

Netty



权威指南

李林锋 / 著

Java高性能NIO通信首选框架

大数据时代构建高可用分布式系统利器

Netty: The Definitive Guide

Netty

权威指南

李林锋 / 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

《Netty 权威指南》是异步非阻塞通信领域的经典之作，基于最新版本 Netty 5.0 编写，是国内首本深入介绍 Netty 原理和架构的技术书籍，也是作者多年实战经验的总结和浓缩。内容包含基础功能、高级应用、系统架构、源码分析和行业应用，深入阐述了 Java I/O 的 Netty NIO 开发、Netty 编解码开发、Netty 多协议开发等各方面的技术要点，包含了对源码的深刻解读，并且对 Netty 的应用现状和未来趋势进行分析，旨在帮助从业人员提升自我，更快更明确地发展职业道路。

本书适合架构师、设计师、软件开发工程师、测试人员和其他对 Java NIO 框架、Java 通信感兴趣的的相关人士阅读，相信通过学习本书，能够熟悉和掌握 Netty 这一优秀的异步通信框架，实现高可用分布式系统的构建。

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 权威指南 / 李林锋著. — 北京：电子工业出版社，2014.6
ISBN 978-7-121-23343-2



I. ①N… II. ①李… III. ①JAVA 语言—程序设计—指南 IV. ①TP312-62

中国版本图书馆 CIP 数据核字（2014）第 114646 号

策划编辑：孙学瑛

责任编辑：徐津平

印 刷：北京中新伟业印刷有限公司

装 订：河北省三河市路通装订厂

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：32.75 字数：625 千字

版 次：2014 年 6 月第 1 版

印 次：2014 年 6 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。

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。

服务热线：（010）88258888。

前 言

大约在 2008 年的时候，我参与设计和开发的一个电信系统在月初出帐期，总是发生大量的连接超时和读写超时异常，业务的失败率相比于平时高了很多，报表中的很多指标都差强人意。后来经过排查，发现问题的主要原因出现在下游网元的处理性能上，月初的时候 BSS 出帐，在出帐期间 BSS 系统运行缓慢，由于双方采用了同步阻塞式的 HTTP+XML 进行通信，导致任何一方处理缓慢都会影响对方的处理性能。按照故障隔离的设计原则，对方处理速度慢或者不回应答，不应该影响系统的其他功能模块或者协议栈，但是在同步阻塞 I/O 通信模型下，这种故障传播和相互影响是不可避免的，很难通过业务层面解决。

受限于当时 Tomcat 和 Servlet 的同步阻塞 I/O 模型，以及在 Java 领域异步 HTTP 协议栈的技术积累不足，当时我们并没有办法完全解决这个问题，只能通过调整线程池策略和 HTTP 超时时间来从业务层面做规避。

2009 年，由于对技术的热爱，我作为业务骨干被领导派去参加一个重点业务平台的研发工作，与两位资深的架构师（其中一位工作 20 年，做华为交换机出身）共同参与。这是我第一次全面接触异步 I/O 编程和高性能电信级协议栈的开发，眼界大开——异步高性能内部协议栈、异步 HTTP、异步 SOAP、异步 SMPP……所有的协议栈都是异步非阻塞。后来的性能测试表明：基于 Reactor 模型统一调度的长连接和短连接协议栈，无论是性能、可靠性还是可维护性，都可以“秒杀”传统基于 BIO 开发的应用服务器和各种协议栈，这种差异本质上是一种代差。

在我从事异步 NIO 编程的 2009 年，业界还没有成熟的 NIO 框架，那个时候 Mina 刚开始起步，功能和性能都达不到商用标准。最困难的是，国内 Java 领域的异步通信还没有流行，整个业界的积累都非常少。那个时候资料匮乏，能够交流和探讨的圈内人很少，一旦踩住“地雷”，就需要夜以继日地维护。在随后 2 年多的时间里，经历了 10 多次的在通宵、凌晨被一线的运维人员电话吵醒等种种磨难之后，我们自研的 NIO 框架才逐渐稳定和成熟。期间，解决的 BUG 总计 20~30 个。

