**模型：**

BIO（Blocking I/O，同步阻塞IO）

NIO（Nonblocking I/O，同步非阻塞IO）主动观察事件结束  
异步阻塞

AIO（异步非阻塞）等事件结束接收通知

**BIO,NIO,AIO区别：**

BIO:发起请求->一直阻塞->处理完成

NIO:selector主动轮询channel->处理请求->请求完成

AIO:发起请求------>通知回调

**阻塞非阻塞：**

1. 阻塞：线程等待结果返回完毕(同步)
2. 非阻塞：线程获取一个状态值，再去请求其它资源，直到拿到对应资源的完成通知(异步)

同步阻塞，同步非阻塞

异步阻塞，异步非阻塞

**NIO：**

多路复用

Channel（双向通道）

Buff（缓冲区）

Server(服务端)

(单线程)Selector(选择器，多路复用器):

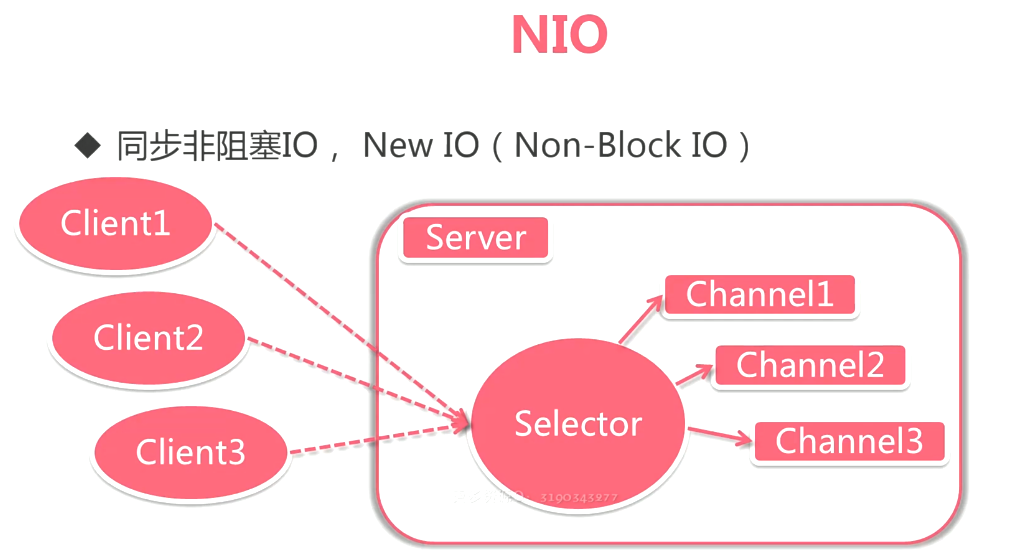
主动轮询bs。进行客户端，服务端注册建立channel连接。

channel中的数据都会存到buff(缓冲区中),读写通过buff完成(非阻塞)

Client1(客户端)

