

www.qconferences.com
www.qconbeijing.com
www.qconshanghai.com

QCon

伦敦 | 北京 | 东京 | 纽约 | 圣保罗 | 上海 | 旧金山
London • Beijing • Tokyo • New York • Sao Paulo • Shanghai • San Francisco

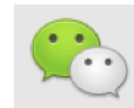
QCon全球软件开发大会

International Software Development Conference

InfoQ^{ueue}



@InfoQ



infoqchina

软件
正在改变世界!



Netty架构剖析和行业应用

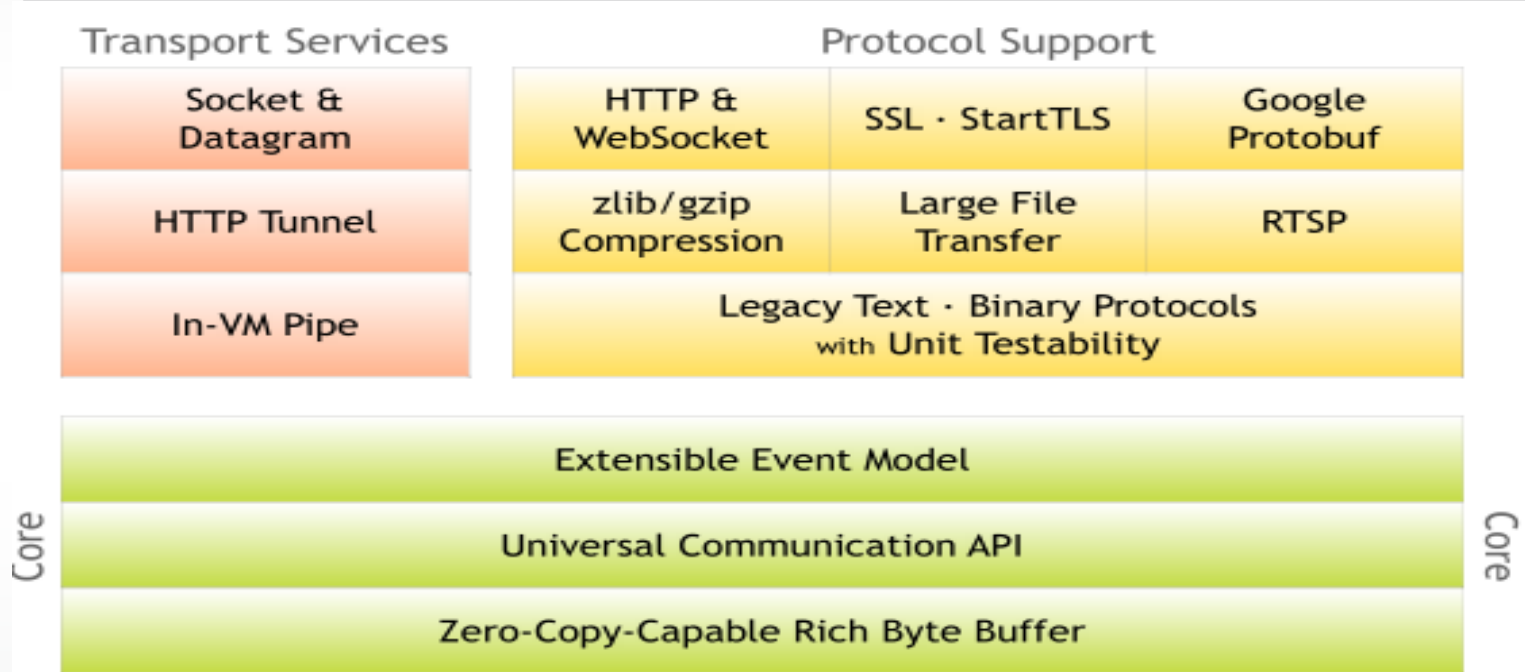
李林锋 neu_lilinfeng@sina.com

新浪微博 Nettying

微信 Nettying

- ✓ **Netty**介绍
- ✓ **Netty**架构剖析
- ✓ **Mina VS Netty**
- ✓ **Netty**行业应用

Netty介绍



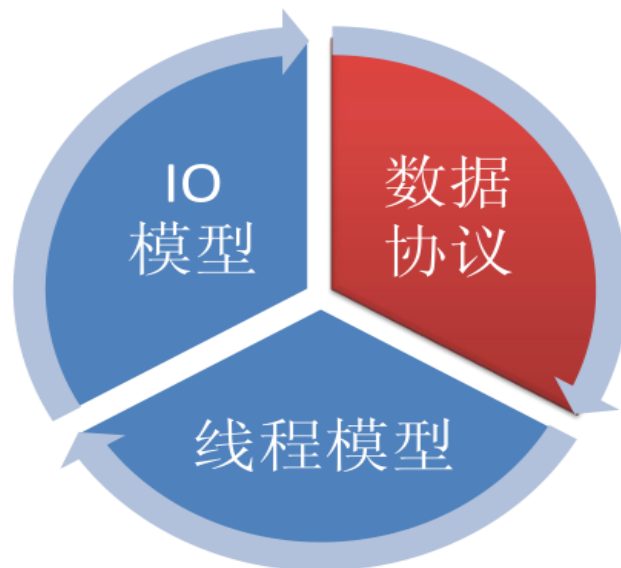
Netty是一个异步、事件驱动的网络应用框架。基于**Netty**，可以快速的开发和部署高性能、高可用的网络服务端和客户端应用。