从 2004 年 JDK1.4 首次提供 NIO 1.0 类库到现在，已经过去了整整 10 年。JSR 51 的设计初衷就是让 Java 能够提供非阻塞、具有弹性伸缩能力的异步 I/O 类库，从而结束 Java 在高性能服务器领域的不利地位。然而，在相当长的一段时间里，Java 的 NIO 编程并没有流行起来，究其原因如下。

1. 大多数高性能服务器，被 C 和 C++ 语言盘踞，由于它们可以直接使用操作系统的异步 I/O 能力，所以对 JDK 的 NIO 并不关心；

2. 移动互联网尚未兴起，基于 Java 的大规模分布式系统极少，很多中小型应用服务对于异步 I/O 的诉求不是很强烈；

3. 高性能、高可靠性领域，例如银行、证券、电信等依然以 C++ 为主导，Java 充当打杂的角色，NIO 暂时没有用武之地；

4. 当时主流的 J2EE 服务器，几乎全部基于同步阻塞 I/O 构建，例如 Servlet、Tomcat 等，由于它们应用广泛，如果这些容器不支持 NIO，用户很难具备独立构建异步协议栈的能力；

5. 异步 NIO 编程门槛比较高，开发和维护一款基于 NIO 的协议栈对很多中小型公司来说像是一场噩梦；

6. 业界 NIO 框架不成熟，很难商用；

7. 国内研发界对 NIO 的陌生和认识不足，没有充分重视。

基于上述几种原因，NIO 编程的推广和发展长期滞后。值得欣慰的是，随着大规模分布式系统、大数据和流式计算框架的兴起，基于 Java 来构建这些系统已经成为主流，NIO 编程和 NIO 框架在此期间得到了大规模的商用。在互联网领域，阿里的分布式服务框架 Dubbo、RocketMQ，大数据的基础序列化和通信框架 Avro，以及很多开源的软件都已经开始使用 Netty 来构建高性能、分布式通信能力，Netty 社区的活跃度也名列前茅。根据目前的信息，Netty 已经在如下几个领域得到了大规模的商业应用。

1. 互联网领域；

2. 电信领域；

3. 大数据领域；

4. 银行、证券等金融领域；

5. 游戏行业；
6. 电力等企业市场。

2014 年春节前，我分享了一篇博文《Netty5.0 架构剖析和源码解读》，短短 1 个月下载量达到了 4000 多。很多网友向我咨询 NIO 编程技术、NIO 框架如何选择等问题，也有一些圈内朋友和出版社邀请我写一本关于 Netty 的技术书籍。作为最流行、表现最优异的 NIO 框架，Netty 深受大家喜爱，但是长期以来除了 UserGuide 之外，国内鲜有 Netty 相关的技术书籍供广大 NIO 编程爱好者学习和参考。由于 Netty 源码的复杂性和 NIO 编程本身的技术门槛限制，对于大多数读者而言，通过自己阅读和分析源码来深入掌握 Netty 的设计原理和实现细节是件困难的事情。从 2011 年开始我系统性的分析和应用了 Netty 和 Mina，转瞬间已经过去了 3 年多。在这 3 年的时间里，我们的系统经受了无数严苛的考验，在这个过程中，对 Netty 和 Mina 有了更深刻的体验，也积累了丰富的运维和实战经验。我们都是开源框架 Netty 的受益者，为了让更多的朋友和同行能够了解 NIO 编程，深入学习和掌握 Netty 这个 NIO 利器，我打算将我的经验和大家分享，同时也结束国内尚无 Netty 学习教材的尴尬境地。

联系方式

尽管我也有技术洁癖，希望诸事完美，但是由于 Netty 代码的庞杂和涉及到的知识点太多，一本书籍很难涵盖所有的功能点。如有遗漏或者错误，恳请大家能够及时批评和指正，如果你有好的建议或者想法，也可以联系我。我的联系方式如下。

