

Broadview[®]
www.broadview.com.cn

Netty进阶之路

跟着案例学Netty

李林锋 著



中国工信出版集团

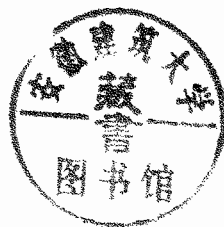


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Netty进阶之路

跟着案例学Netty

李林锋 著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Netty 将 Java NIO 接口封装, 提供了全异步编程方式, 是各大 Java 项目的网络应用开发必备神器。本书作者是国内 Netty 技术的先行者和布道者, 本书是他继《Netty 权威指南》之后的又一力作。

在本书中, 作者将在过去几年实践中遇到的问题, 以及 Netty 学习者咨询的相关问题, 进行了归纳和总结, 以问题案例做牵引, 通过对案例进行剖析, 讲解问题背后的原理, 并结合 Netty 源码分析, 让读者能够真正掌握 Netty, 在实际工作中少犯错。

本书中的案例涵盖了 Netty 的启动和停止、内存、并发多线程、性能、可靠性、安全等方面, 囊括了 Netty 绝大多数常用的功能及容易让人犯错的地方。在案例的分析过程中, 还穿插讲解了 Netty 的问题定位思路、方法、技巧, 以及解决问题使用的相关工具, 对读者在实际工作中用好 Netty 具有很大的帮助和启发作用。

本书适合架构师、设计师、开发工程师、测试工程师, 以及对 Java NIO 框架、Netty 感兴趣的其他相关人士阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

Netty 进阶之路: 跟着案例学 Netty / 李林锋著. —北京: 电子工业出版社, 2019.1
ISBN 978-7-121-35262-1

I. ①N… II. ①李… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 240985 号

责任编辑: 董 英

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16 印张: 21.25 字数: 425 千字

版 次: 2019 年 1 月第 1 版

印 次: 2019 年 1 月第 2 次印刷

定 价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。

序 1

随着近几年的快速发展，移动互联网系统的复杂度快速上升。为了满足业务快速迭代的需求，同时提高系统的可靠性和可维护性，越来越多的大型系统后台开始采用微服务架构。以华为应用市场为例，目前后台微服务数量达上百个，服务器多达数千台。服务器数量的快速增加，必然导致后台系统复杂度的快速增加，从而推动后台技术架构的持续演进。

在单体系统中，一个请求通常只涉及一个部件。但微服务化后，一个请求可能涉及数个微服务的配合。传统的后台服务通常采用同步阻塞调用方式，一个请求独占一个线程，只有在请求处理完成后，线程才能被释放。如果一个请求涉及多个微服务之间的调用，那么每个微服务都需要一个专门的连接和线程来处理这个请求。而系统的线程数和连接数都是有限的。当系统调用关系越来越复杂，一个很小的问题（如网络抖动、突发请求或 GC 引起的延时增加），都可能导致线程阻塞，引发雪崩，导致整个系统不可用。另一方面，同步阻塞的方式很容易导致系统的资源利用率降低。因此，业界领先的互联网公司，都已经或者正在切换到全栈异步方式。

相比同步 I/O 简单直接的编程模型，异步 I/O 在编程模型上有较大的差异，对开发人员也有更高的要求，同时问题的定位也更为复杂。Netty 是当前业界应用最广泛的 Java 开源异步框架。Netty 框架能显著降低异步开发的门槛，使开发人员聚焦业务逻辑，免于处理复杂的底层通信机制和线程模型，从而能够简单和快速地开发异步应用。时至今日，越来越多的国内公司开始使用 Netty 来构建应用，使用 Netty 的开发者也日益增加。

随着 Netty 应用的不断深入，大家在使用过程中也会遇到各种各样的问题。相比 Netty 的火热，市场上 Netty 相关的书籍却很少。作者几年前出版的著作《Netty 权威指南》是国

内第一本系统化讲解 Netty 原理和架构的书籍，在市场上取得了良好的反响。《Netty 进阶之路：跟着案例学 Netty》是作者在 Netty 方面的又一力作。该书从一个个典型的问题出发，让读者能够带着问题来展开学习，并通过代码解读、原理分析和问题总结，对每个问题抽丝剥茧地深入解析；同时，能够通过一个问题，将相关领域的知识理解透彻，达到举一反三的效果，进而实现对 Netty 的系统性学习与理解。

