

第一章：初识 Netty：背景、现状与趋势



扫码试看/订阅

《 Netty 源码剖析与实战 》视频课程

大纲

- 揭开 Netty 面纱
- 为什么舍近求远：不直接用 JDK 的 NIO ?
- 为什么孤注一掷：独选 Netty ?
- Netty 的前尘往事
- Netty 的现状与趋势

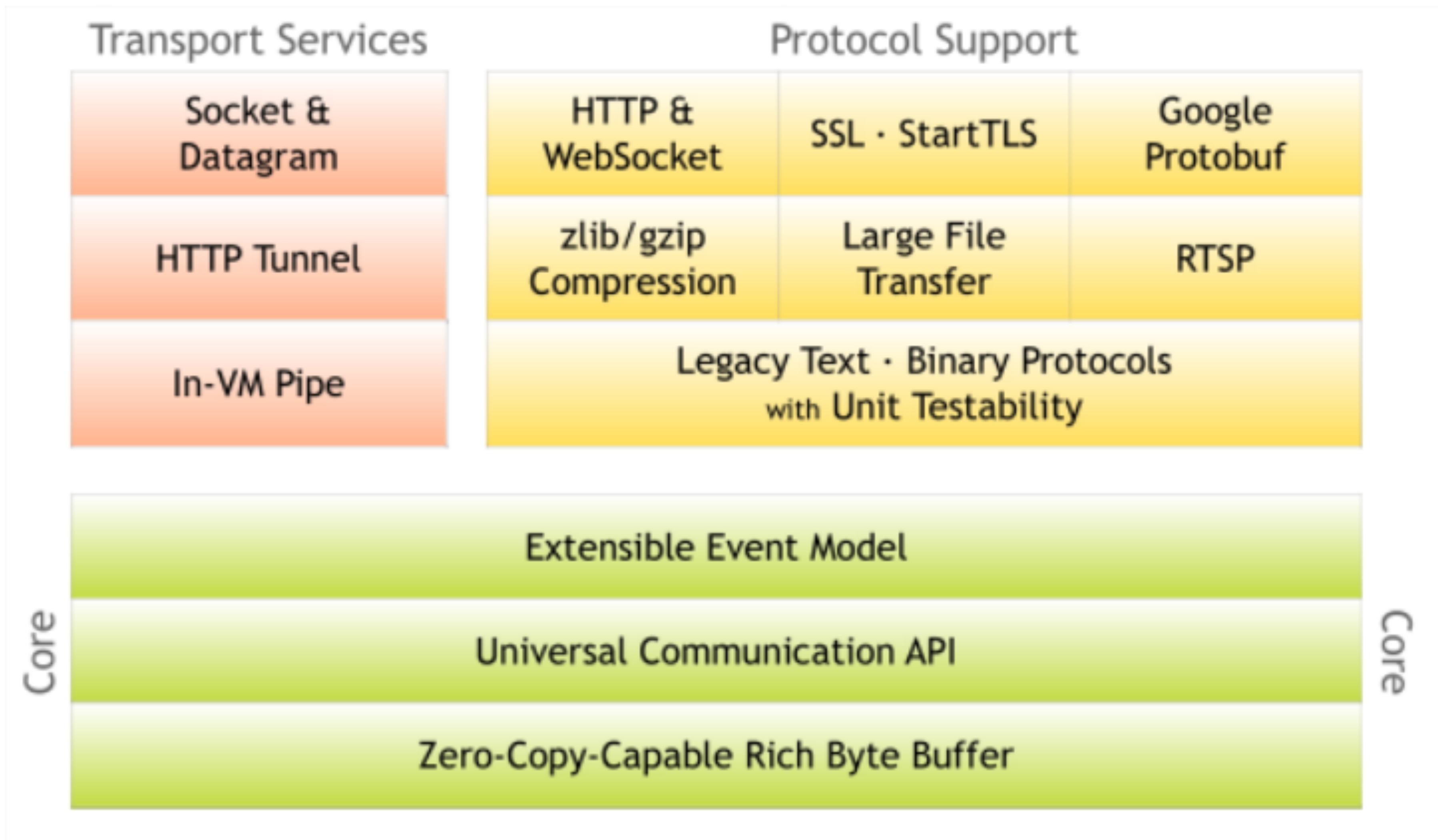
揭开 Netty 面纱

Netty 由 Trustin Lee（韩国，Line 公司）2004 年开发



- 本质：网络应用程序框架
- 实现：异步、事件驱动
- 特性：高性能、可维护、快速开发
- 用途：开发服务器和客户端

揭开 Netty 面纱



Talk is cheap. Show me the code!