Netty架构剖析-性能模型

通信性能三要素

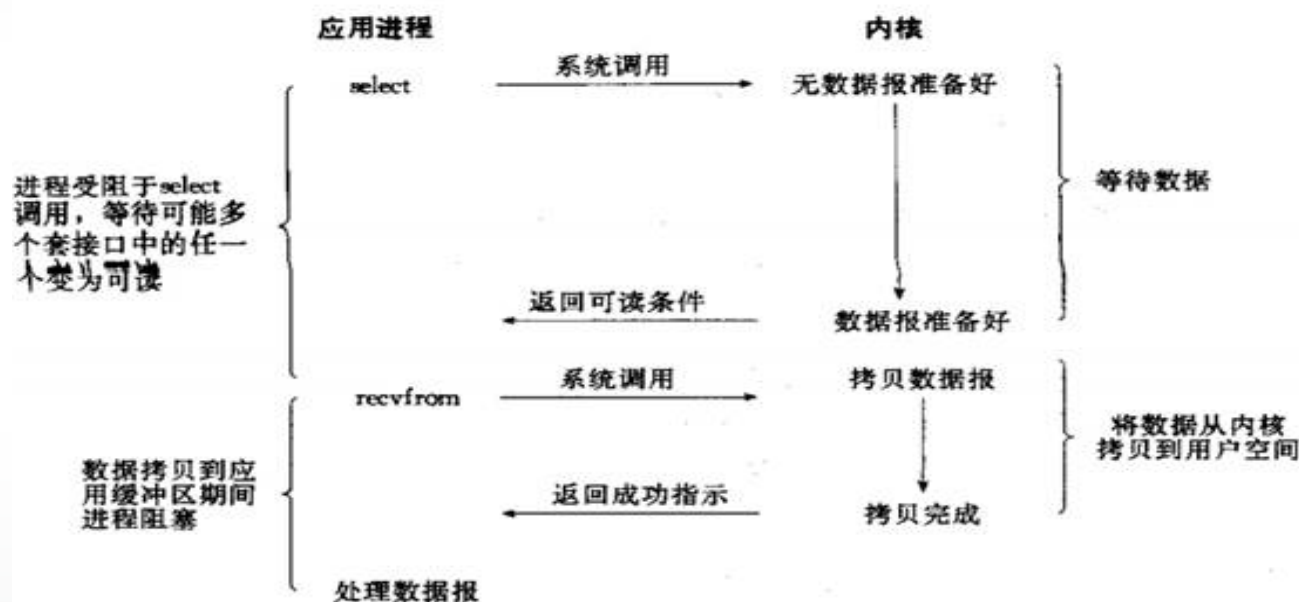
- ✓ 传输
- ✓ 协议
- ✓ 线程

Netty架构剖析-性能模型



- ✓ **IO模型**：异步非阻塞I/O
- ✓ **数据协议**：可定制的编解码框架
- ✓ **线程模型**：**Reactor**线程模型

Netty架构剖析-性能模型



- ✓ JDK1.4开始提供非阻塞I/O (NIO)
- ✓ JDK1.5_update10版本使用epoll替代了传统的select/poll

Netty架构剖析-性能模型

零拷贝

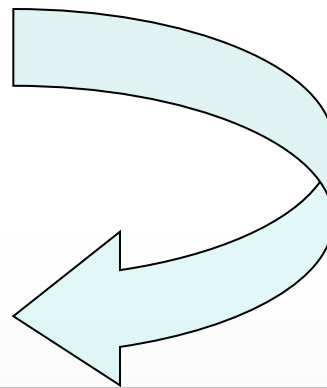
- ✓ Netty的接收和发送ByteBuffer采用**DIRECT BUFFERS**
- ✓ Netty提供了组合Buffer对象，可以聚合多个ByteBuffer对象，用户可以像操作一个Buffer那样方便的对组合Buffer进行操作
- ✓ Netty的文件传输采用了transferTo方法

Netty架构剖析-性能模型

内存池

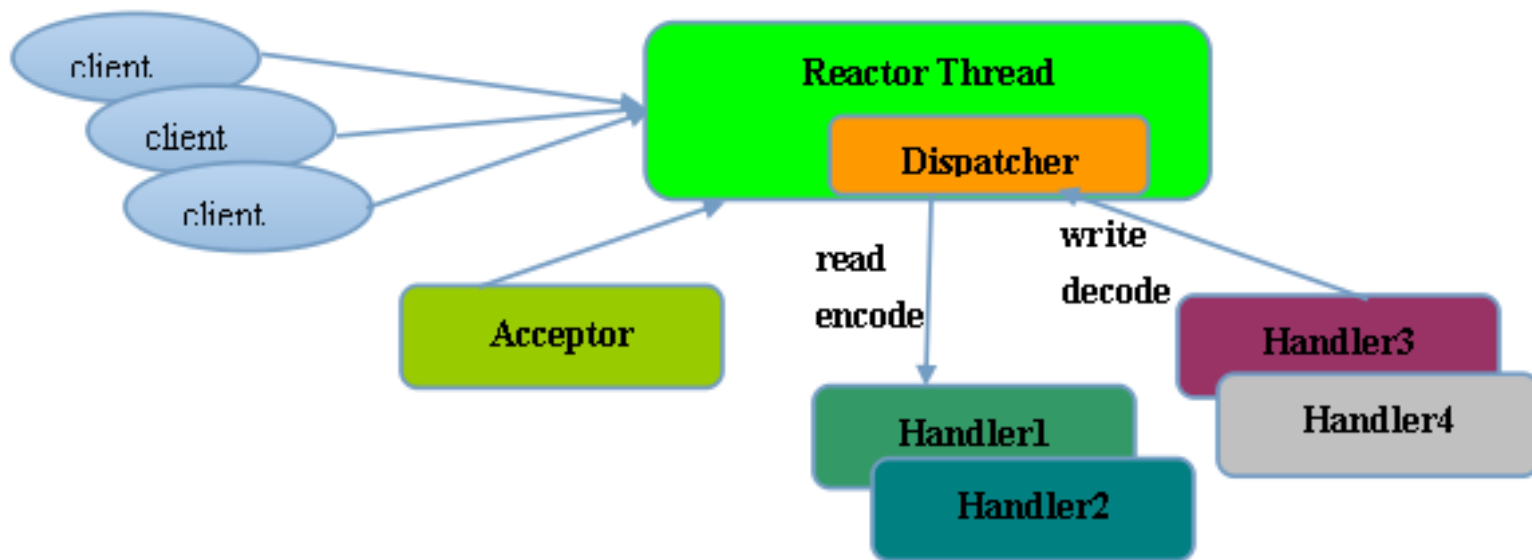
```
int loop = 3000000;
long startTime = System.currentTimeMillis();
ByteBuf poolBuffer = null;
for (int i = 0; i < loop; i++) {
    poolBuffer = PooledByteBufAllocator.DEFAULT.directBuffer(1024);
    poolBuffer.writeBytes(CONTENT);
    poolBuffer.release();
}
```