邮箱：neu_lilinfeng@sina.com。

新浪微博：Nettying。

微信：Nettying。

读者讨论群：195820454。

致谢

如果说个人能够改变自己命运的话，对于程序员来说，唯有通过不断的学习和实践，

努力提升自己的技能，才有可能找到更好的机会，充分发挥和体现自己的价值。我希望本书能够为你的成功助一臂之力。

感谢博文视点的策划编辑丁一琼和幕后的美编，正是你们的辛苦工作才保证了本书能够顺利出版；感谢华为 Netty 爱好者和关注本书的领导同事们的支持，你们的理解和鼓励为我提供了足够的勇气。感谢我的家人和老婆的支持，写书占用了我几乎所有的业余时间，没有你们的理解和支持，我很难安心写作。

最后感谢 Netty 中国社区的朋友，我的微博粉丝和所有喜欢 Netty 的朋友们，你们对技术的热情是鼓励我写书的最重要动力，没有你们，就没有本书。希望大家一如既往的喜欢 NIO 编程，喜欢 Netty，以及相互交流和分享，共同推动整个国内异步高性能通信领域的技术发展。

李林锋

5 月 11 日于南京紫轩阁

十載耕耘 奠定专业地位

博文视点诚邀精锐作者加盟

以书为证 彰显卓越品质

《C++Primer (中文版) (第5版)》、《淘宝技术这十年》、《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

十年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巔。

英雄帖

江湖风云起,代有才人出。

IT界群雄并起,逐鹿中原。

博文视点诚邀天下技术英豪加入,

指点江山,激扬文字

传播信息技术,分享IT心得

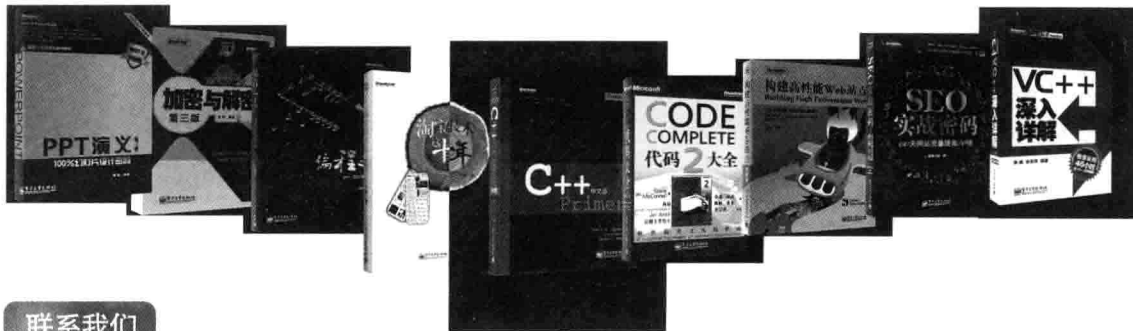
• 专业的作者服务 •

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

善待作者——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯,并会参照作者实际的工作、生活节奏,量身定制写作计划,确保合作顺利进行。

提升作者——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



联系我们

博文视点官网: <http://www.broadview.com.cn>

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

投稿电话: 010-51260888 88254368

投稿邮箱: jsj@phei.com.cn



@博文视点Broadview



博文视点Broadview



目 录

基础篇 走进 Java NIO

第 1 章	Java 的 I/O 演进之路	2
1.1	I/O 基础入门	3
1.1.1	Linux 网络 I/O 模型简介	3
1.1.2	I/O 多路复用技术	6
1.2	Java 的 I/O 演进	8
1.3	总结	10
第 2 章	NIO 入门	11
2.1	传统的 BIO 编程	11
2.1.1	BIO 通信模型图	12
2.1.2	同步阻塞式 I/O 创建的 TimeServer 源码分析	13
2.1.3	同步阻塞式 I/O 创建的 TimeClient 源码分析	16
2.2	伪异步 I/O 编程	18
2.2.1	伪异步 I/O 模型图	19
2.2.2	伪异步式 I/O 创建的 TimeServer 源码分析	19
2.2.3	伪异步 I/O 弊端分析	21
2.3	NIO 编程	24
2.3.1	NIO 类库简介	24
2.3.2	NIO 服务端序列图	28
2.3.3	NIO 创建的 TimeServer 源码分析	31
2.3.4	NIO 客户端序列图	36
2.3.5	NIO 创建的 TimeClient 源码分析	39
2.4	AIO 编程	45
2.4.1	AIO 创建的 TimeServer 源码分析	46