李林锋在电信软件行业有着近十年的异步和服务化方面的开发和架构设计经验，是华为公司该领域的专家。他从 2017 年开始负责华为应用市场在异步和微服务化方面的工作，主导了华为应用市场后台的异步化和微服务化演进，将电信软件严谨、稳定、高性能的优势，与移动互联网海量用户、高并发场景结合起来，显著提升了华为应用市场的可用性和性能。本书凝结了作者多年来在异步化工作方面的经验，将成为希望精通 Netty 开发的读者的重要参考书。

华为应用市场总架构师 刘连喜

序 2

随着互联网对各行各业的渗透，“连接”便成了我们看到的所有美好应用背后的基石。人、系统、物三者之间无所不在的连接让我们甚至感觉这世界已经成为了一个整体。细数一些计算机领域的热门技术，例如云计算、微服务、物联网等，其背后的核心还是连接。在这样一个背景下，掌握 Netty 可以算得上是一个开发人员最重要的技能。

就像码农的学名其实是计算机工程师一样，计算机其实是一个工程学科，是一个天生看重实践的学科。在现在的环境里，各种技术文档在网上一抓一大把，但是仍然有一些东西很难得到，那就是你需要填的那些坑。

李林锋长期工作在技术一线，构建的是压力最大的系统，保证的是要求最高的服务，解决的是诡异难解的问题。而难能可贵的是，他把他自己填过的那些坑都详尽地记录了下来，不仅提供思路，还分析原理，让所有问题都无所遁形。

《Netty 进阶之路：跟着案例学 Netty》就是能够助人出坑的干货。

佛教的修行次第是“信，解，行，证”。李林锋一直以来对技术的热情、“知行合一”的行事方式和对问题刨根问底的工作态度非常难得，相信通过这本书也可以让你由“解”入“行”，他趟过那些大坑的经验也一定会助你早日在 Netty 学习的路上实现“自证”。

华为云高级架构师 张琦

前言

自 2014 年《Netty 权威指南》出版后，我在技术网站上相继写了一些 Netty 专题文章，涵盖性能、线程模型、安全性等知识点，受到很多读者的喜爱。在 4 年多里，很多读者及 Netty 学习者向我咨询 Netty 相关的问题，这些问题加起来多达上千个，通过对问题做汇总和分析，可以归纳为如下几类：

（1）Netty 初学者，想了解学习 Netty 需要储备哪些技能，掌握哪些知识点，有什么学习技巧可以更快地掌握 Netty。

（2）《Netty 权威指南》的读者，学习完理论知识后，想在实际项目中使用，但是真正跟具体项目结合在一起解决实际问题时，又感觉比较棘手，不知道自己使用的方式是否是最优的，希望能够多学一些案例实践方面的知识，以便更好地在业务中使用 Netty。

（3）在实际项目中遇到了问题的工程师，由于对 Netty 底层细节掌握得不扎实，无法有效地定位并解决问题。

Netty 的一个特点就是入门相对容易，但是真正掌握并精通是非常困难的，原因有如下几个：

（1）涉及的知识面比较广。Netty 作为一个高性能的 NIO 通信框架，涉及的知识点包括网络通信、多线程编程、序列化和反序列化、异步和同步、SSL/TLS 安全、内存池、HTTP 等各种协议栈，这些知识点在 Java 语言中本身就是难点和重点，如果对这些基础知识掌握不扎实，是很难真正掌握好 Netty 的。

（2）调试比较困难。因为大量使用异步编程接口，以及消息处理过程中的各种线程切换，相比传统同步代码，Netty 代码调试难度比较大。

(3) 类继承层次比较深，有些代码很晦涩（例如内存池）。对于初学者而言，通过阅读代码来掌握 Netty 的难度还是很大的。

(4) 代码规模庞大。目前，Netty 的代码规模已经非常庞大，特别是协议栈部分，提供了对 HTTP/2、MQTT、WebSocket 等各种协议的支持，相关代码非常多。如果学习方式不当，抓不住重点，则全量阅读 Netty 源码，既耗时又很难吃透，很容易半途而废。

(5) 资料零散，缺乏与实践相关的案例。网上 Netty 的各种资料非常多，但是都以理论讲解为主，Netty 在各行业中的应用、问题定位技巧及案例实践方面的资料很少，缺乏系统性的实践总结，是 Netty 学习的一大痛点。

在过去的几年中，我利用业余时间尽量帮大家答疑解惑，但实际上一个人很难回答所有读者的问题，有些问题需要业务描述、故障场景、日志，甚至要看源码，而且需要反复多次沟通来弄清楚问题，对于个人而言，时间和精力都很难得到保证。另外，一些比较常见的问题，例如服务端接收不到客户端的消息，定位手段是可以固化下来的。很多读者也希望我能写一本 Netty 实践和案例方面的书，通过案例讲解让大家更好地在项目中使用的 Netty，解决遇到的实际问题。

