

《七周七并发模型》

图书基本信息

书名：《七周七并发模型》

13位ISBN编号：9787115386064

出版时间：2015-3

作者：Paul Butcher

页数：244

译者：黄炎

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《七周七并发模型》

内容概要

借助Java、Go等多种语言的特长，深度剖析所有主流并发编程模型

基于锁和线程的并发模型是目前最常用的一种并发模型，但是并发编程模型不仅仅只有这一种，本书几乎涵盖了目前所有的并发编程模型。了解和熟悉各种并发编程模型，在解决并发问题时会有更多思路。

——方腾飞，并发编程网站长

当看到这本书的目录时，我就为之一振。它涉及了当今所有的主流并发编程模型（当然也包括Go语言及其实现的CSP模型）。而在品读之后，我更是有一些激动。其中的大部分章节都深入浅出地描述了某一个并发编程模型的概念、特点、机理以及适用领域。加之有大量相应实现语言的示例呈现，更是让这些模型活灵活现、极易理解。另外，各章之间的衔接会使你更容易把这些知识融会贯通。总之，并发编程乃现代编程技能的必备，而这本书则为并发编程的首选佳作。

——郝林（@特价萝卜），《Go并发编程实战》作者

几十年来，专业的程序员一直在用线程与锁模型来进行并行和并发，但这仅仅是《七周七并发模型》介绍的众多模型之一。当今主流语言都竞相支持不同的并发模型，例如CSP模型、数据并行、函数式编程和Clojure的unified succession model。如果你想紧跟时代，我推荐阅读这本书。

——Stuart Halloway，Cognitect联合创始人

并发编程近年逐渐热起来，Go等并发语言也对并发编程提供了良好的支持，使得并发这个话题受到越来越多人的关注。本书延续了《七周七语言》的写作风格，通过以下七个精选的模型帮助读者了解并发领域的轮廓：线程与锁，函数式编程，Clojure，actor，通信顺序进程，数据级并行，Lambda架构。书中每一章都设计成三天的阅读量。每天阅读结束都会有相关练习，巩固并扩展当天的知识。每一章均有复习，用于概括本章模型的优点和缺陷。

本书适合所有想了解并发的程序员。

《七周七并发模型》

作者简介

Paul Butcher 资深程序员，涉猎广泛，从单片机编码到高级声明式编程无所不精。Paul是一位少年天才，8岁时就已经开始在8位机上编写游戏。最近几年他开始痴迷于赛车，想要去叫板汉密尔顿。除本书外，还著有在亚马逊获得全五星好评的《软件调试修炼之道》。