执行300万次，采用内存池性能比朝生夕灭的ByteBuf高23倍左右



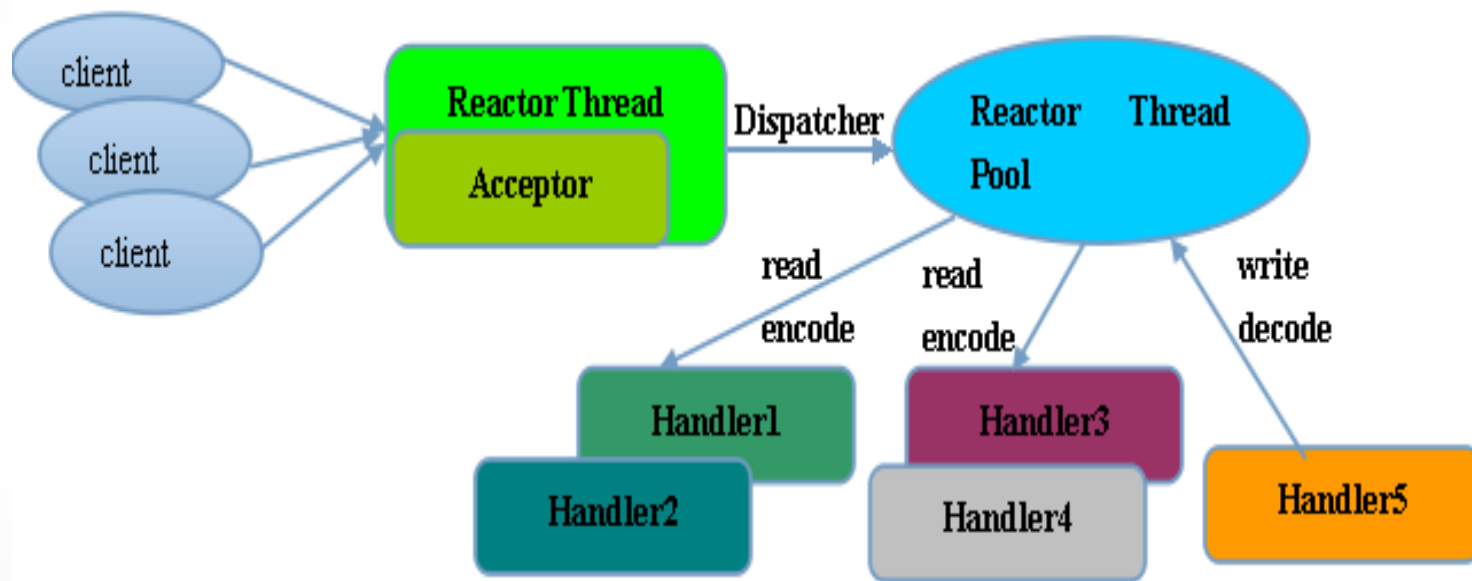
```
long startTime2 = System.currentTimeMillis();
ByteBuf buffer = null;
for (int i = 0; i < loop; i++) {
    buffer = Unpooled.directBuffer(1024);
    buffer.writeBytes(CONTENT);
}
```

