



第二版：Netty 11 道

目录

第二版：Netty 11 道	1
1.Netty 是什么？	1
2.Netty 的特点是什么？	1
3.什么是 Netty 的零拷贝？	1
4.Netty 的优势有哪些？	2
5.Netty 的应用场景有哪些？	2
6.Netty 高性能表现在哪些方面？	2
7.Netty 和 Tomcat 的区别？	3
8.Netty 中有那种重要组件？	3
9.Netty 发送消息有几种方式？	4
10.默认情况 Netty 起多少线程？何时启动？	4
11.Netty 支持哪些心跳类型设置？	4

我们的网站：<https://tech.souyunku.com>

关注我们的公众号：搜云库技术团队，回复以下关键字

回复：**进群** 邀请您进「技术架构分享群」

回复：**内推** 即可进：北京，上海，广州，深圳，杭州，成都，武汉，南京，
郑州，西安，长沙「程序员工作内推群」

回复 **1024** 送 4000G 最新架构师视频

回复 **PPT** 即可无套路获取，以下最新整理调优 PPT!

46 页《JVM 深度调优，演讲 PPT》



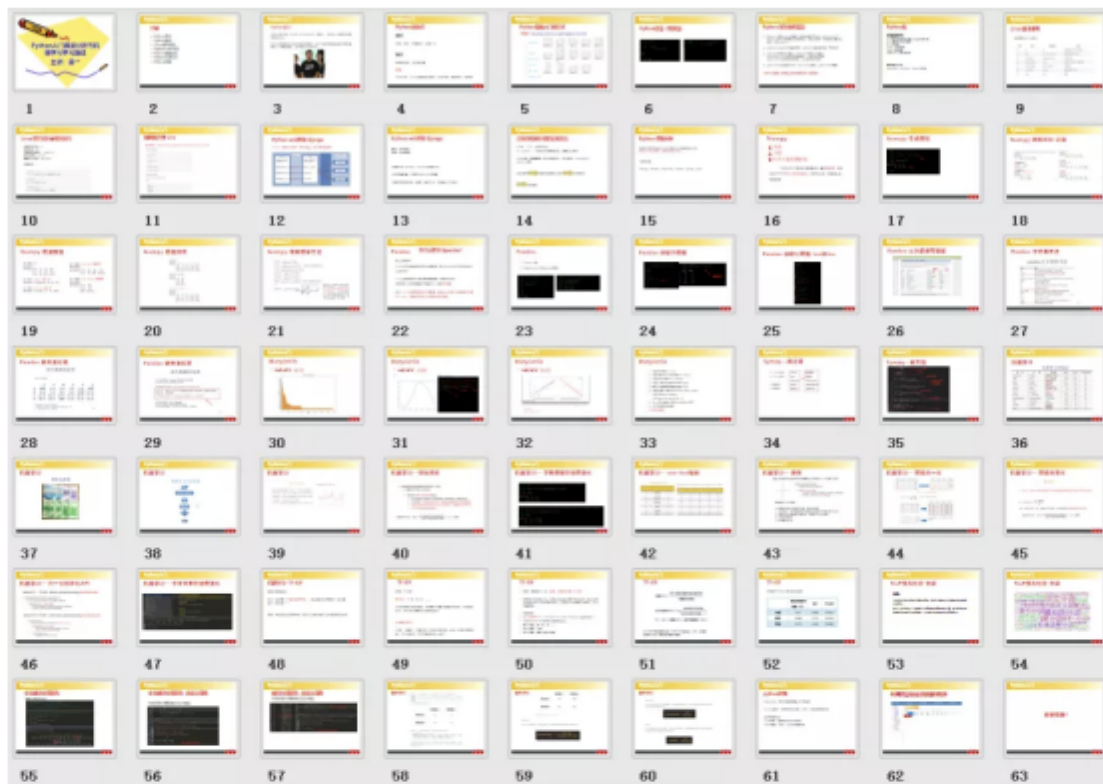
53 页《Elasticsearch 调优演讲 PPT》



63 页《Python 数据分析入门 PPT》

微信搜一搜

搜云库技术团队



微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推
专注于分享最有价值的互联网技术干货文章

1.Netty 是什么?



Netty 是一款基于 NIO (Nonblocking I/O, 非阻塞 IO) 开发的网络通信框架, 对比于 BIO (Blocking I/O, 阻塞 IO), 他的并发性能得到了很大提高。难能可贵的是, 在保证快速和易用性的同时, 并没有丧失可维护性和性能等优势。

2.Netty 的特点是什么?

- 高并发: Netty 是一款基于 NIO (Nonblocking IO, 非阻塞 IO) 开发的网络通信框架, 对比于 BIO (Blocking I/O, 阻塞 IO), 他的并发性能得到了很大提高。
- 传输快: Netty 的传输依赖于零拷贝特性, 尽量减少不必要的内存拷贝, 实现了更高效率的传输。
- 封装好: Netty 封装了 NIO 操作的很多细节, 提供了易于使用调用接口。

3.什么是 Netty 的零拷贝?