2.4.2	AIO 创建的 TimeClient 源码分析	52
2.4.3	AIO 版本时间服务器运行结果	56
2.5	4 种 I/O 的对比	58
2.5.1	概念澄清	58
2.5.2	不同 I/O 模型对比	60
2.6	选择 Netty 的理由	60
2.6.1	不选择 Java 原生 NIO 编程的原因	61
2.6.2	为什么选择 Netty	62
2.7	总结	63

入门篇 Netty NIO 开发指南

第 3 章	Netty 入门应用	66
3.1	Netty 开发环境的搭建	66
3.1.1	下载 Netty 的软件包	67
3.1.2	搭建 Netty 应用工程	67
3.2	Netty 服务端开发	68
3.3	Netty 客户端开发	73
3.4	运行和调试	76
3.4.1	服务端和客户端的运行	76
3.4.2	打包和部署	77
3.5	总结	78
第 4 章	TCP 粘包/拆包问题的解决之道	79
4.1	TCP 粘包/拆包	79
4.1.1	TCP 粘包/拆包问题说明	80
4.1.2	TCP 粘包/拆包发生的原因	80
4.1.3	粘包问题的解决策略	81
4.2	未考虑 TCP 粘包导致功能异常案例	82
4.2.1	TimeServer 的改造	82
4.2.2	TimeClient 的改造	83
4.2.3	运行结果	84
4.3	利用 LineBasedFrameDecoder 解决 TCP 粘包问题	88
4.3.1	支持 TCP 粘包的 TimeServer	88

4.3.2	支持 TCP 粘包的 TimeClient	90
4.3.3	运行支持 TCP 粘包的时间服务器程序	93
4.3.4	LineBasedFrameDecoder 和 StringDecoder 的原理分析	98
4.4	总结	99
第 5 章	分隔符和定长解码器的应用	100
5.1	DelimiterBasedFrameDecoder 应用开发	101
5.1.1	DelimiterBasedFrameDecoder 服务端开发	101
5.1.2	DelimiterBasedFrameDecoder 客户端开发	104
5.1.3	运行 DelimiterBasedFrameDecoder 服务端和客户端	106
5.2	FixedLengthFrameDecoder 应用开发	108
5.2.1	FixedLengthFrameDecoder 服务端开发	108
5.2.2	利用 telnet 命令行测试 EchoServer 服务端	110
5.3	总结	111

中级篇 Netty 编解码开发指南

第 6 章	编解码技术	114
6.1	Java 序列化的缺点	115
6.1.1	无法跨语言	115
6.1.2	序列化后的码流太大	115
6.1.3	序列化性能太低	118
6.2	业界主流的编解码框架	121
6.2.1	Google 的 Protobuf 介绍	121
6.2.2	Facebook 的 Thrift 介绍	123
6.2.3	JBoss Marshalling 介绍	124
6.3	总结	125
第 7 章	Java 序列化	126
7.1	Netty Java 序列化服务端开发	126
7.2	Java 序列化 Netty 客户端开发	132
7.3	运行结果	135
7.4	总结	137

第 8 章	Google Protobuf 编解码	138
8.1	Protobuf 的入门	139
8.1.1	Protobuf 开发环境搭建	139
8.1.2	Protobuf 编解码开发	141
8.1.3	运行 Protobuf 例程	143
8.2	Netty 的 Protobuf 服务端开发	143
8.2.1	Protobuf 版本的图书订购服务端开发	144
8.2.2	Protobuf 版本的图书订购客户端开发	146
8.2.3	Protobuf 版本的图书订购程序功能测试	149
8.3	Protobuf 的使用注意事项	152
8.4	总结	153
第 9 章	JBoss Marshalling 编解码	154
9.1	Marshalling 开发环境准备	154
9.2	Netty 的 Marshalling 服务端开发	155
9.3	Netty 的 Marshalling 客户端开发	158
9.4	运行 Marshalling 客户端和服务端例程	160
9.5	总结	162