Netty架构剖析-单线程模型



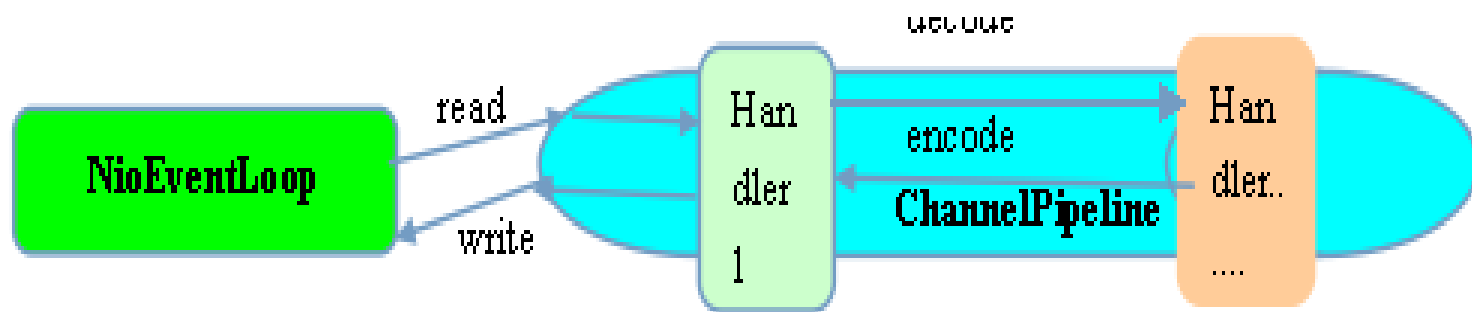
Reactor单线程模型，指的是所有的IO操作都在同一个NIO线程上面完成

Netty架构剖析-多线程模型



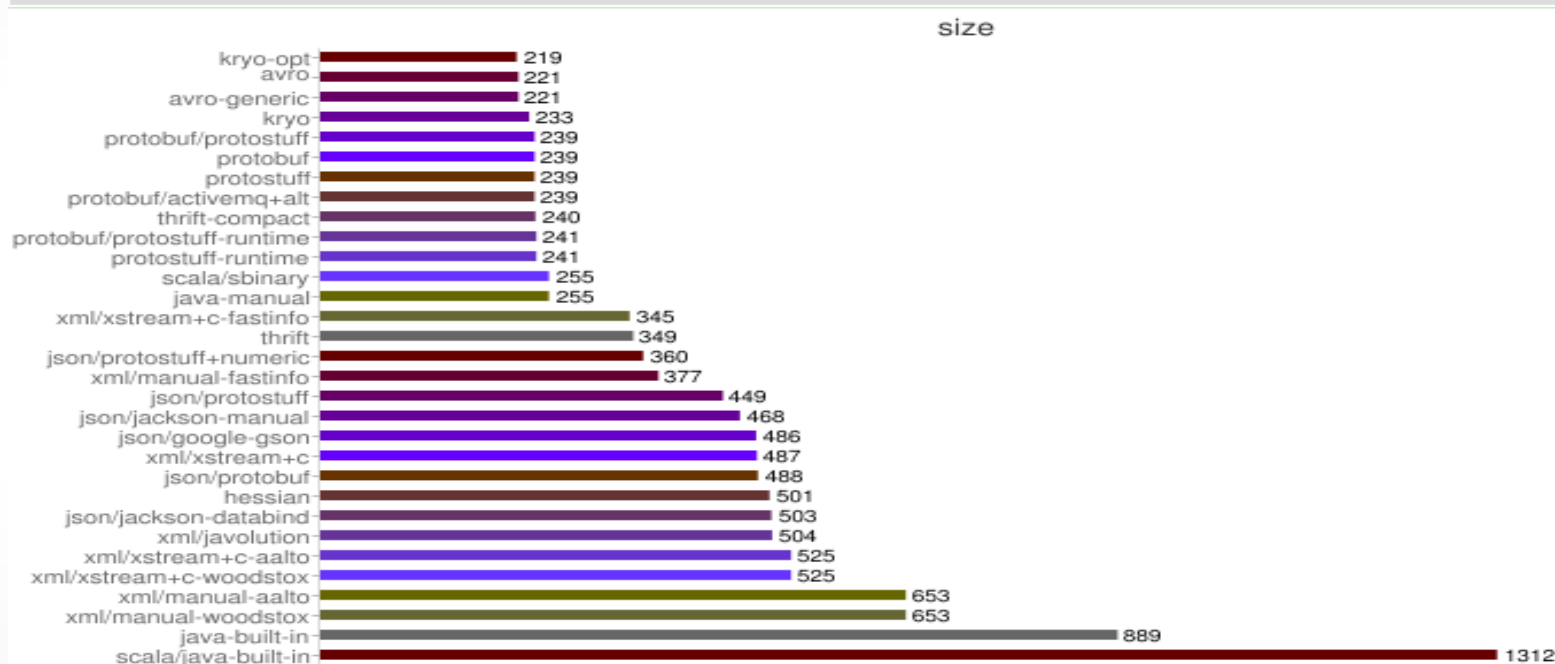
- ✓ 专门一个NIO线程-Acceptor线程用于监听服务端，接收客户端的TCP连接请求；
- ✓ 网络IO操作-读、写等由一个NIO线程池负责

