2. Горизонтальное масштабирование и удаленные взаимодействия ....	159
6.2.1. Реорганизация приложения GoTicks.com .....	161
6.2.2. Удаленные взаимодействия в REPL .....	161
6.2.3. Удаленный поиск.....	167
6.2.4. Удаленное развертывание .....	175
6.2.5. Тестирование с multi-JVM.....	180
6.3. В заключение .....	186
<b>Глава 7. Настройка, журналирование и развертывание .....</b>	<b>188</b>
7.1. Настройка.....	188
7.1.1. Попытка настройки Akka .....	189
7.1.2. Использование значений по умолчанию.....	192
7.1.3. Настройка Akka .....	195
7.1.4. Настройка для нескольких систем.....	196
7.2. Журналирование .....	199
7.2.1. Журналирование в приложении Akka .....	199
7.2.2. Использование журналирования .....	201
7.2.3. Управление журналированием из Akka .....	202
7.3. Развертывание приложений на основе акторов .....	204
7.4. В заключение .....	208
<b>Глава 8. Шаблоны структуризации акторов.....</b>	<b>210</b>
8.1. Конвейеры и фильтры .....	211
8.1.1. Шаблон: конвейеры и фильтры.....	211
8.1.2. Конвейеры и фильтры в Akka .....	212
8.2. Параллельная обработка дроблением с последующим объединением результатов .....	216
8.2.1. Область применения .....	216
8.2.2. Распараллеливание задач в Akka .....	218
8.2.3. Реализация компонента дробления с использованием списка получателей .....	219
8.2.4. Реализация компонента объединения с использованием агрегатора .....	221
8.2.5. Объединение компонентов в реализацию шаблона параллельной обработки дроблением .....	227
8.3. Маршрутизация.....	229
8.4. В заключение .....	234

<b>Глава 9. Маршрутизация сообщений</b>	<b>236</b>
9.1. Шаблон «Маршрутизатор»	237
9.2. Балансировка нагрузки с помощью маршрутизаторов Akka	238
9.2.1. Маршрутизатор с пулом	242
9.2.2. Маршрутизатор с группой	250
9.2.3. Маршрутизатор ConsistentHashing	257
9.3. Реализация шаблона маршрутизатора с применением акторов	262
9.3.1. Маршрутизация по содержимому	262
9.3.2. Маршрутизация на основе состояния	263
9.3.3. Реализации маршрутизаторов	265
9.4. В заключение	266
<b>Глава 10. Каналы обмена сообщениями</b>	<b>268</b>
10.1. Типы каналов	269
10.1.1. Точка-точка	269
10.1.2. Издатель/подписчик	270
10.2. Специальные каналы	280
10.2.1. DeadLetter	281
10.2.2. Гарантированная доставка	283
10.3. В заключение	289
<b>Глава 11. Конечные автоматы и агенты</b>	<b>291</b>
11.1. Использование конечного автомата	292
11.1.1. Краткое введение в конечные автоматы	292
11.1.2. Создание модели конечного автомата	294
11.2. Реализация модели конечного автомата	295
11.2.1. Реализация переходов	296
11.2.2. Реализация действий при входе в состояния	301
11.2.3. Таймеры в конечном автомате	305
11.2.4. Завершение конечного автомата	308
11.3. Реализация общего состояния с помощью агентов	309
11.3.1. Простой доступ к общим данным с помощью агентов	310
11.3.2. Ожидание изменения состояния	312
11.4. В заключение	313
<b>Глава 12. Интеграция с другими системами</b>	<b>315</b>
12.1. Конечные точки сообщений	315
12.1.1. Нормализатор	317
12.1.2. Модель канонических данных	319
12.2. Реализация конечных точек с использованием Apache Camel	322
12.2.1. Реализация конечной точки-потребителя для приема сообщений из внешней системы	323
12.2.2. Реализация конечной точки-производителя для отправки сообщений во внешнюю систему	330

