

Zserver 中的序列包系统详解

目录

Zserver 中的序列包系统详解.....	1
概念:	1
tcp 序列包机制的优点:	1
tcp 序列包机制的缺点:	2
ZServer 序列包机制的优点	2
Zserver 序列包的缺点.....	2
序列包的模型流程.....	3
发送端 IO.....	3
接收端 IO.....	3

概念:

序列包的概念是底层 TCP 的信号校验系统: 网络数据以流方式传输以序列方式进行, 每一个 IP 包都会有一个序列号, 当校验发现数据丢失, 就是信号不稳定, 那么, 在序列中的某一个 IP 包, 就会发生“插队”和“重发”工作。

我们在实现 UDP 的稳定通讯时, 数据校验和序列包就会是一个重要的开发工作, 因为 UDP 没有做序列化处理。

序列包的原理就是如上了

tcp 序列包机制的优点:

- 1, 稳定的数据流校验, 发送和接收数据, 总是会一致
- 2, 序列包可以很好的还原协议栈的工作现场

tcp 序列包机制的缺点：

- 1，使用序列包，会一发一收，来确保数据一致性，导致数据从发送，到接收时，延迟更大
- 2，序列包发送以后，需要很多临时内存对重发请求做准备
- 3，序列包在接收数据时，需要临时内存做数据序列化交换

ZServer 序列包机制的优点

- 1，机理和 TCP 相同
- 2，相比 tcp，Zserver 多了信号开关
- 3，比 tcp 拥有更大的协议栈内存使用，通过加大内存，可以轻松还原 5-60 分钟的通讯现场，也就是说，你断线 60 分钟，然后重连，IO 现场还在，在断线过程中，所有收发的数据包都没有丢失，也不会出错
- 4，支持 UDP 的傻瓜化移植，我们可以自行使用 UDP 在 Zserver 地基上轻松实现稳定通讯
- 5，提供应用程序支持，可以选择走 Zserver 内置序列包协议通讯，以及默认 IO 通讯协议
- 6，ZServer 的序列包内置了分包机制，在公网能节省 5%-10%的带宽流量
- 7，有物理时间概念，自动化控制时间和流量的协调工作

Zserver 序列包的缺点

- 1，数据收发时，性能相比纯 IO 更低
- 2，序列包机制是针对断线重连设计的，它非常吃内存，假如你在 1 秒物理时间中收发一个 1M 的数据包，那么序列包会产生 $1M/mtu \approx 1000$ 个小包，这小包，会在 1 秒的收发时间中，一直暂用 $1M \times 2-3$ 倍空间的内存，当收发完成，才会被释放。在高并发后台中，序列包并不实用，它只能针对现场还原。当然，你可以控制序列包能使用的内存量，也可以使用云服务器模型来分布式负载，还可以使用 XNAT 来做内网服务器，搭建内存超大的后台服务器来支持断线重连。

序列包的模型流程

画图太麻烦了，直接文字描述，请结合程序来理解

发送端 IO

- 1, 开始发送一个 1024kb 的数据块
- 2, 系统以 1024kb / MTU 进行分包，加上序列号
- 3, 每个分包会存在于临时内存表，直到有接收成功的信号才释放
- 4, 按物理时间，检查发送表，需要重发的地方，重发，超出安全限制，断开连接

接收端 IO

- 1, 收到序列包
- 2, 数据校验，如果出问题，发信号让发送端 IO 重发序列包
- 3, 如果数据校验成功，发信号告诉发送端成功收到序列包，你不用再再缓存它了
- 4, 检查数据序列，如果序列符合向前条件，将序列包写入本地序列池
- 5, 按计数器从序列池拾取数据，如果发现序列不存在，发信号让发送端 IO 重发序列包
- 6, 将拾取数据合并到数据块

By qq600585

2018-11-9