Netty架构剖析-线程模型



无锁化的串行设计：为了尽可能提升性能，**Netty** 采用了串行无锁化设计，在**IO**线程内部进行串行操作，避免多线程竞争导致的性能下降。

Netty架构剖析-协议



影响序列化性能的关键因素总结如下：

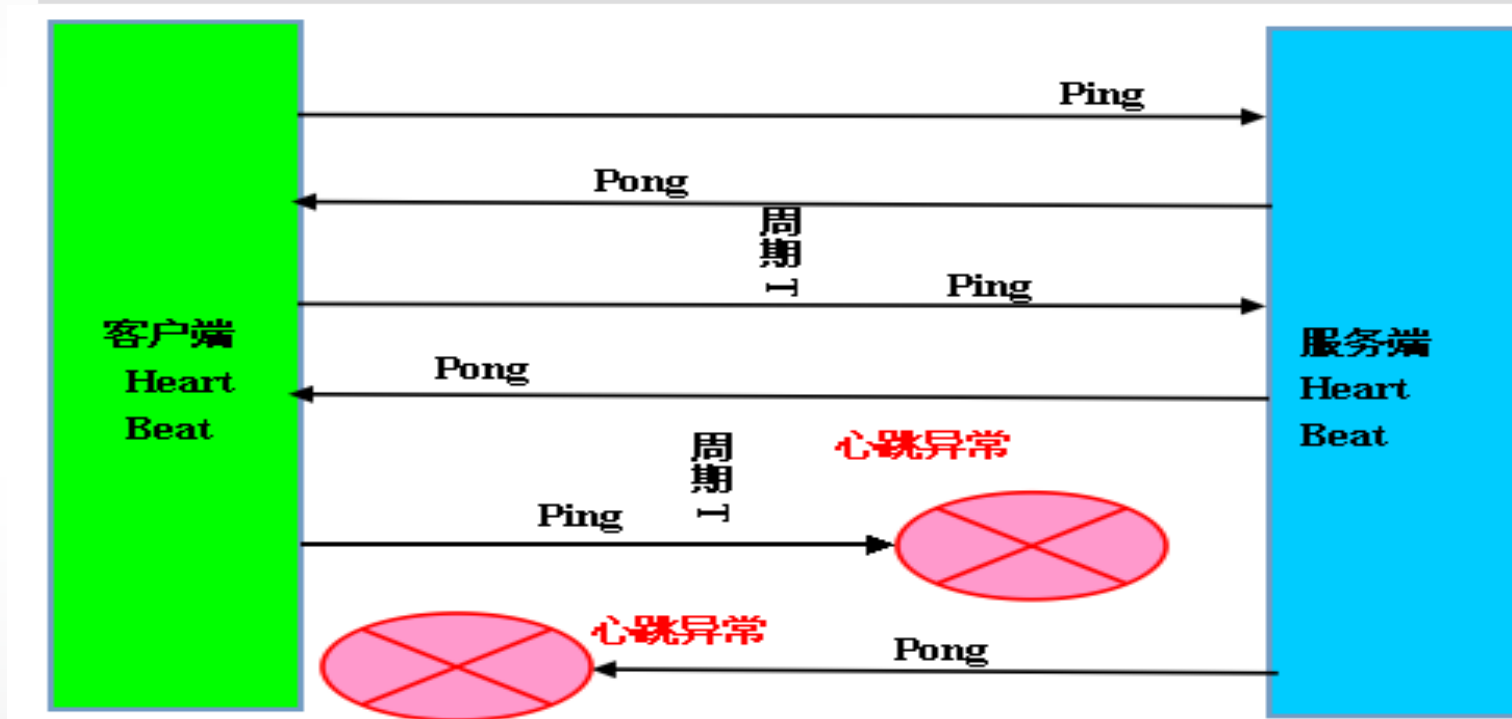
1. 序列化后的码流大小（网络带宽的占用）；
2. 列化&反序列化的性能（CPU资源占用）。

Netty架构剖析-协议

可定制的序列化框架

- ✓ ProtoBuf
- ✓ Java序列化
- ✓ Xml
- ✓ 用户自定义序列化框架

Netty架构剖析-可靠性



心跳检测

读空闲；写空闲；读写空闲

Netty架构剖析-可靠性

Reactor线程的保护

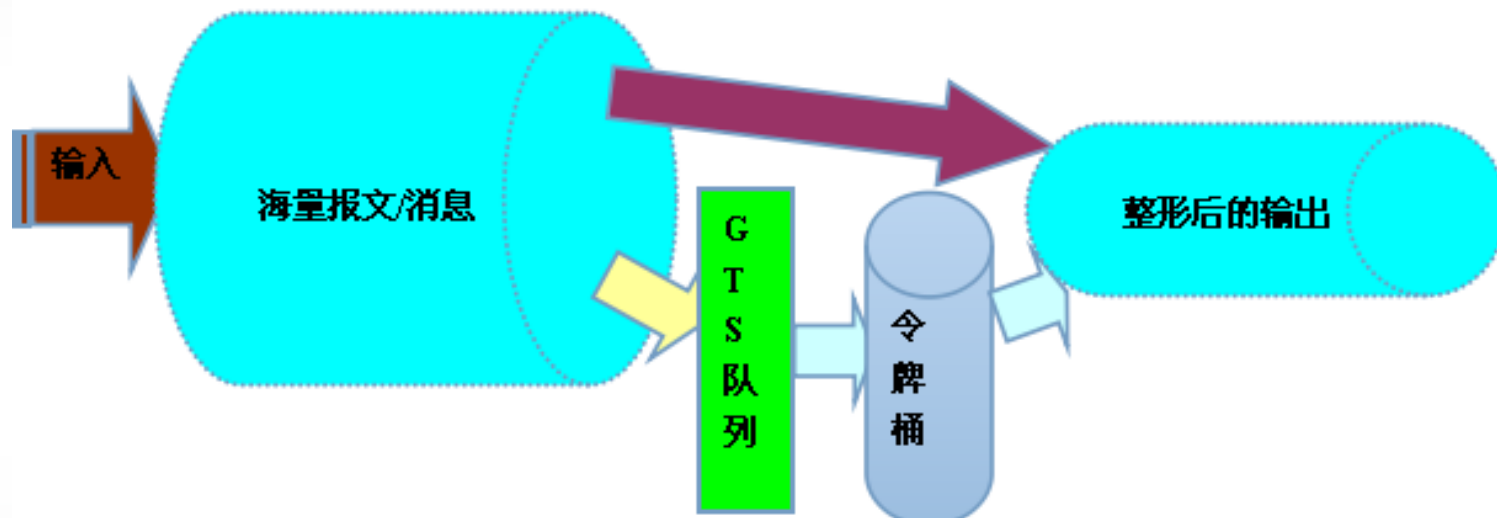
- ✓ 某个消息的异常不应该导致整条链路不可用；
- ✓ 某条链路不可用不应该导致其它链路不可用；
- ✓ JDK epoll 空轮询BUG的检测和修复

