# 简介

高性能异步非阻塞响应式框架。

# 主页

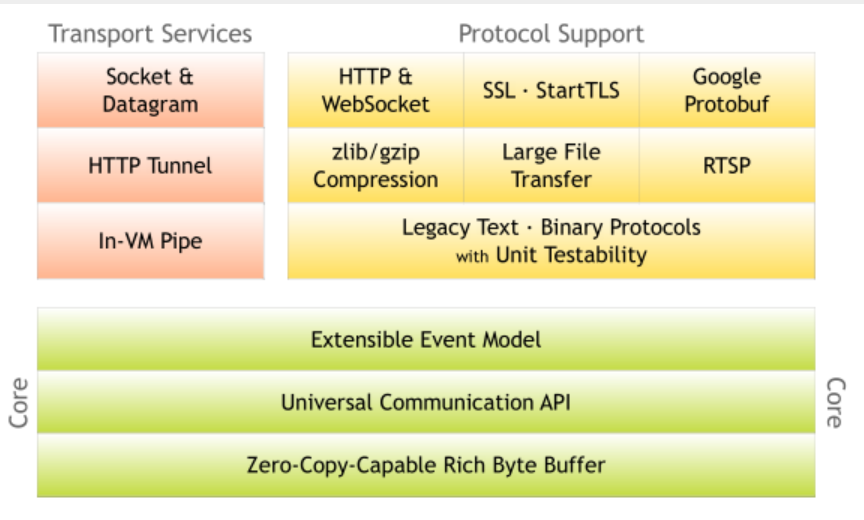
<https://netty.io/>

# 特性

# 用户

# 原理

# 设计

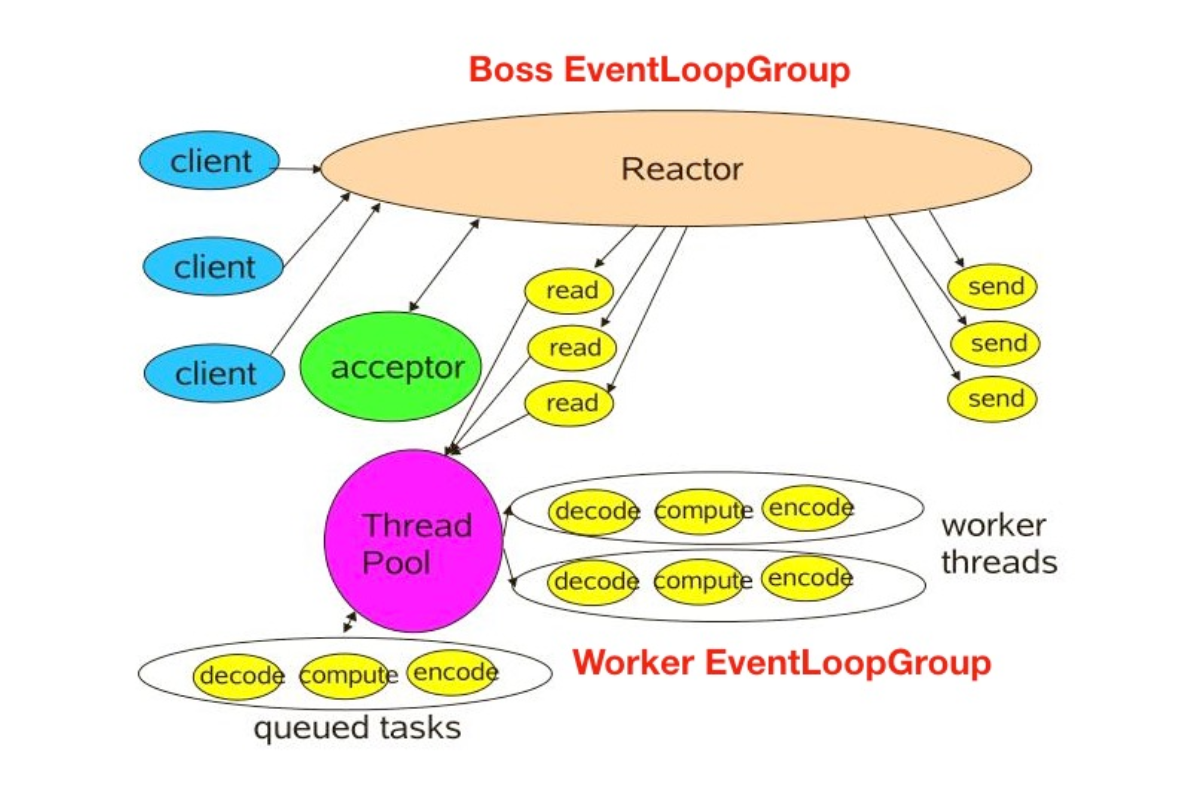


# 性能

# 适用

# 概念

## Reactor



## Proactor

# 用法

# 问答

## ****为什么Netty使用NIO而不是AIO？****

1. Netty不看重Windows上的使用，在Linux系统上，AIO的底层实现仍使用EPOLL，没有很好实现AIO，因此在性能上没有明显的优势，而且被JDK封装了一层不容易深度优化
2. Netty整体架构是reactor模型, 而AIO是proactor模型, 混合在一起会非常混乱,把AIO也改造成reactor模型看起来是把epoll绕个弯又绕回来
3. AIO还有个缺点是接收数据需要预先分配缓存, 而不是NIO那种需要接收时才需要分配缓存, 所以对连接数量非常大但流量小的情况, 内存浪费很多
4. Linux上AIO不够成熟，处理回调结果速度跟不到处理需求，比如外卖员太少，顾客太多，供不应求，造成处理速度有瓶颈（待验证）

**作者原话：**  
**Not faster than NIO (epoll) on unix systems (which is true)  
There is no daragram suppport  
Unnecessary threading model (too much abstraction without usage)**

# 最佳实践

# 优缺点

# 对比

# 优化

# 思考