为什么舍近求远：不直接用 JDK NIO



做得更多

做得更好



为什么舍近求远：不直接用 JDK NIO

Netty 做得更多：

- 支持常用应用层协议；
- 解决传输问题：粘包、半包现象；
- 支持流量整形；
- 完善的断连、Idle 等异常处理等。

为什么舍近求远：不直接用 JDK NIO

Netty 做得更好之一：规避 JDK NIO bug：

- 例 1：经典的 epoll bug：异常唤醒空转导致 CPU 100%

JDK-6670302 : (se) NIO selector wakes up with 0 selected keys infinitely [lnx 2.4]

Type: Bug

Component: core-libs

Sub-Component: java.nio

Affected Version: 6

Priority: P4

Status: Closed

Resolution: Won't Fix

OS: linux

CPU: x86

Submitted: 2008-03-03

Updated: 2013-08-20

Resolved: 2013-08-20

为什么舍近求远：不直接用 JDK NIO

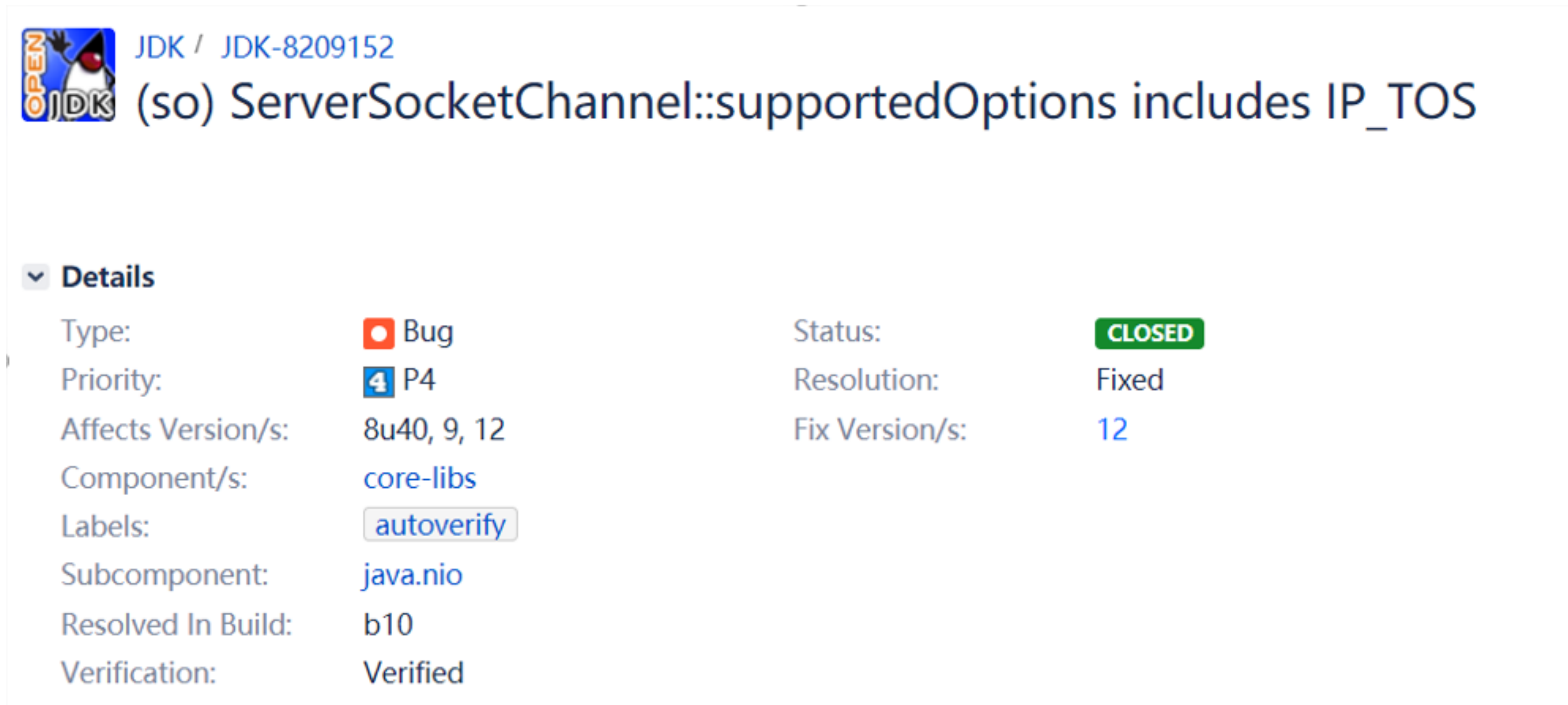
Netty 解决之道：检测问题发生，然后处理


```
} else if (SELECTOR_AUTO_REBUILD_THRESHOLD > 0 &&
    selectCnt >= SELECTOR_AUTO_REBUILD_THRESHOLD) {
    // The code exists in an extra method to ensure the method is not too big to inline as this
    // branch is not very likely to get hit very frequently.
    selector = selectRebuildSelector(selectCnt);
    selectCnt = 1;
    break;
}
```