Netty架构剖析-可靠性

内存保护

- ✓ 缓冲区的内存泄漏保护：内存检测和释放
- ✓ 缓冲区内存溢出保护：对缓冲区进行上限保护

Netty架构剖析-可靠性

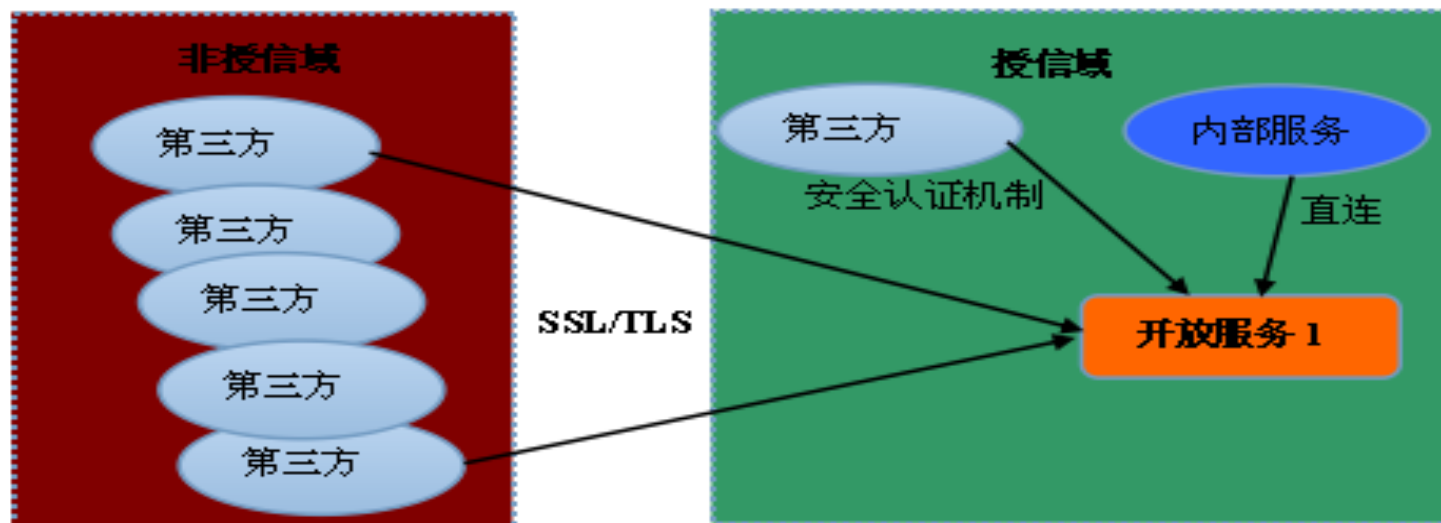


流量整形

流量整形（Traffic Shaping）是一种主动调整流量输出速率的措施。一个典型应用是基于下游网络结点的TP指标来控制本地流量的输出。

Netty提供全局级和链路级流量整形功能。

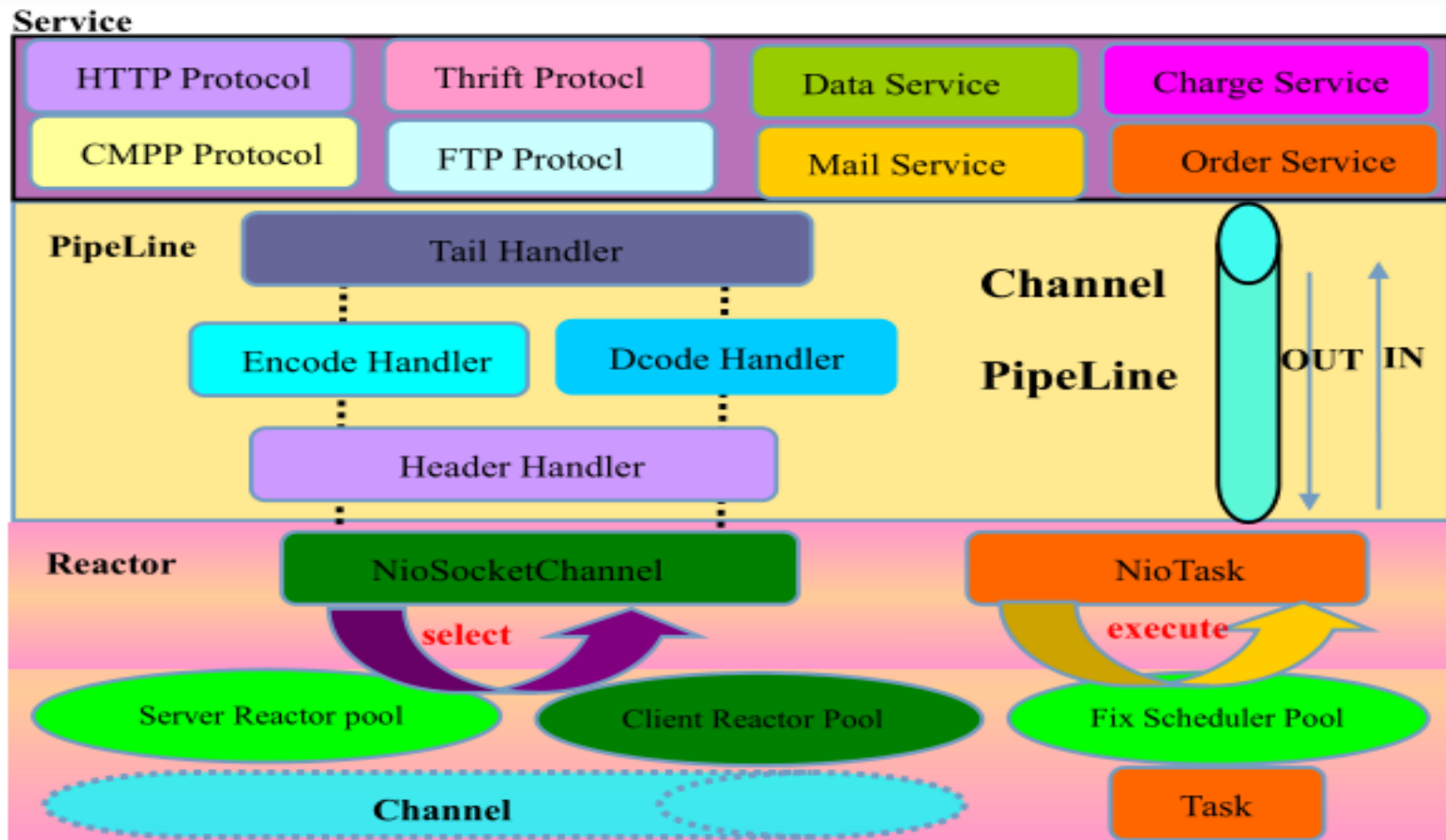
Netty架构剖析-安全性



Netty安全性

- ✓ **SSL认证**：SSL单向认证、SSL双向认证