高级篇 Netty 多协议开发和应用

第 10 章	HTTP 协议开发应用	164
10.1	HTTP 协议介绍	165
10.1.1	HTTP 协议的 URL	165
10.1.2	HTTP 请求消息 (HttpRequest)	165
10.1.3	HTTP 响应消息 (HttpResponse)	168
10.2	Netty HTTP 服务端入门开发	169
10.2.1	HTTP 服务端例程场景描述	170
10.2.2	HTTP 服务端开发	170
10.2.3	Netty HTTP 文件服务器例程运行结果	178
10.3	Netty HTTP+XML 协议栈开发	182
10.3.1	开发场景介绍	183
10.3.2	HTTP+XML 协议栈设计	186
10.3.3	高效的 XML 绑定框架 JiBx	187

10.3.4	HTTP+XML 编解码框架开发	195
10.3.5	HTTP+XML 协议栈测试	211
10.3.6	小结	213
10.4	总结	214
第 11 章	WebSocket 协议开发	215
11.1	HTTP 协议的弊端	216
11.2	WebSocket 入门	216
11.2.1	WebSocket 背景	217
11.2.2	WebSocket 连接建立	218
11.2.3	WebSocket 生命周期	219
11.2.4	WebSocket 连接关闭	220
11.3	Netty WebSocket 协议开发	221
11.3.1	WebSocket 服务端功能介绍	221
11.3.2	WebSocket 服务端开发	222
11.3.3	运行 WebSocket 服务端	230
11.4	总结	231
第 12 章	UDP 协议开发	233
12.1	UDP 协议简介	234
12.2	UDP 服务端开发	235
12.3	UDP 客户端开发	238
12.4	运行 UDP 例程	240
12.5	总结	241
第 13 章	文件传输	242
13.1	文件的基础知识	243
13.1.1	文件的概念	243
13.1.2	文件路径	243
13.1.3	文件名称	244
13.1.4	FileChannel 简介	244
13.2	Netty 文件传输开发	245
13.3	运行 Netty 文件传输服务例程	248
13.4	总结	250

第 14 章	私有协议栈开发	251
14.1	私有协议介绍	251
14.2	Netty 协议栈功能设计	253
14.2.1	网络拓扑图	253
14.2.2	协议栈功能描述	254
14.2.3	通信模型	254
14.2.4	消息定义	255
14.2.5	Netty 协议支持的字段类型	256
14.2.6	Netty 协议的编解码规范	257
14.2.7	链路的建立	259
14.2.8	链路的关闭	260
14.2.9	可靠性设计	260
14.2.10	安全性设计	262
14.2.11	可扩展性设计	262
14.3	Netty 协议栈开发	263
14.3.1	数据结构定义	263
14.3.2	消息编解码	267
14.3.3	握手和安全认证	271
14.3.4	心跳检测机制	275
14.3.5	断连重连	278
14.3.6	客户端代码	279
14.3.7	服务端代码	281
14.4	运行协议栈	282
14.4.1	正常场景	282
14.4.2	异常场景：服务端宕机重启	283
14.4.3	异常场景：客户端宕机重启	286
14.5	总结	286

源码分析篇 Netty 功能介绍和源码分析

第 15 章	ByteBuf 和相关辅助类	290
15.1	ByteBuf 功能说明	290
15.1.1	ByteBuf 的工作原理	291
15.1.2	ByteBuf 的功能介绍	296