io.netty.channel.socket.nio.NioChannelOption#setOption



为什么舍近求远：不直接用 JDK NIO

- 例 2：IP_TOS 参数（IP 包的优先级和 QoS 选项）使用时抛出异常
`java.lang.AssertionError: Option not found`



 **JDK / JDK-8209152**
(so) ServerSocketChannel::supportedOptions includes IP_TOS

Details

Type:	 Bug	Status:	CLOSED
Priority:	 P4	Resolution:	Fixed
Affects Version/s:	8u40, 9, 12	Fix Version/s:	12
Component/s:	core-libs		
Labels:	autoverify		
Subcomponent:	java.nio		
Resolved In Build:	b10		
Verification:	Verified		

<https://bugs.openjdk.java.net/browse/JDK-8209152>

为什么舍近求远：不直接用 JDK NIO

Netty 解决之道：遇到问题绕路走

```
55 // Internal helper methods to remove code duplication between Nio*Channel implementations.
56 @static <T> boolean setOption(Channel jdkChannel, NioChannelOption<T> option, T value) {
57     java.nio.channels.NetworkChannel channel = (java.nio.channels.NetworkChannel) jdkChannel;
58     if (!channel.supportedOptions().contains(option.option)) {
59         return false;
60     }
61     if (channel instanceof ServerSocketChannel && option.option == java.net.StandardSocketOptions.IP_TOS) {
62         // Skip IP_TOS as a workaround for a JDK bug:
63         // See http://mail.openjdk.java.net/pipermail/nio-dev/2018-August/005365.html
64         return false;
65     }
66     try {
67         channel.setOption(option.option, value);
68         return true;
69     } catch (IOException e) {
70         throw new ChannelException(e);
71     }
72 }
```

io.netty.channel.nio.NioEventLoop#select

为什么舍近求远：不直接用 JDK NIO

Netty 做得更好之二：API 更友好更强大

- JDK 的 NIO 一些 API 不够友好，功能薄弱，例如 ByteBuffer -> Netty's ByteBuf
- 除了 NIO 外，也提供了其他一些增强：ThreadLocal -> Netty's FastThreadLocal