于是我对手头大家咨询的问题做了归类分析，结合我们自己的业务和平台多年来在 Netty 实践中积累的经验，写作了本书。本书以问题案例做牵引，通过对案例进行剖析，讲解问题背后的原理，并结合 Netty 源码分析，让读者能够真正掌握 Netty，在实际工作中少犯错。在案例的分析过程中，还穿插讲解了 Netty 的问题定位思路、方法、技巧，以及解决问题使用的相关工具，“授人以鱼不如授人以渔”，只有掌握了这些才能在项目中更放心地使用 Netty。

本书的内容分类主要包括：

- (1) Netty 的启动和停止
- (2) Netty 的内存
- (3) Netty 的并发多线程
- (4) Netty 的性能
- (5) Netty 的可靠性
- (6) Netty 的安全

（7）Netty 的实践

（8）Netty 的学习

书中的案例涵盖了 Netty 绝大多数常用的功能，以及容易犯错的地方，具有通用性和普遍性。学习这些案例，对于在实际业务工作中用好 Netty 有很大的帮助和启发作用。另外，在讲解 Netty 框架本身的同时，也会穿插一些背景知识介绍，例如 Java 信号量和优雅停机机制、Java 的 NIO 类库、HTTP 协议栈等。知识都是相互关联的，很难在基础知识不扎实的情况下掌握更高阶的知识。

通过本书的学习，希望广大 Netty 初学者和爱好者能够更快、更好地进入高级阶段，在项目中用好 Netty，为业务创造更多的价值。

尽管我也有技术洁癖，希望诸事完美，但是由于 Netty 代码的庞杂和涉及的知识点太多，以及受限于我个人的经历和水平，很难在一本书里同时满足所有读者的诉求。本书如有遗漏或者错误，恳请大家及时批评和指正，如果大家有好的建议或者想法，也可以联系我。联系方式如下：

◎ 微信：Nettying

◎ 新浪微博：Nettying

能够完成本书要感谢很多人，首先感谢华为公司给我提供了足够大的舞台，感谢华为消费者云服务应用市场团队领导张凡、叶文武、刘连喜等，以及这些年与我在平台和业务团队一起战斗过的架构师、设计师、开发工程师、测试工程师和资料员等同事。

其次感谢我的家人，你们一直在背后默默地支持我。感谢参与本书编辑的英姐、美工及其他人员，你们的辛苦换来了本书的如期上市。

最后感谢所有《Netty 权威指南》和《分布式服务框架原理与实践》的读者，你们的支持和鼓励是我写作本书的动力源泉。

李林锋

2018 年国庆节于南京

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 您即可享受以下服务。

- ◎ **下载资源:** 本书提供示例代码及资源文件, 均可在 [下载资源](#) 处下载。
- ◎ **提交勘误:** 您对书中内容的修改意见可在 [提交勘误](#) 处提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- ◎ **与作者交流:** 在页面下方 [读者评论](#) 处留下您的疑问或观点, 与作者和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/35262>



目 录

第 1 章	Netty 服务端意外退出案例	1
1.1	Netty 服务端意外退出问题	1
1.1.1	Java Daemon 线程简介	2
1.1.2	Netty 服务端启动原理	4
1.1.3	如何防止 Netty 服务端意外退出	6
1.1.4	实际项目中的优化策略	8
1.2	Netty 优雅退出机制	9
1.2.1	Java 优雅退出机制	10
1.2.2	Java 优雅退出的注意点	12
1.2.3	Netty 优雅退出机制	14
1.2.4	Netty 优雅退出原理和源码分析	15
1.2.5	Netty 优雅退出的一些误区	20
1.3	总结	21
第 2 章	Netty 客户端连接池资源泄漏案例	22
2.1	Netty 连接池资源泄漏问题	22
2.1.1	连接池创建代码	23
2.1.2	内存溢出和线程膨胀	23
2.1.3	错用 NIO 编程模式	25
2.1.4	正确的连接池创建方式	26
2.1.5	并发安全和资源释放	28
2.2	Netty 客户端创建机制	29
2.2.1	Java NIO 客户端创建原理分析	29

