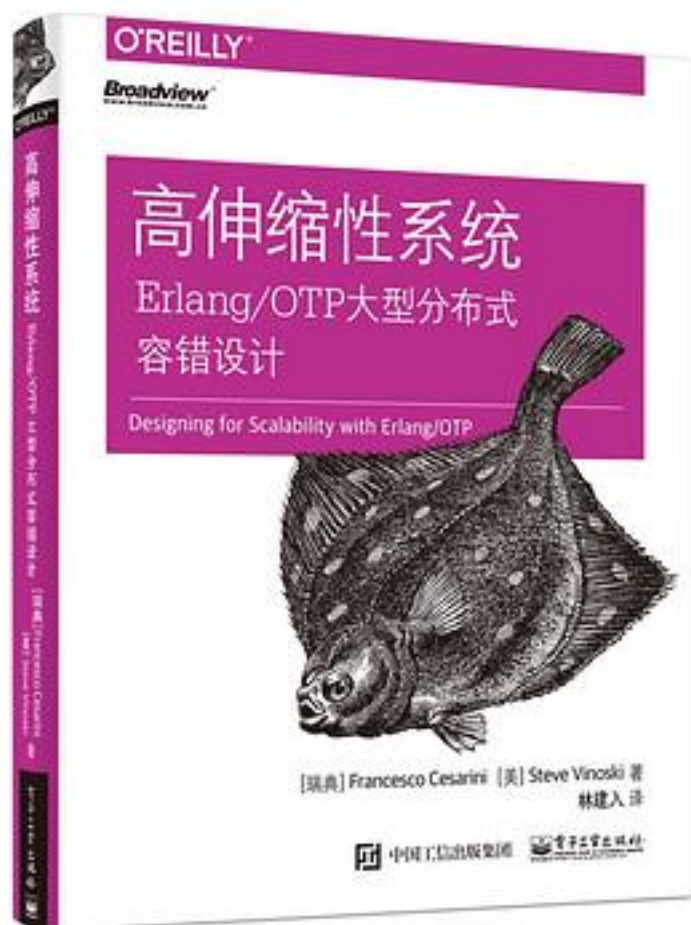


高伸缩性系统：Erlang/OTP大型分布式容错设计



[高伸缩性系统：Erlang/OTP大型分布式容错设计_下载链接1](#)

著者:【英】 Francesco Cesarini

出版者:电子工业出版社

出版时间:2018-6

装帧:平装

isbn:9787121337475

《高伸缩性系统：Erlang/OTP大型分布式容错设计》是一本罕见的站在核心设计者而非普通开发者角度介绍 Erlang/OTP系统的权威书籍。两位作者均是深耕分布式计算领域超过20年的专家。《

高伸缩性系统：Erlang/OTP大型分布式容错设计》内容兼具深度与广度，不仅带领读者通过一步步实践的方式深入剖析了Erlang/OTP中各类核心进程的行为模式的设计原理，并且还介绍了特殊进程、自定义行为模式、发行包制作等高级主题。除此之外，还用了大量篇幅向读者介绍了Erlang/OTP系统中的设计原则、架构分布式系统的方法，以及在此基础上实现容错和规模伸缩所需了解的相关知识。

对于任何一位渴望基于Erlang/OTP构建出商业级的分布式、高伸缩性、容错型系统的开发者，《高伸缩性系统：Erlang/OTP大型分布式容错设计》都是不容错过的经典之作。

作者介绍:

译者简介

林建入是一家远程工作的资深程序员，熟悉网络协议与分布式系统设计以及其他。优点是开朗、话多，热爱家庭，志存高远，亲切友好。缺点是一有机会就溜去游戏厅打拳皇，玩了20年竟也不腻，我都拍着胸口佩服我自己，但是最近总被爱妻抓回家有点困扰。如果你不是资深硬核小众精英程序员，我不建议你买这本 Erlang 神书，因为它会让你太快地完成工作导致空闲时间过多不知如何是好，或者让你过度沉迷其源码与实现机制，废寝忘食导致身体机能下降和精神过度亢奋。

目录: 第 1章 概述	1
定义问题	2
OTP	4
Erlang	6
工具和库	7
系统设计原则	9
Erlang 节点	10
分布式、基础设施、多核	11
总结	12
通过本书你将学到什么	13
第 2章 Erlang.简介.	18
递归与模式匹配	18
受函数式的影响	22
玩转匿名函数	22
列表推导：生成与测试	23
进程与消息传递	25
不怕出错	30
用于监督的链接与监视器	31
链接	31
监视器	33
记录	34
映射组	37
宏	38
模块升级	39
ETS：Erlang 元素存储	41
分布式 Erlang	44
命名与通信	45
节点间的连接与可见性	45
总结	47
接下来是什么	47

- 第 3 章 行为模式. 49
 - 进程的骨架 49
 - 设计模式 52
 - 回调模块 53
 - 抽取出通用的行为模式 56
 - 启动 server 57
 - client 函数 60
 - server 循环 62
 - server 内部函数 64
 - 通用服务器 65
 - 消息传递：冰山之下 68
 - 总结 71
 - 接下来是什么 72
- 第 4 章 通用型服务器.gen_server. 73
 - gen_server 73
 - behavior 指令 74
 - 启动一个 server 75
 - 消息传递 77
 - 同步式消息传递 78
 - 异步式消息传递 79
 - 其他消息 81
 - 未处理的消息 82
 - 同步客户端 83
 - 终止 84
 - 调用超时 86
 - 死锁 89
 - 通用型 server 的超时问题 90
 - 使 behavior 休眠 92
 - 全局化 92
 - 链接 behavior 94
 - 总结 94
 - 接下来是什么 95
- 第 5 章 深入控制 OTP 行为模式 96
 - sys 模块 96
 - 追踪与记录 96
 - 系统消息 98
 - 你自己的追踪函数 98
 - 统计信息和当前状态 99
 - sys 模块总结 102
 - 分裂时的可选项 103
 - 内存管理与垃圾回收 104
 - 分裂时应该避免使用的可选项 108
 - 超时 109
 - 总结 109
 - 接下来是什么 109
- 第 6 章 有限状态机. 110
 - Erlang 风格的有限状态机 111
 - Coffee FSM 112
 - 硬件桩 114
 - Erlang 版咖啡机 114
 - gen_fsm 118
 - 一个基于行为模式的例子 119
 - 启动 FSM 119
 - 发送事件 123