- ✓ **IP黑白名单校验**
- ✓ **接入认证**

Netty逻辑架构



Mina VS Netty

相同点：

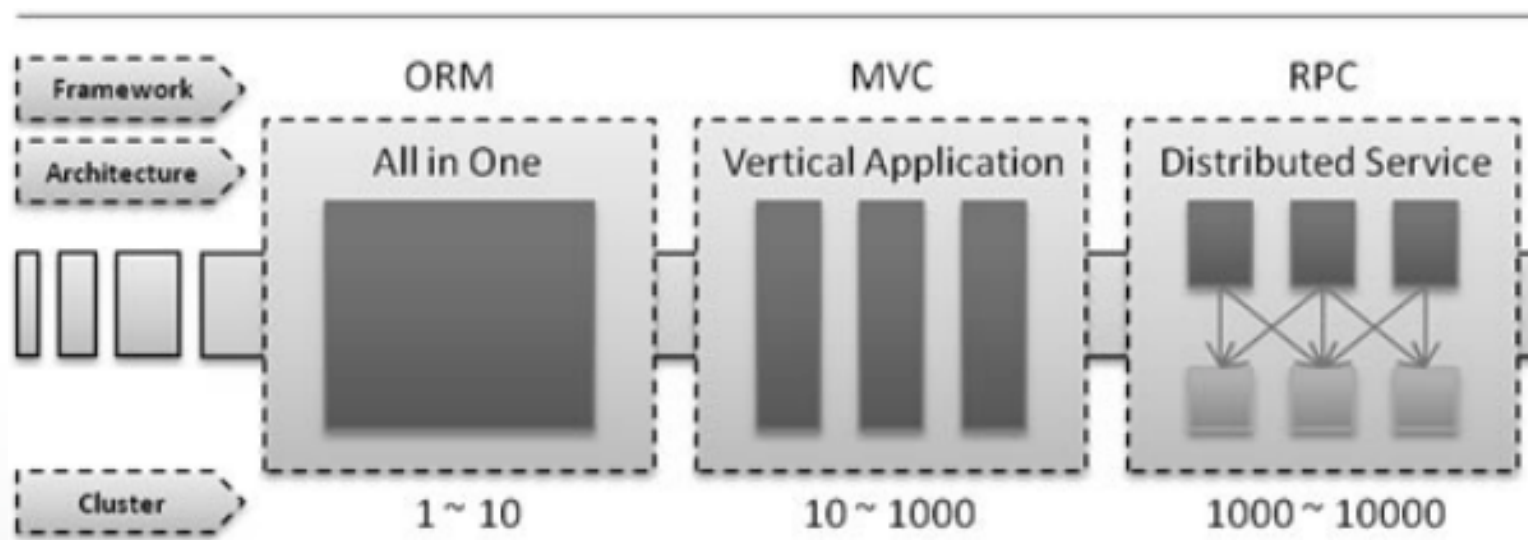
- ✓ 基于**Reactor**模式的**NIO**框架
- ✓ 基于链式的**Handler**编排机制
- ✓ 对**JDK**的**NIO**类库进行了封装，屏蔽底层细节

Mina VS Netty

不同点：

- ✓ **ByteBuffer**的封装和功能不同
- ✓ **I/O** 线程模型不同
- ✓ 内置的编解码能力不同
- ✓ 内置的应用层协议不同
- ✓ 当前活跃度不同
- ✓ 功能的丰富度不同
- ✓ 未来的前景不同