Netty 的零拷贝主要包含三个方面:

- Netty 的接收和发送 ByteBuffer 采用 DIRECT BUFFERS, 使用堆外直接内存进行 Socket 读写, 不需要进行字节缓冲区的二次拷贝。如果使用传统的堆内存 (HEAP BUFFERS) 进行 Socket 读写, JVM 会将堆内存 Buffer 拷贝一份到直接内存中, 然后才写入 Socket 中。相比于堆外直接内存, 消息在发送过程中多了一次缓冲区的内存拷贝。
- 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, 避免了传统通过内存拷贝的方式将几个小 Buffer 合并成一个大的 Buffer。
- Netty 的文件传输采用了 transferTo 方法, 它可以直接将文件缓冲区的数据发送到目标 Channel, 避免了传统通过循环 write 方式导致的内存拷贝问题。



4.Netty 的优势有哪些？

- 使用简单：封装了 NIO 的很多细节，使用更简单。
- 功能强大：预置了多种编解码功能，支持多种主流协议。
- 定制能力强：可以通过 ChannelHandler 对通信框架进行灵活地扩展。
- 性能高：通过与其他业界主流的 NIO 框架对比，Netty 的综合性能最优。
- 稳定：Netty 修复了已经发现的所有 NIO 的 bug，让开发人员可以专注于业务本身。
- 社区活跃：Netty 是活跃的开源项目，版本迭代周期短，bug 修复速度快。

5.Netty 的应用场景有哪些？

典型的应用有：阿里分布式服务框架 Dubbo，默认使用 Netty 作为基础通信组件，还有 RocketMQ 也是使用 Netty 作为通讯的基础。

6.Netty 高性能表现在哪些方面？

- IO 线程模型：同步非阻塞，用最少的资源做更多的事。
- 内存零拷贝：尽量减少不必要的内存拷贝，实现了更高效率的传输。
- 内存池设计：申请的内存可以重用，主要指直接内存。内部实现是用一颗二叉查找树管理内存分配情况。
- 串行化处理读写：避免使用锁带来的性能开销。
- 高性能序列化协议：支持 protobuf 等高性能序列化协议。



7.Netty 和 Tomcat 的区别?

- 作用不同: Tomcat 是 Servlet 容器, 可以视为 Web 服务器, 而 Netty 是异步事件驱动的网络应用程序框架和工具用于简化网络编程, 例如 TCP 和 UDP 套接字服务器。
- 协议不同: Tomcat 是基于 http 协议的 Web 服务器, 而 Netty 能通过编程自定义各种协议, 因为 Netty 本身自己能编码/解码字节流, 所有 Netty 可以实现, HTTP 服务器、FTP 服务器、UDP 服务器、RPC 服务器、WebSocket 服务器、Redis 的 Proxy 服务器、MySQL 的 Proxy 服务器等等。

8.Netty 中有那种重要组件?

- Channel: Netty 网络操作抽象类, 它除了包括基本的 I/O 操作, 如 bind、connect、read、write 等。
- EventLoop: 主要是配合 Channel 处理 I/O 操作, 用来处理连接的生命周期中所发生的事情。
- ChannelFuture: Netty 框架中所有的 I/O 操作都为异步的, 因此我们需要 ChannelFuture 的 addListener()注册一个 ChannelFutureListener 监听事件, 当操作执行成功或者失败时, 监听就会自动触发返回结果。
- ChannelHandler: 充当了所有处理入站和出站数据的逻辑容器。
ChannelHandler 主要用来处理各种事件, 这里的事件很广泛, 比如可以是连接、数据接收、异常、数据转换等。
- ChannelPipeline: 为 ChannelHandler 链提供了容器, 当 channel 创建时, 就会被自动分配到它专属的 ChannelPipeline, 这个关联是永久性的。

9.Netty 发送消息有几种方式?



Netty 有两种发送消息的方式：

- 直接写入 Channel 中，消息从 ChannelPipeline 当中尾部开始移动；
- 写入和 ChannelHandler 绑定的 ChannelHandlerContext 中，消息从 ChannelPipeline 中的下一个 ChannelHandler 中移动。

10.默认情况 Netty 起多少线程？何时启动？

Netty 默认是 CPU 处理器数的两倍，bind 完之后启动。

11.Netty 支持哪些心跳类型设置？

- readerIdleTime: 为读超时时间（即测试端一定时间内未接受到被测试端消息）。
- writerIdleTime: 为写超时时间（即测试端一定时间内向被测试端发送消息）。
- allIdleTime: 所有类型的超时时间。

作者：Java 中文社群

链接：<https://juejin.im/post/5c81b08f5188257a323f4cef>

来源：掘金

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。