15.2	ByteBuf 源码分析	310
15.2.1	ByteBuf 的主要类继承关系	311
15.2.2	AbstractByteBuf 源码分析	312
15.2.3	AbstractReferenceCountedByteBuf 源码分析	321
15.2.4	UnpooledHeapByteBuf 源码分析	323
15.2.5	PooledByteBuf 内存池原理分析	328
15.2.6	PooledDirectByteBuf 源码分析	331
15.3	ByteBuf 相关的辅助类功能介绍	334
15.3.1	ByteBufHolder	334
15.3.2	ByteBufAllocator	335
15.3.3	CompositeByteBuf	336
15.3.4	ByteBufUtil	338
15.4	总结	339
第 16 章	Channel 和 Unsafe	340
16.1	Channel 功能说明	340
16.1.1	Channel 的工作原理	341
16.1.2	Channel 的功能介绍	342
16.2	Channel 源码分析	345
16.2.1	Channel 的主要继承关系类图	345
16.2.2	AbstractChannel 源码分析	346
16.2.3	AbstractNioChannel 源码分析	349
16.2.4	AbstractNioByteChannel 源码分析	352
16.2.5	AbstractNioMessageChannel 源码分析	356
16.2.6	AbstractNioMessageServerChannel 源码分析	357
16.2.7	NioServerSocketChannel 源码分析	358
16.2.8	NioSocketChannel 源码分析	360
16.3	Unsafe 功能说明	366
16.4	Unsafe 源码分析	367
16.4.1	Unsafe 继承关系类图	367
16.4.2	AbstractUnsafe 源码分析	367
16.4.3	AbstractNioUnsafe 源码分析	377
16.4.4	NioByteUnsafe 源码分析	380

16.5	总结	387
第 17 章	ChannelPipeline 和 ChannelHandler	389
17.1	ChannelPipeline 功能说明	390
17.1.1	ChannelPipeline 的事件处理	390
17.1.2	自定义拦截器	392
17.1.3	构建 pipeline	393
17.1.4	ChannelPipeline 的主要特性	394
17.2	ChannelPipeline 源码分析	394
17.2.1	ChannelPipeline 的类继承关系图	394
17.2.2	ChannelPipeline 对 ChannelHandler 的管理	394
17.2.3	ChannelPipeline 的 inbound 事件	397
17.2.4	ChannelPipeline 的 outbound 事件	398
17.3	ChannelHandler 功能说明	399
17.3.1	ChannelHandlerAdapter 功能说明	400
17.3.2	ByteToMessageDecoder 功能说明	400
17.3.3	MessageToMessageDecoder 功能说明	401
17.3.4	LengthFieldBasedFrameDecoder 功能说明	402
17.3.5	MessageToByteEncoder 功能说明	405
17.3.6	MessageToMessageEncoder 功能说明	406
17.3.7	LengthFieldPrepender 功能说明	406
17.4	ChannelHandler 源码分析	407
17.4.1	ChannelHandler 的类继承关系图	407
17.4.2	ByteToMessageDecoder 源码分析	408
17.4.3	MessageToMessageDecoder 源码分析	411
17.4.4	LengthFieldBasedFrameDecoder 源码分析	413
17.4.5	MessageToByteEncoder 源码分析	417
17.4.6	MessageToMessageEncoder 源码分析	418
17.4.7	LengthFieldPrepender 源码分析	419
17.5	总结	420
第 18 章	EventLoop 和 EventLoopGroup	421
18.1	Netty 的线程模型	421
18.1.1	Reactor 单线程模型	422

18.1.2	Reactor 多线程模型	423
18.1.3	主从 Reactor 多线程模型	424
18.1.4	Netty 的线程模型	425
18.1.5	最佳实践	427
18.2	NioEventLoop 源码分析	427
18.2.1	NioEventLoop 设计原理	427
18.2.2	NioEventLoop 继承关系类图	428
18.2.3	NioEventLoop	429
18.3	总结	440
第 19 章	Future 和 Promise	441
19.1	Future 功能	441
19.2	ChannelFuture 源码分析	446
19.3	Promise 功能介绍	448
19.4	Promise 源码分析	450
19.4.1	Promise 继承关系图	450
19.4.2	DefaultPromise	450
19.5	总结	453

架构和行业应用篇 Netty 高级特性

第 20 章	Java 多线程编程在 Netty 中的应用	456
20.1	Java 内存模型与多线程编程	456
20.1.1	硬件的发展和多任务处理	456
20.1.2	Java 内存模型	457
20.2	Netty 的并发编程实践	459
20.2.1	对共享的可变数据进行正确的同步	459
20.2.2	正确的使用锁	460
20.2.3	volatile 的正确使用	462
20.2.4	CAS 指令和原子类	465
20.2.5	线程安全类的应用	467
20.2.6	读写锁的应用	470
20.2.7	线程安全性文档说明	472
20.2.8	不要依赖线程优先级	473