

## 延迟反馈机制详解

介绍:

在标准响应式的通讯模型中（标准通讯架构）:

1. Client 向 server 发送一条命
2. server 处理
3. sever 反馈给 client

延迟反馈是从内核机制在第 2 步执行处理,在第 3 步暂停反馈(使用 `PauseResultSend` 方法),在暂停反馈其间,服务器不会有任何 `wait` 机制执行（零时间片和 `cpu` 消耗）直到服务器使用（`ContinueResultSend`）方法才会继续反馈

下面,我们来对比传统的延迟方法和 `zServer` 的延迟反馈方法

以传统 `http` 为例

- 1, client 向 `http` 服务器发送一条请求
- 2, `http` 服务器收到请求
- 3, `http` 服务器使用卡线程或则卡调用方法,在请求触发事件中做循环检查
- 4, `http` 服务器再向另一台 `http2` 服务器发送请求
- 5, 当 `http2` 反馈了请求
- 6, `http` 停止循环检查,并且反馈给客户端

`zServer` 的延迟反馈例子:

- 1, client 向 server 发送一条请求
- 2, server 收到请求
- 3, server 使用 `PauseResultSend`,然后退出请求触发事件
- 4, server 再向另一台 `server2` 发送请求
- 5, 当 `server2` 反馈了请求
- 6, 在 `server2` 反馈回来请求的触发事件中,server 使用 `ContinueResultSend`,立即反馈给客户端

对比两种机制,`zServer` 的延迟机制是专为跨服务器通讯而设计的,延迟反馈具有零资源消耗和使用更加简易的特点

By600585

2018-5