

JavaAIO初探(异步网络IO)Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_JavaAIO_E5_88_9D_c104_644595.htm 按照《Unix网络编程》的划分，IO模型可以分为：阻塞IO、非阻塞IO、IO复用、信号驱动IO和异步IO，按照POSIX标准来划分只分为两类：同步IO和异步IO。如何区分呢？首先一个IO操作其实分成了两个步骤：发起IO请求和实际的IO操作，同步IO和异步IO的区别就在于第二个步骤是否阻塞，如果实际的IO读写阻塞请求进程，那么就是同步IO，因此阻塞IO、非阻塞IO、IO复用、信号驱动IO都是同步IO，如果不阻塞，而是操作系统帮你做完IO操作再将结果返回给你，那么就是异步IO。阻塞IO和非阻塞IO的区别在于第一步，发起IO请求是否会被阻塞，如果阻塞直到完成那么就是传统的阻塞IO，如果不阻塞，那么就是非阻塞IO。Java nio 2.0的主要改进就是引入了异步IO（包括文件和网络），这里主要介绍下异步网络IO API的使用以及框架的设计，以TCP服务端为例。首先看下为了支持AIO引入的新的类和接口：`java.nio.channels.AsynchronousChannel`标记一个channel支持异步IO操作。

`java.nio.channels.AsynchronousServerSocketChannel` `ServerSocket`的aio版本，创建TCP服务端，绑定地址，监听端口等。

`java.nio.channels.AsynchronousSocketChannel`面向流的异步socket channel，表示一个连接。

`java.nio.channels.AsynchronousChannelGroup`异步channel的分组管理，目的是为了资源共享。一个

`AsynchronousChannelGroup`绑定一个线程池，这个线程池执

行两个任务：处理IO事件和派发

CompletionHandler.AsynchronousServerSocketChannel创建的时候可以传入一个 AsynchronousChannelGroup，那么通过AsynchronousServerSocketChannel创建的

AsynchronousSocketChannel将同属于一个组，共享资源。

java.nio.channels.CompletionHandler异步IO操作结果的回调接口，用于定义在IO操作完成后所作的回调工作。AIO的API允许两种方式来处理异步操作的结果：返回的Future模式或者注册CompletionHandler，我更推荐用 CompletionHandler的方式，这些handler的调用是由AsynchronousChannelGroup的线程池派发的。显然，线程池的大小是性能的关键因素

。AsynchronousChannelGroup允许绑定不同的线程池，通过三个静态方法来创建：public static AsynchronousChannelGroup withFixedThreadPool (int nThreads , ThreadFactory threadFactory) throws IOException public static

AsynchronousChannelGroup withCachedThreadPool

(ExecutorService executor , int initialSize) public static

AsynchronousChannelGroup withThreadPool (ExecutorService executor) throws IOException 需要根据具体应用相应调整，

从框架角度出发，需要暴露这样的配置选项给用户。在介绍完了aio引入的TCP的主要接口和类之后，我们来设想下一个aio框架应该怎么设计。参考非阻塞nio框架的设计，一般都是采用Reactor模式，Reacot负责事件的注册、Oselect、事件的

派发；相应地，异步IO有个Proactor模式，Proactor负责CompletionHandler的派发，查看一个典型的IO写操作的流程来看两者的区别：Reactor：100Test 下载频道开通，各类考

来看两者的区别：Reactor：100Test 下载频道开通，各类考

试题目直接下载。详细请访问 www.100test.com