- 终止 132
- 总结 133
- 亲力亲为 134
- 电话控制器 134
- 让我们测试一下 136
- 接下来是什么 138
- 第 7 章 事件处理器. 139
- 事件 139
- 通用事件管理器/处理器 141
- 启动/停止事件管理器 141
- 添加事件处理器 142
- 删除事件处理器 144
- 发送同步的或异步的事件 145
- 获取数据 148
- 对错误以及无效返回值的处理 150
- 交换事件处理器 152
- 融会贯通 154
- SASL 警报处理器 157
- 总结 159
- 接下来是什么 159
- 第 8 章 监督者 160
- 监督树 161
- OTP 监督者 165
- 监督者行为模式 166
- 启动监督者 166
- 监督者规格 169
- 动态子进程 176
- 非 OTP 兼容进程 184
- 可伸缩性和短期进程 186
- 确定性同步启动 187
- 测试你的监督策略 188
- 与传统方法相比又如何 190
- 总结 190
- 接下来是什么 191
- 第 9 章 OTP.application 192
- OTP application 是如何运行的 193
- OTP application 的结构 194
- 回调模块 198
- 启动和停止 application 198
- application 资源文件 202
- 基站控制器的 application 文件 204
- 启动 application 205
- 环境变量 208
- application 的类型与终止策略 210
- 分布式 application 211
- 分阶段启动 215
- 内含型 application 217
- 内含型 application 的分阶段启动 217
- 将监督者与 application 组合到一起 219
- SASL 应用 220
- 进度报告 224
- 错误报告 225
- 崩溃报告 226
- 监督者报告 227

- 总结 228
- 接下来是什么 229
- 第10章 基于特殊进程打造自己的 behavior 230
- 特殊进程 230
- 互斥体 231
- 启动特殊进程 232
- 互斥体的状态 235
- 处理退出 236
- 系统消息 237
- 跟踪与日志事件 238
- 合在一起 239
- 动态模块和休眠 243
- 属于你自己的 behavior 244
- 创建 behavior 时的要求 245
- 一个处理 TCP流的例子 245
- 总结 249
- 接下来是什么 250
- 第 11章 系统原则与发行包制作. 251
- 系统原则 252
- 发行包目录结构 253
- 发行包资源文件 257
- 创建发行包 260
- 创建 boot 文件 262
- 打包发行包 271
- 启动脚本以及目标上的配置 275
- 参数和标志 277
- init模块 289
- rebar3 290
- 生成一个 rebar3 发行包项目 292
- 使用 rebar3 创建发行包 295
- 使用 rebar3 处理制作发行包时的项目依赖问题 298
- 总结 300
- 接下来是什么 304
- 第 12章 发行包升级 305
- 软件升级 305
- 第一个版本的咖啡机 FSM 308
- 添加一个新状态 311
- 为发行包创建升级 314
- 负责升级的代码 318
- 应用程序升级文件 322
- 高级指令 325
- 发行包升级文件 328
- 低级指令 330
- 安装升级 332
- 发行包处理器 334
- 升级环境变量 338
- 升级特殊进程 338
- 在分布式环境下升级 339
- 升级模拟器和核心 application 340
- 使用 Rebar3进行升级 341
- 总结 344
- 接下来是什么 346
- 第 13章 分布式架构 347
- 节点类型与家族 348

联网 351
分布式 Erlang 353
套接字与 SSL 359
面向服务和微服务的架构 361
点对点 362
接口 364
总结 366
接下来是什么 367
第.14.章 永不停止的系统 368
可用性 368
容错 369
弹性 370
可靠性 371
数据共享 375
一致性和可用性之间的权衡 383
总结 384
接下来是什么 385
第.15.章 水平规模伸缩 386
水平规模伸缩与垂直规模伸缩 386
容量规划 390
容量测试 392
平衡你的系统 394
找寻瓶颈 396
系统蓝图 398
负载调节与背压 399
总结 401
接下来是什么 403
第 16章 监视与抢救性支持 404
监视 405
日志 406
指标 411
警报 414
抢救性支持 416
总结 418
接下来是什么 420
索引 421
• • • • • ([收起](#))

[高伸缩性系统：Erlang/OTP大型分布式容错设计_下载链接1](#)

标签

Erlang

分布式

OTP

函数式编程

Elixir

计算科学

计算机

IT

评论

是也乎、(ノ▽ノ)ノ 87分钟先舔一舔 --> 春雨习惯性在每一本图书上挂名 <-- 好习惯~
真的是到处都是老梗的好书…用 UML 统一所有章节图示 --> 也暗示了 OTP
在所有侧向都统一了风格 --> 所有方面的 --> 设计~代码~运行时~架构…………怪不得用上
OTP 就离不了了 --> 被三十年前的暖男们关照的感觉太舒服了啊啊啊啊啊啊啊啊

OTP很强大,就是应用场景少.

翻译略有瑕疵。书很棒!

[高伸缩性系统：Erlang/OTP大型分布式容错设计_下载链接1](#)

书评

[高伸缩性系统：Erlang/OTP大型分布式容错设计_下载链接1](#)