为什么舍近求远：不直接用 JDK NIO

Netty 做得更好之三：隔离变化、屏蔽细节






- 隔离 JDK NIO 的实现变化：nio -> nio2(aio)->
- 屏蔽 JDK NIO 的实现细节

为什么舍近求远：不直接用 JDK NIO

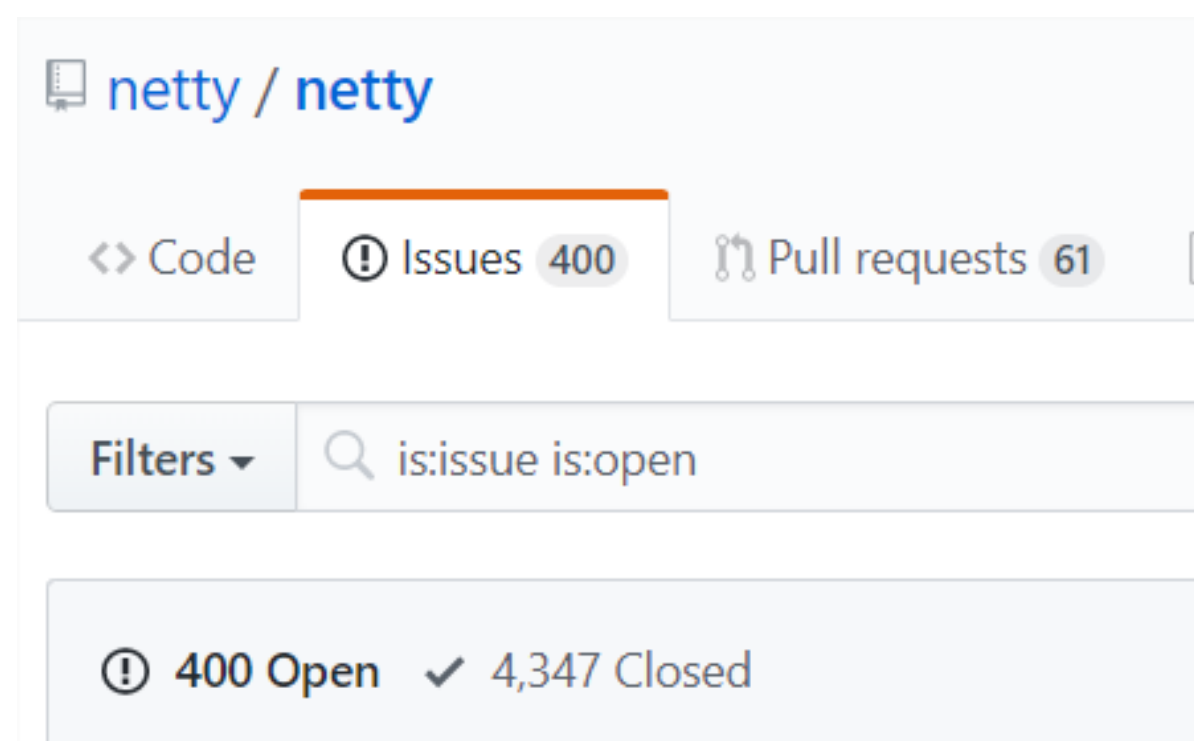
自己直接用 JDK NIO 实现的可能性？

- 大概写多少行代码？

Netty Transport Package:

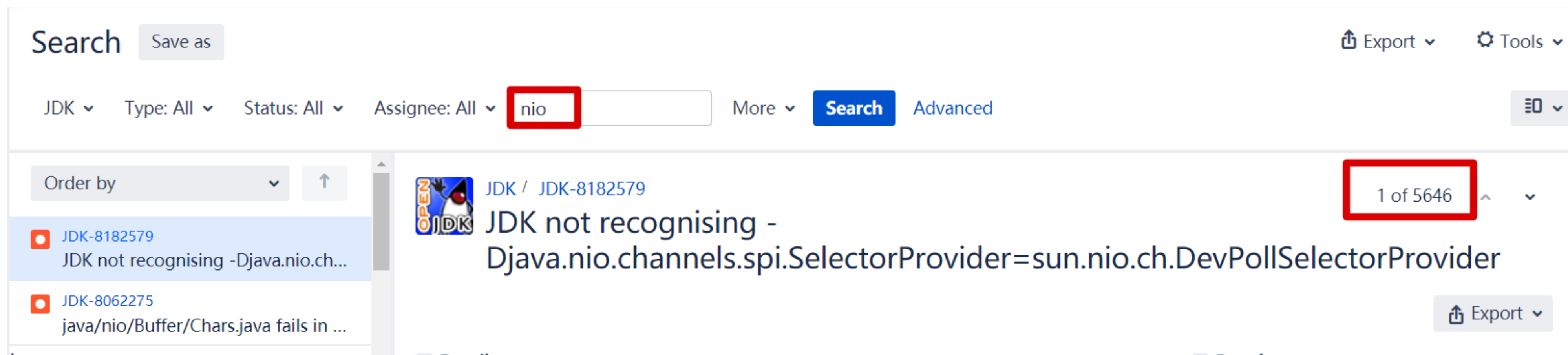
Source File	Total Lines	Source Code Lines ▼	Source Code Lines [%]
 Total:	 31582	 18584	 59%  9173

- 可能面对的问题：400 open, 4347 closed （2019/09 统计）



为什么舍近求远：不直接用 JDK NIO

- “踏平” 多少 JDK NIO bug：5646



- 未来能持续维护多久？Netty 已维护 15 年（from 2004 to 2019）

直接使用 JDK NIO = 一个人在战斗！

为什么孤注一掷：独选 Netty ?

- Apache Mina
- Sun Grizzly
- Apple Swift NIO、ACE 等
- Cindy 等
- Tomcat、Jetty

为什么孤注一掷：独选 Netty ?

- 为什么不选 Apache Mina :

同一作者，推荐 Netty ;

Currently, most features available in MINA are also available in Netty. In my opinion, Netty has cleaner and more documented API since **Netty is the result of trying to rebuild MINA from scratch and address the known issues**. If you find that an essential feature is missing, please feel free to post your suggestion to the forum. I'd be glad to address your concern.