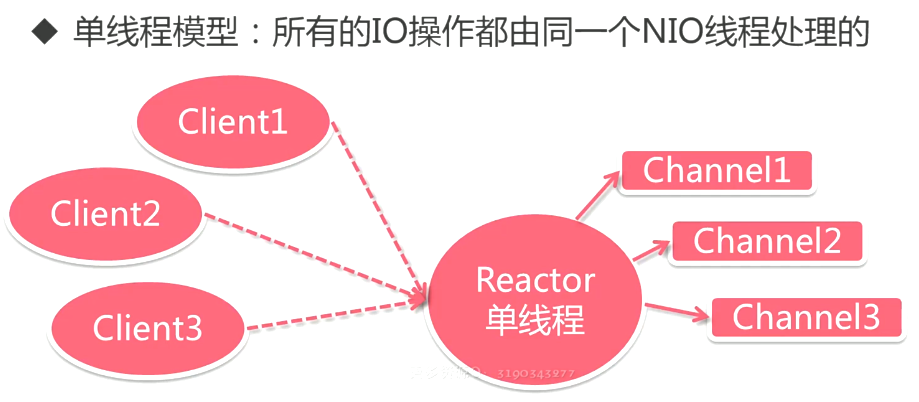
Client2

Client3

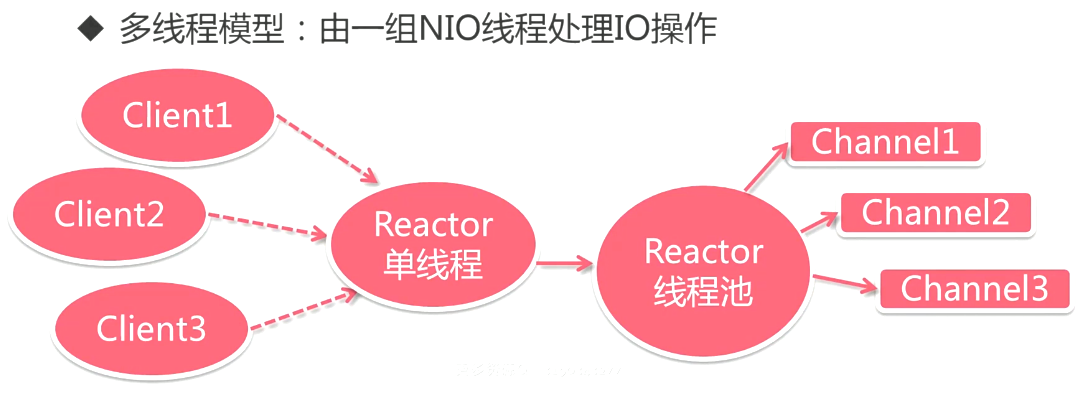


Reachor线程模型（三种）：

**单线程模型：所有的IO操作都由同一个NIO线程处理**



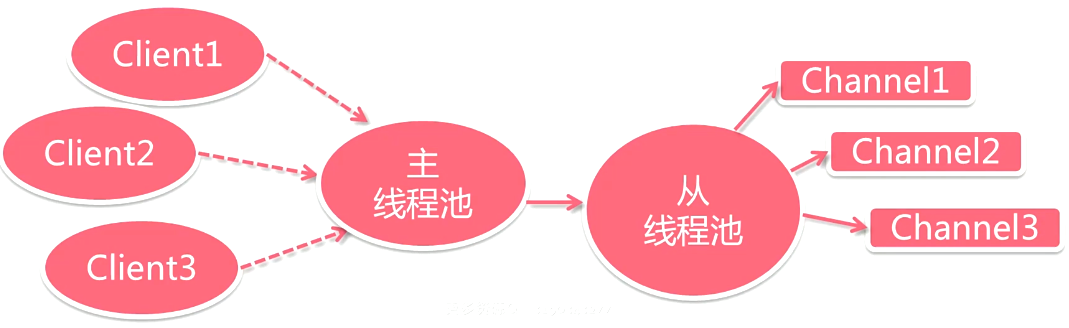
**多线程模型：由一组NIO线程处理IO操作**



左边处理与客户端连接，将其放入右边线程池

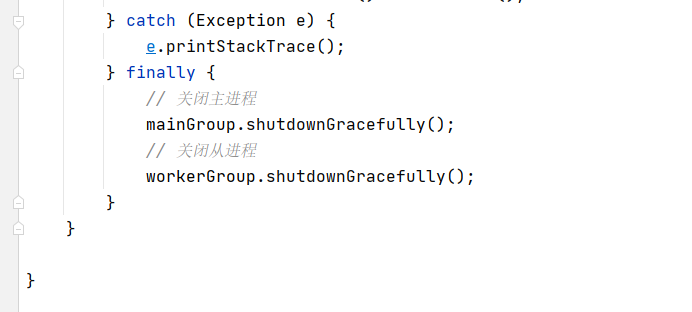
右边线程池处理额外的读写IO等操作

**主从线程模型：一组线程池接受请求，一组线程池处理IO（推荐）**



启动hello netty服务器





**设置channel初始化器**

每一个channel由多个handler共同组成管道(pipeline),管道内的handler(助手函数)需要自己编写

