

## ZServer4D 日常问题总结

日常问题收集于 qq 的高频率提问和解答，由于我使用拼音输入法，而拼音输入法都很蠢，请理解一下。

### 目录

zServer4D 究竟是个什么东西？ .....	1
zServer4D 可以干什么用？ .....	2
zServer4D 可以带动多少用户？ .....	2
zServer4D 为什么不支持 http 和 https .....	2
zServer4D 要开发 http 支持的后台应该怎么来做？ .....	2
zServer4D 中的 ICS 服务器接口是怎么工作的？ .....	2
zServer4D 中的 ICS 客户端接口是怎么工作的？ .....	3
zServer4D 中的 Indy 做服务器怎么样？ .....	3
用 zServer4D 中的 CrossSocket 接口做服务器怎么样？ .....	3
为什么 zServer4D 没有集成 DIOCP 这种优秀的 IOCP？ .....	4
基于 zsGateway 做内网穿透是什么原理？ .....	4
我的操作系统应为用的 ZS 被卡死，连任务管理器都打不开，这是 bug 吗？ .....	4
ZServer4D 收费吗？ .....	4
为什么双通道验证服务器框架，在验证不通过时，不给客户端断线？ .....	5
双通道的工作模型和命令执行流程是什么？ .....	5
我做业务开发需要考虑同步和异步吗？ .....	5
用 ZSGateway 架设宅服是不是真的可以一年省几十万？ .....	6
更多解答请参考单元库说明 .....	6

## zServer4D 究竟是个什么东西？

从架构来说，zSERVER4D 是一个中间件。

从生产来说，zSERVER4D 是一套后台系统的开发工艺，也可以说是一套系统标准。

从程序和技术来说，iocp,epoll,kqueue 它们只是一套从操作系统提供出来的服务器开发接口，拥有这类接口的后台服务器现在多如牛毛，而我们基于这些接口要做系统级开发，是件非常艰深的工作：优化内存池，服务器系统安全，处理并发后台的同步和异步，处理各种错误，新增通讯接口，兼容手机，Linux，Windows 等等平台。zSERVER4D 是作为系统的生产工艺，用标准开发模型解决了这些技术问题。

## zServer4D 可以干什么用？

zs 是拿给你做二次开发的底层，你要用它，需要做二次开发，包成你的业务系统，你需要自己设计你的后台开发工艺后，你才能展开开发工作。简单来说，你需要做一次系统设计，但是这并不难，你需要花上一周切入和掌握 zs 的模型，到完成系统，你只会花一个月，你将拥有自己的后台开发工艺，这会更针对你的开发工作，或则说更针对你的公司，而你不必依靠三方厂商给你提供工艺，你将拥抱自由和核心技术：后台制作工艺。

Zs 支持的后台业务系统可以是任何面向终端用户的后台，包括多媒体，游戏，日常应用

## zServer4D 可以带动多少用户？

单独 zs 的服务器承载量可能上万，也可能几千，根据你对业务实现而定

如果你基于 zs 提供的云服务器模型开发后台，服务器承载量可以轻松跑上百万，也可能是几万，根据你对业务实现而定。

## zServer4D 为什么不支持 http 和 https

HTTP 的后台开发控件多如牛毛，开发类似 https 的后台服务，请自行使用三方控件去集成

## zServer4D 要开发 http 支持的后台应该怎么做？

在你的业务核心实现 HTTP 即可（通常来说 app 的业务核心都集中在 DB，如果你的 DB 是分布式，要跑高并发，请考虑 http 分流），面向外网使用 nginx，frp 这类工具做个反向代理，cache 处理等等机制，做 HTTP 架构的解决方案请自行知乎+百度。

## zServer4D 中的 ICS 服务器接口是怎么工作的？

ICS+zs 做服务器，是 100%的异步框架，它是每个链接创建一个独立线程，在没有响应时，ics 用消息钩子 getmessage 自己把自己卡在线程中。由于 getmessage 内部是用 sleep 在巡检消息 所以 ics 在响应时 会有一个延迟 这也是用消息内核做服务器的通病之一。另一点是都线程需要 1M 以上的堆栈，当 ics 的链接多了，线程多了，莫名其妙的内存开销也跟着多起来。

## zServer4D 中的 ICS 客户端接口是怎么工作的？

**ICS+zs 做客户端是稳定后台接口。**ics 客户端接口也是基于消息，没有线程，每次收到数据或则发送完成数据，均会触发一次消息，告诉框架。和 Tform 的工作原理类似，如果不考虑 0.5 秒的延迟，它是非常稳定的接口。

## zServer4D 中的 Indy 做服务器怎么样？

**INDY 的服务器是半异步框架。**用 indy10 这个版本，做服务器，其实也还可以。indy10 的服务器和 ics 类似 都是一个链接对应一个独立线程，用线程池来进行管理，zs 会在线程中用同步方式处理一次 buff，所以 indy+zs 做服务器只能算半异步框架。indy 的线程优于不用 getmessage 等待，所以 indy 的后台非常快，处理速度几乎可以媲美 iocp。由于 indy 的设计就是以前端应用为主，没有考虑并发，我们只能用 indy 带起百来个链接，多了 indy 服务器就会崩溃。