It is also important to note that Netty has faster development cycle. Simply, check out the release date of the recent releases. Also, you should consider that MINA team will proceed to a major rewrite, MINA 3, which means they will break the API compatibility completely.

[share](#) [improve this answer](#)

answered Jan 4 '10 at 15:14



trustin

9,644 ● 6 ● 33 ● 47

为什么孤注一掷：独选 Netty ?

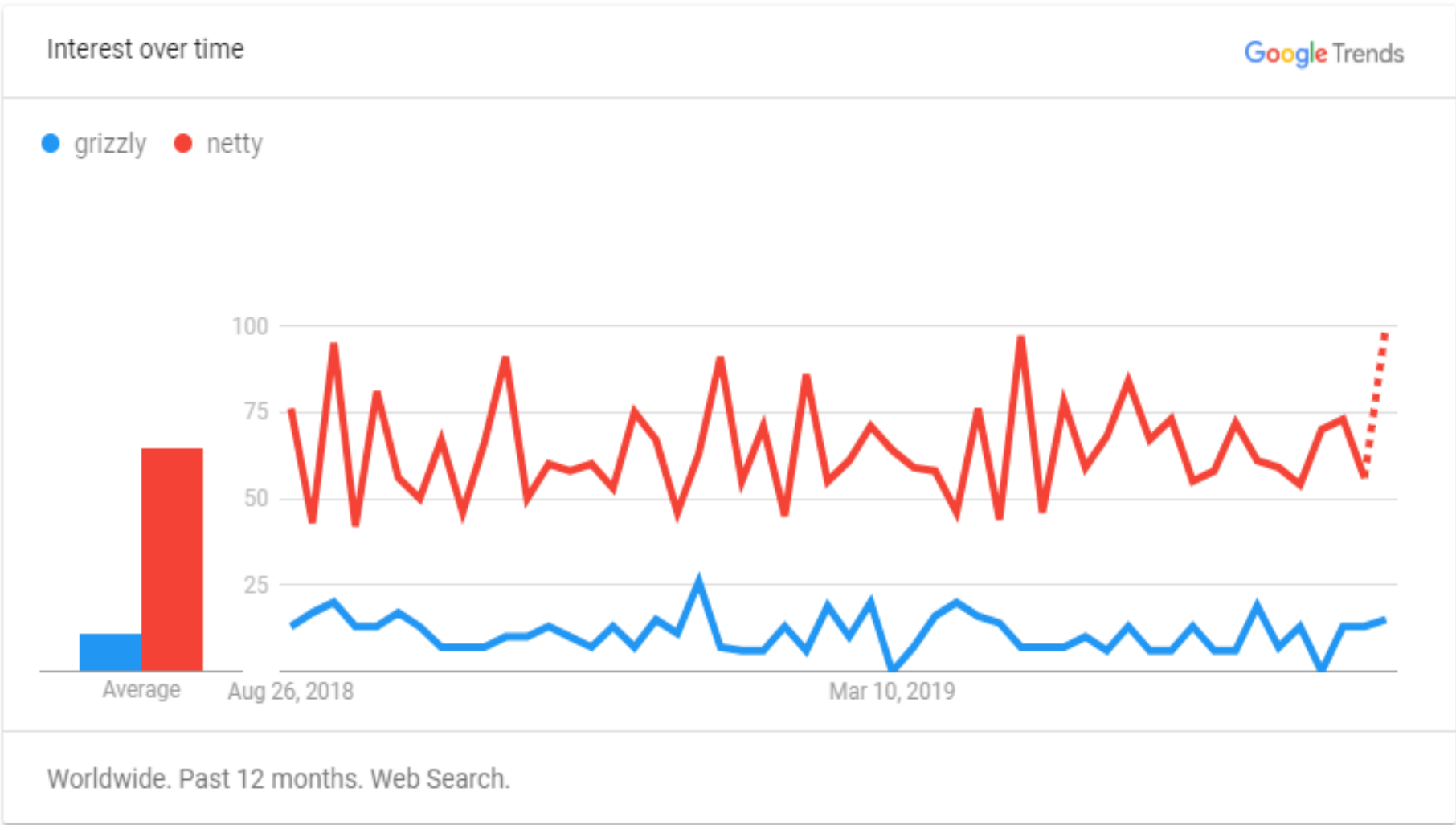
- 为什么不选 Sun 的 Grizzly :

三少：用得少、文档少，更新少。

Grizzly		Netty
Repository		
98	★ Stars	20,168
15	👁 Watchers	1,751
29	🔗 Forks	9,232
over 2 years ago	🕒 Last Commit	about 1 month ago
More		
L4	Code Quality	L2
Java	</> Language	Java
Networking	🔖 Tags	Networking

Interest over time of Grizzly and Netty

Note: It is possible that some search terms could be used in multiple areas and that could skew some graphs.