12.3. Реализация HTTP-интерфейса .....	335
12.3.1. Пример HTTP-интерфейса.....	336
12.3.2. Реализация конечной точки REST на основе akka-http .....	338
12.4. В заключение .....	344
<b>Глава 13. Потокковые приложения.....</b>	<b>346</b>
13.1. Основы потоковой обработки .....	347
13.1.1. Копирование файлов.....	351
13.1.2. Материализация запускаемых графов .....	355
13.1.3. Обработка событий в потоке .....	360
13.1.4. Обработка ошибок в потоках.....	364
13.1.5. Создание протокола с BidiFlow.....	366
13.2. Потокковая передача данных через HTTP .....	369
13.2.1. Прием потока данных по HTTP .....	370
13.2.2. Возврат потока данных по HTTP .....	372
13.2.3. Согласование контента .....	373
13.3. Ветвление и слияние со специализированным языком описания графов .....	378
13.3.1. Ветвление потоков .....	378
13.3.2. Слияние потоков .....	381
13.4. Посредничество между производителями и потребителями.....	384
13.4.1. Использование буферов.....	385
13.5. Обособление частей графа, действующих с разной скоростью .....	389
13.5.1. Медленный потребитель, накопление событий в блоках .....	389
13.5.2. Быстрый потребитель, дополнительные показатели .....	390
13.6. В заключение .....	391
<b>Глава 14. Кластеры.....</b>	<b>393</b>
14.1. Зачем нужны кластеры? .....	393
14.2. Членство в кластере .....	395
14.2.1. Присоединение к кластеру .....	396
14.2.2. Выход из кластера .....	404
14.3. Обработка заданий в кластере .....	410
14.3.1. Запуск кластера .....	412
14.3.2. Распределение заданий с использованием маршрутизаторов.....	414
14.3.3. Надежная обработка заданий.....	417
14.3.4. Тестирование кластера.....	424
14.4. В заключение .....	428
<b>Глава 15. Хранимые акторы.....</b>	<b>430</b>
15.1. Восстановление состояния с технологией Event Sourcing.....	432
15.1.1. Обновление записей на месте .....	432
15.1.2. Сохранение состояния без изменения.....	433
15.1.3. Event Sourcing для акторов .....	435

15.2. Хранимые акторы .....	436
15.2.1. Хранимый актор .....	437
15.2.2. Тестирование .....	441
15.2.3. Моментальные снимки .....	443
15.2.4. Запрос хранимых событий .....	449
15.2.5. Сериализация .....	451
15.3. Кластер на основе хранимых акторов .....	457
15.3.1. Расширение cluster singleton .....	461
15.3.2. Расширение cluster sharding .....	465
15.4. В заключение .....	470
<b>Глава 16. Советы по повышению производительности .....</b>	<b>471</b>
16.1. Анализ производительности .....	472
16.1.1. Производительность системы .....	472
16.1.2. Показатели производительности .....	474
16.2. Оценка производительности акторов .....	477
16.2.1. Сбор данных в почтовом ящике .....	478
16.2.2. Сбор и обработка данных .....	485
16.3. Улучшение производительности устранением узких мест .....	487
16.4. Настройка диспетчера .....	489
16.4.1. Выявление проблем с пулами потоков .....	489
16.4.2. Использование нескольких экземпляров диспетчеров .....	491
16.4.3. Изменение размера пула потоков статически .....	493
16.4.4. Изменение размера пула потоков динамически .....	496
16.5. Изменение поведения механизма освобождения потоков .....	498
16.5.1. Ограничения механизма освобождения потоков .....	500
16.6. В заключение .....	502
<b>Глава 17. Заглядывая вперед .....</b>	<b>504</b>
17.1. Модуль akka-typed .....	505
17.2. Akka Distributed Data .....	509
17.3. В заключение .....	509
<b>Предметный указатель .....</b>	<b>511</b>