2.2.2	Netty 客户端创建原理分析	32
2.2.3	Bootstrap 工具类源码分析	34
2.3	总结	36
第 3 章	Netty 内存池泄漏疑云案例	37
3.1	Netty 内存池泄漏问题	37
3.1.1	路由转发服务代码	38
3.1.2	响应消息内存释放玄机	39
3.1.3	采集堆内存快照分析	42
3.1.4	ByteBuf 申请和释放的理解误区	45
3.2	Netty 内存池工作机制	48
3.2.1	内存池的性能优势	48
3.2.2	内存池工作原理分析	51
3.2.3	内存池核心代码分析	54
3.3	总结	58
第 4 章	ByteBuf 故障排查案例	59
4.1	HTTP 协议栈 ByteBuf 使用问题	59
4.1.1	HTTP 响应 Body 获取异常	59
4.1.2	ByteBuf 非法引用问题	63
4.1.3	ByteBuf 使用注意事项	66
4.2	Netty ByteBuf 实现机制	67
4.2.1	Java 原生 ByteBuffer 的局限性	67
4.2.2	Netty ByteBuf 工作原理分析	67
4.2.3	ByteBuf 引用计数器工作原理和源码分析	70
4.3	总结	73
第 5 章	Netty 发送队列积压导致内存泄漏案例	74
5.1	Netty 发送队列积压案例	74
5.1.1	高并发故障场景	74
5.1.2	内存泄漏原因分析	76
5.1.3	如何防止发送队列积压	78
5.1.4	其他可能导致发送队列积压的因素	80

5.2	Netty 消息发送工作机制	82
5.2.1	WriteAndFlushTask 原理和源码分析	83
5.2.2	ChannelOutboundBuffer 原理和源码分析	86
5.2.3	消息发送源码分析	88
5.2.4	消息发送高低水位控制	94
5.3	总结	95
第 6 章	API 网关高并发压测性能波动案例	96
6.1	高并发压测性能波动问题	96
6.1.1	故障场景模拟	96
6.1.2	性能波动原因定位	98
6.1.3	主动内存泄漏定位法	101
6.1.4	网关类产品的优化建议	102
6.2	Netty 消息接入内存申请机制	102
6.2.1	消息接入的内存分配原理和源码分析	102
6.2.2	Netty ByteBuf 的动态扩容原理和源码分析	107
6.3	总结	108
第 7 章	Netty ChannelHandler 并发安全案例	109
7.1	Netty ChannelHandler 并发安全问题	109
7.1.1	串行执行的 ChannelHandler	110
7.1.2	跨链路共享的 ChannelHandler	114
7.1.3	ChannelHandler 的并发陷阱	116
7.2	Netty ChannelHandler 工作机制	118
7.2.1	职责链 ChannelPipeline 原理和源码分析	118
7.2.2	用户自定义 Event 原理和源码分析	122
7.3	总结	123
第 8 章	车联网服务端接收不到车载终端消息案例	124
8.1	车联网服务端接收不到车载终端消息问题	124
8.1.1	故障现象	125
8.1.2	故障期线程堆栈快照分析	126
8.1.3	NioEventLoop 线程防挂死策略	128

8.2	NioEventLoop 线程工作机制	129
8.2.1	I/O 读写操作原理和源码分析	130
8.2.2	异步任务执行原理和源码分析	133
8.2.3	定时任务执行原理和源码分析	135
8.2.4	Netty 多线程最佳实践	137
8.3	总结	137
第 9 章	Netty 3.X 版本升级案例	139
9.1	Netty 3.X 的版本升级背景	139
9.1.1	被迫升级场景	140
9.1.2	升级不当遭遇各种问题	140
9.2	版本升级后数据被篡改问题	141
9.2.1	数据篡改原因分析	142
9.2.2	问题总结	143
9.3	升级后上下文丢失问题	143
9.3.1	上下文丢失原因分析	144
9.3.2	依赖第三方线程模型的思考	144
9.4	升级后应用遭遇性能下降问题	145
9.4.1	性能下降原因分析	145
9.4.2	性能优化建议	146
9.5	Netty 线程模型变更分析	147
9.5.1	Netty 3.X 版本线程模型	147
9.5.2	Netty 4.X 版本线程模型	149
9.5.3	线程模型变化点源码分析	150
9.5.4	线程模型变化总结	152
9.6	总结	154
第 10 章	Netty 并发失效导致性能下降案例	155
10.1	业务 ChannelHandler 无法并发执行问题	155
10.1.1	服务端并发设计相关代码分析	155
10.1.2	无法并行执行的 EventExecutorGroup	159
10.1.3	并行执行优化策略和结果	161
10.2	Netty DefaultEventExecutor 工作机制	163