<https://java.libhunt.com/compare-grizzly-vs-netty>

为什么孤注一掷：独选 Netty ?

- 为什么不选 Apple SwfitNIO 、ACE 等：

其他语言，不考虑。

- 为什么不选 Cindy 等：

生命周期不长。

- 为什么不选 Tomcat、Jetty：

还没有独立出来。

Netty 的前尘往事

从归属组织上看发展

- JBoss
- Netty

Netty 的前尘往事

从版本演变上看发展

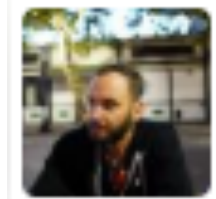
- 2004 年 6 月 Netty2 发布

声称 Java 社区中第一个基于事件驱动的应用网络框架

- 2008 年 10 月 Netty3 发布
- 2013 年 7 月 Netty4 发布
- 2013 年 12 月发布 5.0.0.Alpha1
- 2015 年 11 月废弃 5.0.0

Netty 的前尘往事

- 题外 1： 废弃 5.0 原因：



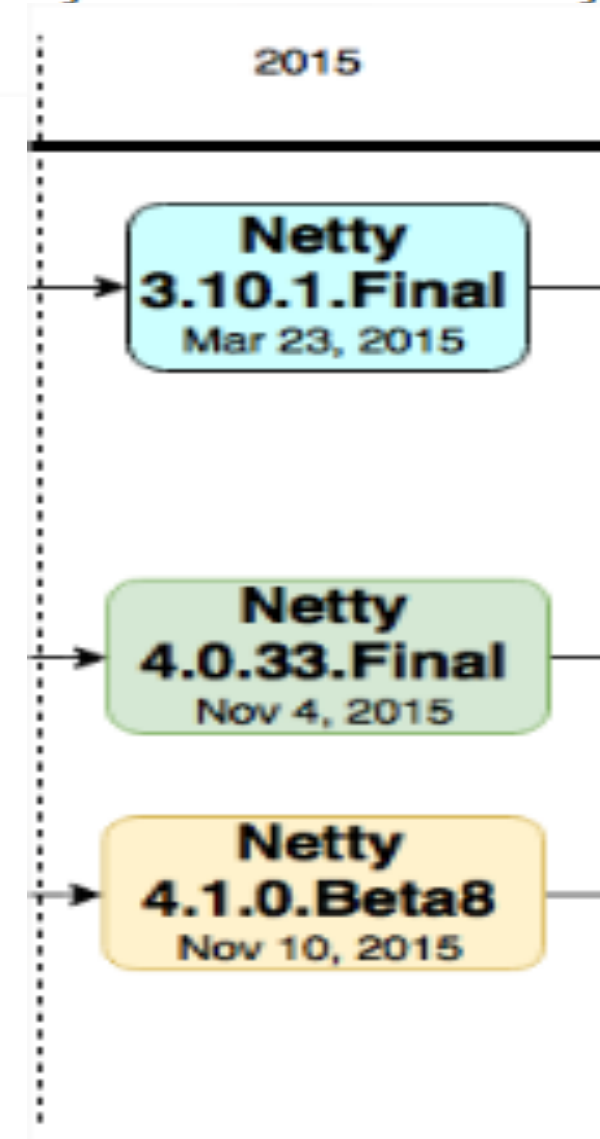
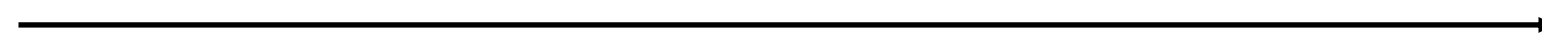
normanmaurer commented on 11 Nov 2015

Member +😊 ...

After talking with @Scottmitch and also with @nmittler I would like to propose "dropping" current master and so not release any new 5.0 version for now.

The major change of using a ForkJoinPool increases complexity and has not demonstrated a clear performance benefit. Also keeping all the branches in sync is quite some work without a real need for it as there is nothin in current master which I think justifies a new major release.

