<https://segmentfault.com/q/1010000004543463>

UNP网络编程中，有I/O复用的使用，以select和epoll为例，  
其中有以下说明：  
IO多路复用是指内核一旦发现进程指定的一个或者多个IO条件准备读取，它就通知该进程。IO多路复用适用如下场合：

　　（1）当客户处理多个描述字时（一般是交互式输入和网络套接口），必须使用I/O复用。

　　（2）当一个客户同时处理多个套接口时，而这种情况是可能的，但很少出现。

　　（3）如果一个TCP服务器既要处理监听套接口，又要处理已连接套接口，一般也要用到I/O复用。

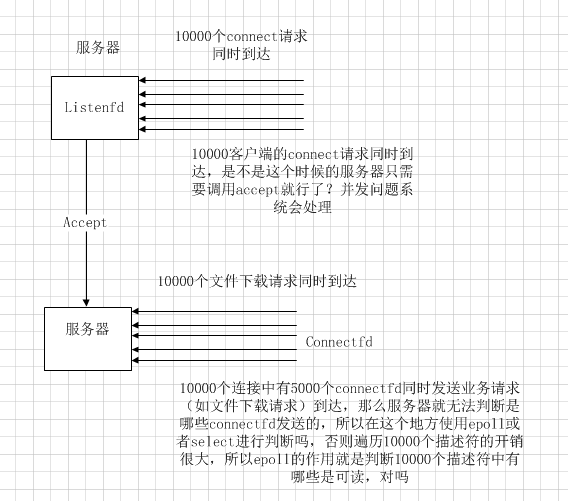
　　（4）如果一个服务器即要处理TCP，又要处理UDP，一般要使用I/O复用。

　　（5）如果一个服务器要处理多个服务或多个协议，一般要使用I/O复用。

　　与多进程和多线程技术相比，I/O多路复用技术的最大优势是系统开销小，**系统不必创建进程/线程，也不必维护这些进程/线程**，从而大大减小了系统的开销。

我的理解：假设现在已经有10000个客户端和服务器通过TCP三次握手，处于已连接状态，也就是有10000个connected\_socket，当这10000客户端同时发送数据请求时，服务器通过使用select或者epoll可以同时处理这10000个请求，请问我理解的IO复用对吗？

如果是的话，假设10000个客户端同时发送文件下载的请求，不还是要创建多个进程/线程来处理下载请求吗，否则，一个个的处理，最后一个客户端的文件下载请求要很长时间才能完成啊，怎么就不必创建多个进程/线程



回答：

1.

个人觉得，IO复用的本质是在内核级别对fd进行轮询，哪个准备好了就通知用户代码，这么做最优。如果没有IO复用的话，你需要自己去轮询哪个fd准备好了，亦或者再次一点，一个线程阻塞等待一个fd。

看了你画的图，我再补充一点。IO复用只在fd是否就绪这个问题上帮助用户代码，所谓就绪包括有下载请求到来。但是真正处理下载（即像客户端发送数据），是由用户自己的工作线程去处理的，如果同时请求量很大，超过了单机的处理能力，那么用户需要自己设计排队或者分流机制，这跟用不用IO复用没有关系。

2.

如果没有io复用, 一个线程同时只等待一个fd. 为了同时等待很多fd就需要有很多线程, 这导致不小的线程切换开销.

复用(select/epoll)是说你可以用**少量**线程同时在很多fd上等待

3.

我觉得题主可以再多理解一下并发。首先不论创建1000个线程还是少量线程做IO复用，都离不开排队执行。并发的数量跟你使用的CPU的core数量有关系。我这里说的并发是指同一时间内同时处理请求的数量，时间单位你可以认为是1个CPU cycle。假设是4核的CPU，那最大并发数是4。你开1000线程和开4个线程IO复用，大家都是来抢4个CPU资源排队执行，但4个线程的做法明显开销小。另外很多人谈并发的时候是指再一段很长时间单位内比如1min，server的同时处理能力。题主要区分一下。如果想了解更多，可以关注一下我的几篇文章。