10.2.1	DefaultEventExecutor 原理和源码分析	164
10.2.2	业务线程池优化策略	165
10.2.3	Netty 线程绑定机制原理和源码分析	168
10.3	总结	170
第 11 章 IoT 百万长连接性能调优案例		171
11.1	海量长连接接入面临的挑战	171
11.1.1	IoT 设备接入特点	172
11.1.2	IoT 服务端性能优化场景	172
11.1.3	服务端面临的性能挑战	172
11.2	智能家居内存泄漏问题	173
11.2.1	服务端内存泄漏原因定位	173
11.2.2	问题背后的一些思考	174
11.3	操作系统参数调优	174
11.3.1	文件描述符	175
11.3.2	TCP/IP 相关参数	175
11.3.3	多网卡队列和软中断	177
11.4	Netty 性能调优	177
11.4.1	设置合理的线程数	177
11.4.2	心跳优化	180
11.4.3	接收和发送缓冲区调优	183
11.4.4	合理使用内存池	184
11.4.5	防止 I/O 线程被意外阻塞	185
11.4.6	I/O 线程和业务线程分离	187
11.4.7	针对端侧并发连接数的流控	187
11.5	JVM 相关性能优化	189
11.5.1	GC 调优	189
11.5.2	其他优化手段	193
11.6	总结	193
第 12 章 静态检查修改不当引起性能下降案例		195
12.1	Edge Service 性能严重下降问题	195
12.1.1	Edge Service 热点代码分析	195

12.1.2	静态检查问题不是简单的一改了之	197
12.1.3	问题反思和改进	200
12.2	克隆和浅拷贝	201
12.2.1	浅拷贝存在的问题	201
12.2.2	Netty 的对象拷贝实现策略	203
12.3	总结	204
第 13 章	Netty 性能统计误区案例	205
13.1	时延毛刺排查相关问题	205
13.1.1	时延毛刺问题初步分析	205
13.1.2	服务调用链改进	207
13.1.3	都是同步思维惹的祸	208
13.1.4	正确的消息发送速度性能统计策略	209
13.1.5	常见的消息发送性能统计误区	212
13.2	Netty 关键性能指标采集策略	212
13.2.1	Netty I/O 线程池性能指标	213
13.2.2	Netty 发送队列积压消息数	214
13.2.3	Netty 消息读取速度性能统计	215
13.3	总结	215
第 14 章	gRPC 的 Netty HTTP/2 实践案例	216
14.1	gRPC 基础入门	216
14.1.1	RPC 框架简介	216
14.1.2	当前主流的 RPC 框架	218
14.1.3	gRPC 框架特点	218
14.1.4	为什么选择 HTTP/2	219
14.2	gRPC Netty HTTP/2 服务端工作机制	220
14.2.1	Netty HTTP/2 服务端创建原理和源码分析	220
14.2.2	服务端接收 HTTP/2 请求消息原理和源码分析	224
14.2.3	服务端发送 HTTP/2 响应消息原理和源码分析	231
14.3	gRPC Netty HTTP/2 客户端工作机制	234
14.3.1	Netty HTTP/2 客户端创建原理和源码分析	235
14.3.2	客户端发送 HTTP/2 请求消息原理和源码分析	238

14.3.3	客户端接收 HTTP/2 响应消息原理和源码分析	242
14.4	gRPC 消息序列化机制	243
14.4.1	Google Protobuf 简介	243
14.4.2	消息的序列化原理和源码分析	244
14.4.3	消息的反序列化原理和源码分析	245
14.5	gRPC 线程模型	246
14.5.1	服务端线程模型	246
14.5.2	客户端线程模型	247
14.5.3	线程模型总结	248
14.6	总结	249
第 15 章	Netty 事件触发策略使用不当案例	250
15.1	channelReadComplete 方法被调用多次问题	250
15.1.1	ChannelHandler 调用问题	250
15.1.2	生产环境问题模拟重现	252
15.2	ChannelHandler 使用的一些误区总结	255
15.2.1	channelReadComplete 方法调用	255
15.2.2	ChannelHandler 职责链调用	257
15.3	总结	258
第 16 章	Netty 流量整形应用案例	259
16.1	Netty 流量整形功能	259
16.1.1	通用的流量整形功能简介	260
16.1.2	Netty 流量整形功能简介	260
16.2	Netty 流量整形应用	261
16.2.1	流量整形示例代码	261
16.2.2	流量整形功能测试	263
16.3	Netty 流量整形工作机制	264
16.3.1	流量整形工作原理和源码分析	264
16.3.2	并发编程在流量整形中的应用	271
16.3.3	使用流量整形的一些注意事项总结	274
16.4	总结	278