- 复杂
- 没有证明明显性能优势
- 维护不过来



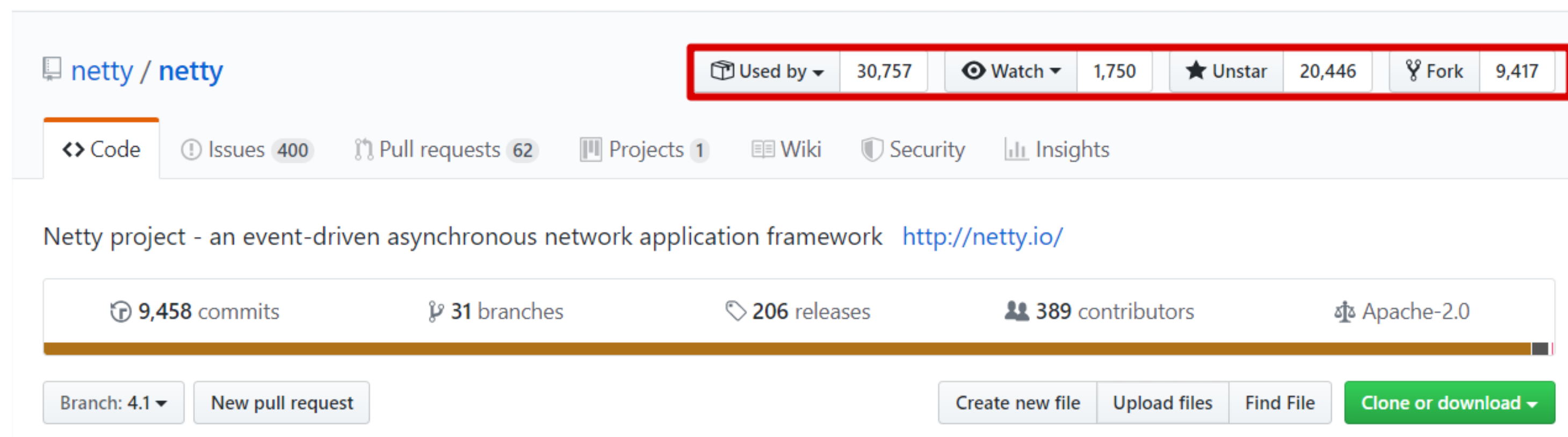
Netty 的前尘往事

- 题外 2：与 Apache Mina 的关系？
 - 同一作者开发，都处于维护阶段（当前：MINA 2.1.3：2019/06；Netty 4.1.39.Final：2019/08）
 - 2004 年 6 月 Netty2 发布
 - 2005 年 5 月 Mina 发布
 - Alex 为 Apache Directory 开发网络框架，但是觉得不好用，看到 Netty2 后 找到作者邀请合作开发，结合两种框架，随后有了 MINA。

Netty 的现状与趋势

社区现状:

- <https://github.com/netty/netty>



- Stars: 20000+ (top 25 repository on java)

<https://github.com/search?l=java&q=stars%3A%3E20000&type=Repositories>

Netty 的现状与趋势

- 维护者
 - 22 members (core: [Trustin Lee](#) and [Norman Maurer](#))
- 分支
 - 4.1 master (16 年 5 月创建, hours/days ago) 支持 Android
 - 4.0 (13 年 7 月创建, Latest commit [be5b5a3](#) on 13 Feb 2018) 线程模型优化、包结构、命名

Netty 的现状与趋势

最新版本

- Netty 4.1.39.Final (2019 年 8 月)
- Netty 4.0.56.Final (2018 年 2 月)
- Netty 3.10.6.Final (2016 年 6 月)

Netty 的现状与趋势

应用现状：

- 截止 2019 年 9 月，30000+ 项目在使用
- 统计方法：依赖项中声明 `io.netty:netty-all`
- 未考虑的情况：非开源软件和 Netty 3.x 使用者