为什么正规方案有很多项目都，使用 indy 做 http 服务器，而 indy 只能带起百来个链接。那是因为 http 是短链接，处理完成 response 之后，链接就被断开了，随之附带的线程也被结束，系统开销因此降低，消能自然就起来了。

## 用 zServer4D 中的 CrossSocket 接口做服务器怎么样？

**CROSS+zs 是半异步框架。**cross 在 windows 是用的完成端口，完成端口是硬件提供给驱动程序使用的 api，专用于卡线程，我们在硬盘，光驱，nvidia 的驱动，都有使用完成端口，完成端口卡线程和消息又不一样，完成端口是基于硬件响应和挂起线程的方法，特别适用于高频率触发运停线程。

简单来说,为什么 iocp+poll+epoll+kqueue 会在服务器显得特别强大,就是因为卡线程的机制。我躺过 cross 坑非常多,很多地方,几乎无法调试,在 onReceive 事件内部,对线程池的管理分别有 kq+ep+iocp,管理是反查 hash 表,有点反人类,我不想改太多 cross 的代码,所以,我用的是同步方式 fill 缓冲区,因为这一个函数,就让 fill 工作变成了半异步化了,经过我实际测试,效率,还能接受,毕竟现在服务器主频都在 1G 以上,后台写精简点,单台跑上 1 万并发是没问题的。

CrossSocket 另一个好处是支持 Linux 的 Epoll :>

## 为什么 zServer4D 没有集成 DIOCP 这种优秀的 IOCP?

由于开源的问题，许多商业服务器方案，我不太好明显的集成到 zs，而 diocp 这类优秀的底层，是因为它太庞大同时又有点 d7 的影子，要做许多改进才能使用，没空做，所以暂时，只支持了 indy,ics,cross 三套底层通讯库。

## 基于 zsGateway 做内网穿透是什么原理？

zSGATEWAY 的内网穿透是基于流量代理，将远程服务器的流量，转发到你的宅服实现的内网穿透。核心是使用的 frp（支持国货），通过预编译好的 oxc 展开配置（oxc 是 frp 的可执行文件打包）。

我的操作系统应为用的 zs 被卡死，连任务管理器都打不开，这是 bug 吗？

新一代的操作系统，都把内核 api 拿出来公用了，而 pasmp 就使用操作系统的 atomic，它基于硬件驱动程序的 api 体系在工作。如果跑的如果不是高并发，比如用分布式云解决业务问题，使用时请关闭。ATOMIC 会有一定几率造成操作系统卡顿 甚至连任务管理器都无法打开（去问微软），如果你的一台服务器承载的并行太多，卡顿会是经常的，建议你只对负载最高的那台服务器开并行，开一台服务器并发就足够了。开关并行，在 zDEFINE.INC 中会找选项。

## ZServer4D 收费吗？

免费的！我都 340 岁了，我还纠结什么付出回报，大家全都是有需求，需要做项目，要解决目标问题。 穷人也好，有钱人也好，我都不在意！只要我能帮大家解决问题，哪怕一点点，我就觉得开心！我觉得满足！而且，我很不喜欢搞保留技术。

## 为什么双通道验证服务器框架，在验证不通过时，不给客户端断线？

未验证，或验证不通过，并不会马上断线，而是先发一个反馈过去，告诉那台客户端，是什么导致了验证不通过 什么时候断线 是由客户端决定的。

如果验证不通过 马上就断线 客户端会云里雾里 不知道怎么回事，其次 如果在 login 命令验证不通过 马上断线 后面会报告一条 `send failed, response failed` 这类提示，咋一看你还以为是出 bug 了，其实是因为 Login 是反馈型的 cmd，这是因为数据缓冲还在队列中，但是客户端已经断线，底层检测到目标无效，然后抛了个提示。

正常情况下，LOGIN 是反馈型的 CMD，按机制思路，要在反馈完成以后才去断线

## 双通道的工作模型和命令执行流程是什么？

- 1，客户端 connect
- 2，这时候，客户端只可以执行 `connectInit,Login,RegUser` 这三条指令，别的指令都不能执行
- 3，当执行完成 Login 后，必须 TunnelLink 建立对等通道，一个收一个发
- 4，这时候才能正常执行指令，在第 3 步未处理前，所有的指令都会拒绝执行，Demo 都有范式的

另一个 NoAuth 的双通道工作模型

- 1，客户端 connect
- 2，当执行完成 connect 后，必须 TunnelLink 建立对等通道，一个收一个发，省去了 Login,RegUser 这一步
- 3，TunnelLink 成功以后，候才能正常执行指令，在第 2 步未处理前，所有的指令都会拒绝执行，Demo 也是有范式的

## 我做业务开发需要考虑同步和异步吗？

所有的 buff 收发，是根据不同底层，indy，cross，ics 有些是多线程收发，有些是 mainThread 收发。但是到执行命令，触发事件时，所有的命令和事件都在主线程执行，就是 mainThread，做业务开发时不用考虑底层的同步异步问题

用 **ZSGateway** 架设宅服是不是真的可以一年省几十万？

答案：[可以省很多钱](#)。

更多解答请参考单元库说明

By qq600585

2018-1-20