黄炎 现供职于上海爱可生，从事数据库高可用软件开发。简介和代码一样简洁。

书籍目录

译者序

推荐序

致谢

前言

第1章 概述

1.1 并发？还是并行？

1.2 并行架构

1.3 并发：不只是多核

1.4 七个模型

第2章 线程与锁

2.1 简单粗暴

2.2 第一天 互斥和内存模型

2.3 第二天 超越内置锁

2.4 第三天 站在巨人的肩膀上

2.5 复习

第3章 函数式编程

3.1 若不爽，就另辟蹊径

3.2 第一天 抛弃可变状态

3.3 第二天 函数式并行

3.4 第三天 函数式并发

3.5 复习

第4章 Clojure之道——分离标识与状态

4.1 混搭的力量

4.2 第一天 原子变量与持久数据结构

4.3 第二天 代理和软件事务内存

4.4 第三天 深入学习

4.5 复习

第5章 Actor

5.1 更加面向对象

5.2 第一天 消息和信箱

5.3 第二天 错误处理和容错性

5.4 第三天 分布式

5.5 复习

第6章 通信顺序进程

6.1 万物皆通信

6.2 第一天 channel和go块

6.3 第二天 多个channel与IO

6.4 第三天 客户端CSP

6.5 复习

第7章 数据并行

7.1 隐藏在笔记本电脑中的超级计算机

7.2 第一天 GPGPU编程

7.3 第二天 多维空间与工作组

7.4 第三天 OpenCL和OpenGL——全部在GPU上运行

7.5 复习

第8章 Lambda架构

8.1 并行计算搞定大数据

《七周七并发模型》

8.2 第一天 MapReduce

8.3 第二天 批处理层

8.4 第三天 加速层

8.5 复习

第9章 圆满结束

9.1 君欲何往

9.2 未尽之路

9.3 越过山丘

参考书目

《七周七并发模型》

精彩短评

- 1、今年读到超多好书，但这本书仍是俺2015年遇到的最最佳书之一。
 - 2、快速翻过
 - 3、讲的比较简略，封面上写着大大的 Java 和 Go，其实大部分代码都是 Clojure 啦。
 - 4、看了前面两章java的，并发编程入门书籍
 - 5、讲述了不同语言并发设计上的模式和优缺点对比，知识面之广令人佩服，无论熟悉不熟悉closure erlang elixir openc1等语言，都值得一看，思想是可以相互借鉴的
 - 6、quick and straight
 - 7、没读完。其实我更想了解不同并发模型本身的区别，多讲一讲模型本身就好了，本书以示例为主线。感觉欠缺一些提纲挈领的骨架。
 - 8、讲的东西挺多，但是深入看还得自己去找资料。而且讲的并不是很浅显易懂。
 - 9、以前提到并发只知道多线程和锁，其他并发还有很多其他概念和技术，着实拓宽的技术面和见解，很多概念还不是非常清晰，后面还是要不停翻阅作为参考。
 - 10、受限于个人语言基础，只先粗略度一遍。值得后续细读，尤其actor和CSP模型，现在没啥体会
 - 11、2015年唯一一本没看懂的，C10K问题那里过来的，前路漫漫
 - 12、老大评论这本书说，很系统的讲了并发，还讲了GPU
 - 13、并发模型和函数式编程语法介绍参杂在一起挺乱的，快速认识常见并发模型如actor，但具体应用的业务场景就要看别的书了
 - 14、讲了锁（很蛋疼）、函数式不变性在并发里的优势、Actor模型、channel、GPU、lambda架构等。linus说的对，不要高看并发，其实场景很有限
 - 15、closure的语法真绕。。预发层面的东西太多了一点。actor、csp模型的本质的东西讲的不够深入。
 - 16、泛读 了解下并发 感觉这本书可以写得更短一点
 - 17、讲的比较泛，看看结论就好
 - 18、略过时
 - 19、既然是讲并发模型，感觉不应该这么强语言相关。
- 7个并发模型有3个选用的clojure，不熟悉的话只能草草略过。
- PS：就连CSP模型都选的clojure orz
- 20、各种并发模型讲得很清楚 受益匪浅
 - 21、讲Java那一章精彩!
 - 22、虽然书很薄，但是对各个并发模型都讲解的很细致深入。不过最好还是有函数式编程的基础再看。
 - 23、辣鸡
 - 24、对我这种Java er来说，第一二章还能看得清楚，从第三章函数式编程开始，整个人就感觉不大好了，看来还得加强自己的内功，：P
 - 25、暂时4星，只读了Java部分，在Clojure和Scala的部分由于不熟悉暂时打住，闲暇时学习Clojure后再来继续
 - 26、太多clojure的内容了，副标题的go也是骗人的
 - 27、感觉还不错，挺薄的一本书，看完挺长见识的。每个模型没有很深入，但基本点都说到了
 - 28、挺不错的，简单扼要
 - 29、没读完，书中实例讲解为主，原理性的东西欠缺。讲线程和锁那一章还不错。
 - 30、嗯，已了然于胸
 - 31、偶有惊艳，闲时可以翻看。
 - 32、不错的介绍分布式和开发框架的书籍啦
 - 33、讲的不错，有大概了解
 - 34、扩展视野的书，线程&锁，Lmbada部分讲的好。
 - 35、信息量确实很大，但只适合浮光掠影的阅读。
 - 36、P53
 - 37、前几章写得更好。Clojure讲的比较多，也许以后会用起来。

《七周七并发模型》

- 38、这本书思路很开阔，值得一看，当然要全部搞懂是不可能的。我觉得对于一个复杂的话题，这种多看看多想想是个很好的模式，钻牛角尖很容易帆船。
- 39、整体还是不错的，java的部分基本面已经覆盖到了，更多的新意在clojure的函数式编程以及go和CSP模型，clojure底层还是依赖jvm，所以其中的原子变量，copyonwrite,future基本还是java中的概念，很好奇actor的实现，如何做到那么轻量级（甚至每个操作一个go），又能做到并发，难道是共用一个全局的线程池？Erlang的actor模型是个亮点，非常简洁和有效，而且还支持分布式，牛逼！还有那个GPU太底层，一般的非算法场景也用不到，最后说到的hadoop和Storm就是老生长谈，但值得关注
- 40、一天看完。。。语言知识太多。。。golang那个标题骗人。。。感觉最大的收获在模型的一些思想
- 41、讲的比较泛，提纲挈领的作用，要用erlang做示例语言就好了
- 42、Quick scan
- 43、有点意思，但是觉得怪怪的，不是特别接地气
- 44、五一在南山书城大概翻了下，actor那章讲的比那些scala的书好太多了
- 45、从最基本的线程讲起，谈到了我们在处理并发问题需要面临的问题。深入浅出，值得推荐。
- 46、打开了新世界的大门！
- 47、应该专注于模型本身去讲的，然而作者一直尝试解释每一门语言的语法特点，有点吃力不讨好的感觉，最后，代码片段中不断出现的《银河系漫游指南》的梗实在是太欢乐了。
- 48、感觉在讲几种并发模型在几种语言的应用，而不是原理，这类书大概就这样吧
- 49、一天看完
- 50、用clojure的实现细节很多。函数式编程和并发的关系不是很清楚，只是稍微提到了STM、原子操作、固定类型等。很多并发的实现和clojure语言的实现相关。CSP讲得不是很透彻，没有用golang还是用了clojure来讲，感觉有点怪。

《七周七并发模型》

章节试读

《七周七并发模型》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com