你以为没有 Netty 的地方其实大多都有！

Netty 的现状与趋势

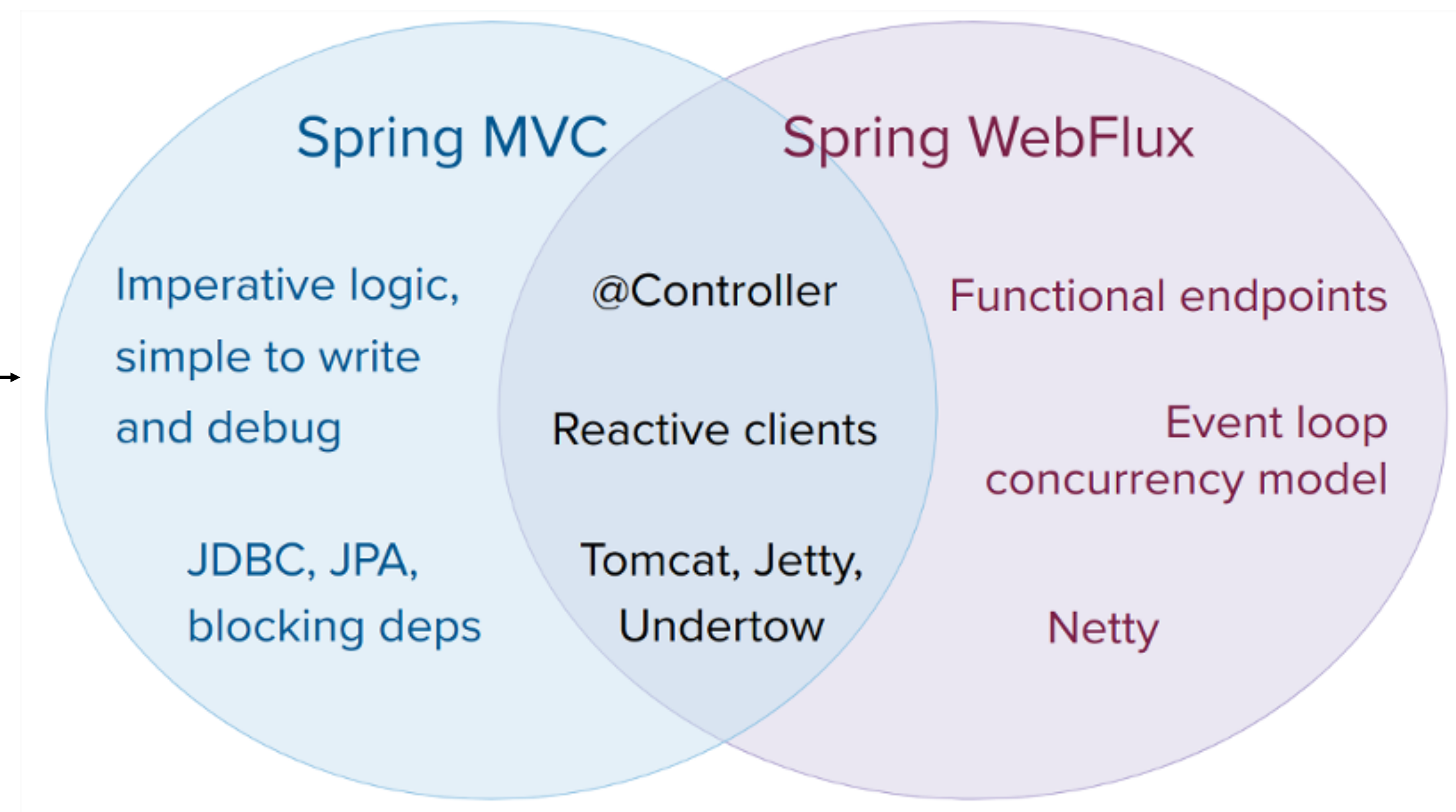
一些典型项目：

- 数据库：Cassandra
- 大数据处理：Spark、Hadoop
- Message Queue：RocketMQ

Netty 的现状与趋势


一些典型项目：

- 检索：Elasticsearch
- 框架：gRPC、Apache Dubbo、Spring5 →
- 分布式协调器：ZooKeeper
- 工具类：async-http-client
- 其他参考：<https://netty.io/wiki/adopters.html>



Netty 的现状与趋势

- 趋势
- 更多流行协议的支持
- 紧跟 JDK 新功能步伐
- 更多易用、人性化的功能：
IP 地址黑白名单、流量整形等
- 应用越来越多



```
▶ codec-dns [netty-codec-dns]
▶ codec-haproxy [netty-codec-haproxy]
▶ codec-http [netty-codec-http]
▶ codec-http2 [netty-codec-http2]
▶ codec-memcache [netty-codec-memcache]
▶ codec-mqtt [netty-codec-mqtt]
▶ codec-redis [netty-codec-redis]
▶ codec-smtp [netty-codec-smtp]
▶ codec-socks [netty-codec-socks]
▶ codec-stomp [netty-codec-stomp]
▶ codec-xml [netty-codec-xml]
```




扫码试看/订阅

《 Netty 源码剖析与实战 》视频课程