行业应用-互联网应用



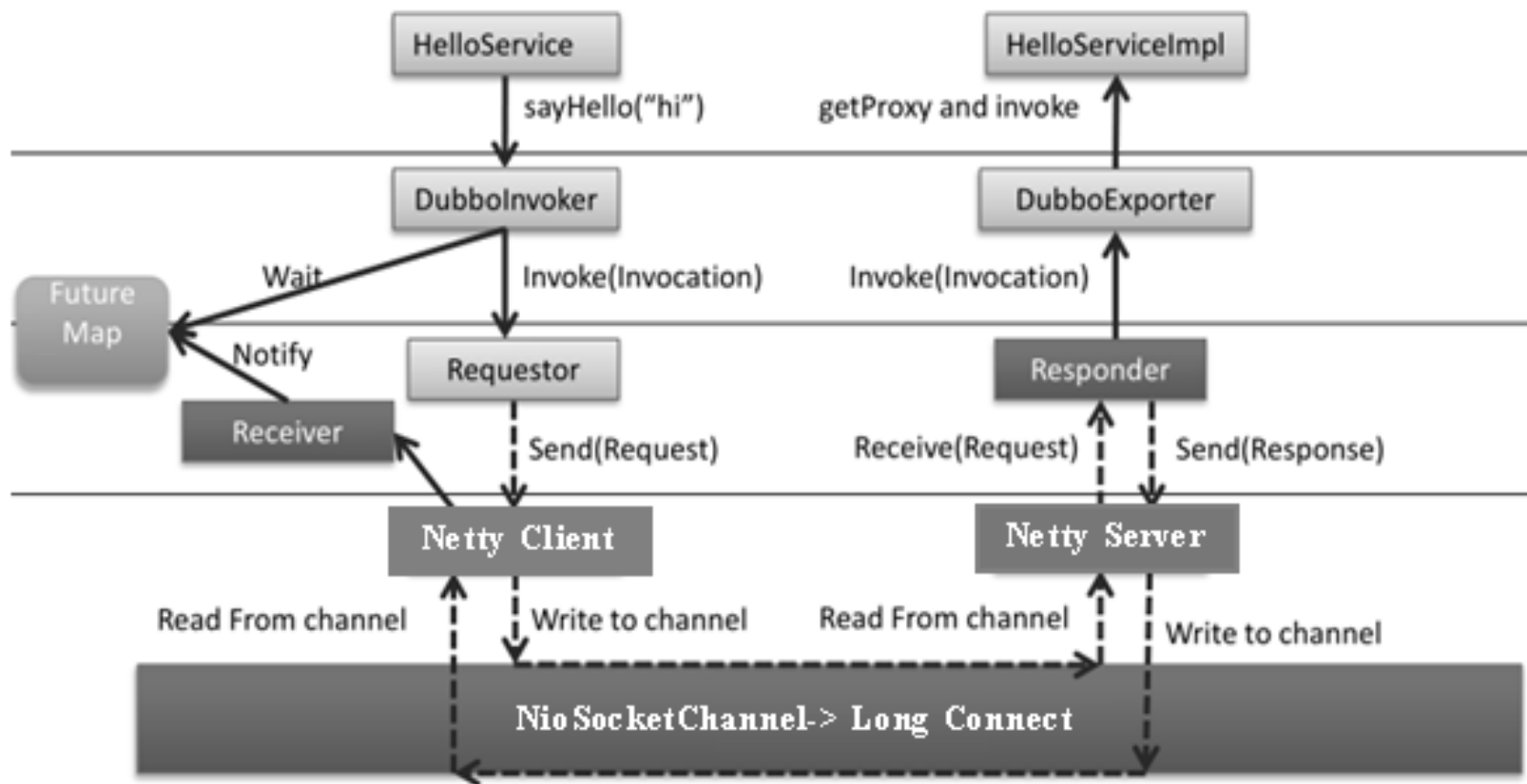
互联网架构演进：由垂直架构向分布式服务框架演进

Netty互联网应用-Dubbo

Dubbo的RPC远程服务调用默认使用Netty +Dubbo协议实现：

- ✓ **Dubbo RCP**框架默认推荐使用**Dubbo**协议进行通信和数据传输，相比于老的**Hessian**协议性能更高
- ✓ 支持异步**I/O**通信

Netty互联网应用-Dubbo



Netty互联网应用-Twitter

Netty在Twitter得到了大量的应用：

- ✓ **Finagle**是**Twitter**协议无关的**RPC**框架，用于实现大部分内部服务，如搜索，它的传输层建立在**Netty**之上
- ✓ **TFE**（**Twitter**前端）是专有的填鸭式反向代理，它使用**Netty**为大部分面向公众的**HTTP**和**SPDY**流量提供服务
- ✓ **Cloudhopper**每月使用**Netty**向遍布世界各地数以百计的运营商发送数十亿条短信息

Netty互联网应用-FaceBook

- ✓ **Nifty**是facebook公司开源的，基于**netty**的**thrift**服务端和客户端实现。
- ✓ **Thrift**是一个可伸缩的跨语言的服务开发框架，利用**Nifty**可以快速开发基于**Netty**的**thrift**服务端和客户端程序。

Netty互联网应用-雅虎

Yahoo工程师Bobby Evans：Netty让Storm飞速运行：

✓ 在小规模测试中（没有资源冲突），Netty比zeromq更快（40-100%）

✓ 在大规模测试中，当把它限制在单线程上，我们每秒能够获得比zeromq多85%到111%的消息

✓ Netty现在是Yahoo Storm集群的默认消息层框架

Netty行业应用-大数据

Apache Avro是**Hadoop**、**Flume**等的底层序列化和数据传输框架。

老版本的**Avro**基于**HTTP**协议提供服务，新版本默认基于**Netty**提供序列化和数据传输能力。

Netty行业应用总结

- ✓高并发：由于采用异步非阻塞模式，一个Netty服务端可以同时处理成千上万的客户端；
- ✓高性能：Netty的综合性能在各个NIO框架中最高，它的单节点吞吐量非常高
- ✓安全性：支持HTTPS、SSL等，可以在传输层进行安全控制；
- ✓定制性：可以方便的实现业务逻辑的定制
- ✓可靠性：内存保护、流量整形等



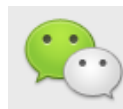
Q & A



谢谢各位聆听



@InfoQ



infoqchina

软件
正在改变世界!